А. В. ГЛАДКИЙ

ФОРМАЛЬНЫЕ ГРАММАТИКИ И ЯЗЫКИ

ИЗДАТЕЛЬСТВО «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
МОСКВА 1973

518
Г 52
УДК 519-95

Формальные грамматики и языки. Гладкий А. В.

Книга посвящена теории формальных грамматик и языков, являющейся важнейшей составной частью так называемой математической лингвистики. Эта теория вызвана к жизни потребностями лингвистики, но нашла свою почву в чистой математике и стала полноправной отраслью математической логики, тесно связанной с теорией алгоритмов и теорией автоматов.

В книге рассматривается ряд важных проблем теории формальных грамматик — таких, как взаимоотношения между различными классами грамматик и классами задаваемых ими языков, связь между грамматиками и автоматами, оценки сложности вывода в грамматиках, алгоритмические проблемы для грамматик.

Книга представляет большой интерес для специалистов как в области математической лингвистики, так и в смежных областях, например в теории алгоритмов и автоматов.

Страниц 368. Иллюстраций 17.

Алексей Всеволодович Гладкий
ФОРМАЛЬНЫЕ ГРАММАТИКИ И ЯЗЫКИ
М., 1973 г., 368 стр. с илл.

Редактор *В. В. Донченко*

Техн. редактор *Е. Н. Земская* Корректор *Н. Б. Румянцева*

Сдано в набор 10/VII 1972 г. Подписано к печати 8/II 1973 г. Бумага 84×108/32, тип. № 2. Физ. печ. л. 11,5. Условн. печ. л. 19,32. Уч.-изд. л. 20,20. Тираж 7000 экз. Т-00756. Цена книги 1 р. 51 к. Заказ № 273

Издательство «Наука»
Главная редакция физико-математической литературы
117071, Москва, В-71, Ленинский проспект, 15

Ордена Трудового Красного Знамени
Ленинградская типография № 2 имени Евгении Соколовой
«Союзполиграфпрома» при Государственном комитете Совета Министров
СССР по делам издательств, полиграфии и книжной торговли
г. Ленинград, Л-52, Измайловский проспект, 29

Г $\frac{0223-1714}{042\,(02)-73}$ 37-72

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

Настоящая книга представляет собой руководство по теории формальных грамматик, предназначенное для тех, кто хочет познакомиться с ее современным состоянием достаточно основательно — в общеобразовательных целях или ради прикладных задач (например, связанных с языками программирования). Это первая книга такого рода на русском языке; монография [Ginsburg 1966] *), русский перевод которой вышел в 1970 г., посвящена исключительно бесконтекстным языкам **); другая недавно переведенная книга [Gross — Lentin 1967] значительно элементарнее и содержит лишь основные понятия теории формальных грамматик ***).

Книга адресована в первую очередь читателю-математику; во всяком случае, для ее чтения нужна извест-

*) Фамилия и год в квадратных скобках — ссылка на помещенный в конце книги список литературы. Если в списке указано несколько работ одного автора, имеющих один и тот же год издания, то добавляются буквы в круглых скобках, например [Chomsky 1959(б)].

**) Кроме того, в настоящей книге изложены некоторые разделы теории бесконтекстных языков, которых нет в [Ginsburg 1966]: оценки сложности вывода, категориальные и доминационные грамматики, металинейные и итерационно-линейные языки; подробнее трактуются алгоритмические проблемы. С другой стороны, мы не рассматриваем преобразователей и ограниченных бесконтекстных языков, изучению которых уделено много внимания в книге С. Гинзбурга. Еще одно отличие нашей книги от названной — меньшая формализованность изложения.

***) Кроме названных двух, в мировой литературе имеется (насколько известно автору) еще только одна книга по формальным языкам [Hopcroft — Ullman 1969], также довольно сильно отличающаяся по содержанию от нашей.

ная привычка к математическому мышлению и изучению математической литературы. Что касается фактических знаний, то формально их почти не требуется: достаточно владеть основными понятиями теории множеств. Тем не менее весьма желательно предварительное знакомство хотя бы с главными идеями и концепциями теории алгоритмов, в частности с машинами Тьюринга, поскольку «алгоритмический дух» пронизывает всю теорию грамматик; не приобщившись к нему, невозможно составить представление о месте формальных грамматик в математике. При этом речь идет именно о направляющих идеях и общем духе: все относящиеся к алгоритмам технические определения — в терминах машин Тьюринга — в книге приводятся, и все необходимые факты доказываются *).

Лингвистические интерпретации упоминаются в основном тексте книги лишь изредка и вскользь (больше внимания им уделено в приложениях I и II, но рассматриваемый там материал не относится, строго говоря, к грамматикам). Однако знакомство с лингвистическим аспектом теории грамматик представляется автору полезным для ее изучения даже в чисто математическом плане; для ознакомления с этим аспектом можно рекомендовать книгу [Гладкий — Мельчук 1969]. «Программистских» приложений мы не касаемся.

В конце каждой главы помещены упражнения. Некоторые из них носят чисто тренировочный характер, но большинство предназначено служить дополнением к основному тексту; в упражнения вынесены многие факты, наличие которых в тексте не обязательно для логической последовательности изложения, а в отдельных случаях и такие утверждения, которые в дальнейшем используются, если их доказательства просты и основаны на уже «отработанной» технике. Диапазон трудности упражнений весьма широк; к некоторым из наиболее трудных и наиболее важных даны краткие указания.

*) В частности, полностью замкнутым в себе сделано изложение неразрешимых алгоритмических проблем (гл. 8). Из теории алгоритмов в этом изложении используется только теорема Райса, доказательство которой приводится там же.

Список литературы не претендует ни на какое подобие полноты; в него включены только те работы, которые упоминаются в книге, главным образом в библиографических замечаниях, также не претендующих на полный охват относящихся к делу источников.

При написании книги автор использовал свой курс лекций, изданный в свое время ротапринтным способом [Гладкий 1966]; однако имеющийся там материал составляет менее половины объема книги и излагается в существенно переработанном виде.

Автор рассматривает эту книгу как некоторый итог своих занятий математической лингвистикой — в особенности чтения лекций и ведения семинаров по этому предмету — в течение последних десяти лет, и ему хотелось бы упомянуть здесь о тех, кто в наибольшей степени помог ему в этих занятиях и непосредственно в работе над книгой. Прежде всего, интересы автора в теории грамматик сформировались в значительной степени под влиянием Б. А. Трахтенброта. Читатель, знакомый с его работами по теории сложности вычислений, легко поймет, в чем это влияние проявилось. Очень многим автор обязан слушателям своих лекций и участникам семинаров, их постоянному вниманию и критике; не имея возможности назвать их всех, он хотел бы особо упомянуть А. Я. Диковского, Л. С. Модину, Н. Г. Щербакову, М. К. Валиева, Р. В. Фрейвалда. Для выработки взглядов на лингвистическую сторону дела первостепенное значение имела многолетняя совместная работа с И. А. Мельчуком. Важными были замечания читателей упоминавшегося выше курса лекций, особенно М. В. Ломковской. В оформлении рукописи неоценимую помощь

оказали автору товарищи по работе, главным образом Т. И. Шедько, Л. С. Модина и Н. Г. Щербакова. Рукопись была прочитана рядом коллег, которые высказали немало ценных соображений и помогли устранить многие погрешности. Особенно существенными были замечания и предложения М. И. Белецкого; важные указания автор получил от М. В. Ломковской, Л. С. Модиной, а по приложению I — также от И. А. Мельчука и Е. В. Падучевой и по приложению II — от И. И. Ревзина. Р. Д. Равич составила предметный указатель и много помогала автору на последней стадии оформления рукописи. Л. С. Модина составила указатель обозначений. Всем этим коллегам автор выражает искреннюю признательность.

ВВЕДЕНИЕ

Когда говорят о «владении» или «умении пользоваться» тем или иным естественным языком, под этим понимают, с одной стороны, умение строить «правильные тексты» на этом языке, выражающие заданный смысл, с другой — умение определять смысл заданных «правильных текстов». При этом «правильные тексты» представляют собой последовательности элементарных единиц из некоторого конечного набора (скажем, фонем), образованные по определенным правилам; в совокупности эти правила составляют то, что называют грамматикой языка. Сами по себе элементарные единицы никаким смыслам не соответствуют, смысл имеют только их сочетания; поэтому их внутреннее строение и материальная природа для грамматики несущественны. С аналогичной ситуацией приходится встречаться в математике: в различных ее разделах используются различные «языки», выражения которых также строятся по определенным правилам из элементарных единиц (элементарных символов); этим выражениям сопоставляются определенные смыслы, представляющие собой комбинации содержательных математических понятий. Здесь, правда, смыслы приписываются обычно и элементарным символам (например, в языке арифметики + означает сложение, 1 — единицу и т. п.), однако отсутствует какая-либо связь между «внутренним устройством» элементарного символа и природой сопоставляемого ему понятия, так что и в этом случае внутреннее строение элементарных символов несущественно и выбор их произволен *). Математические языки отличаются от естественных несравненно более простым строением, позволяющим достигать в их описании предельной четкости и однозначности; зато они столь же сильно уступают естественным языкам в широте сферы применимости.

*) Если отвлечься от таких непринципиальных соображений, как удобство воспроизведения и запоминания, следование традиции и т. п.

В последнее время возникла особая разновидность математических языков — так называемые языки программирования, предназначенные для работы с вычислительными машинами; они, как правило, имеют более универсальную применимость и соответственно более сложное строение, чем «традиционные» математические языки, но и в том и в другом отношении весьма далеки от естественных. Так или иначе, в самых разных областях (и не только в упомянутых выше) возникают ситуации, когда приходится иметь дело с множествами выражений, построенных по определенным «грамматическим правилам» из элементарных единиц и используемых для передачи некоторых «смыслов». Представляет интерес исследование этой ситуации в общем виде, т. е. описание общих закономерностей строения самих грамматических правил; знание таких закономерностей позволило бы, в свою очередь, устанавливать общие закономерности строения образуемых по этим правилам выражений («текстов»). Именно это и является предметом теории грамматик — особой математической дисциплины, возникшей в 50-х годах нынешнего столетия. В ней строятся и изучаются формальные грамматики — абстрактные объекты, призванные служить моделями конкретных систем грамматических правил. Формальным грамматикам отвечают множества «правильно построенных выражений» — формальные языки, служащие моделями совокупностей «правильных текстов».

Возникнув в результате усилий, направленных на выработку точных методов описания естественного языка, теория грамматик сразу же сомкнулась, с одной стороны, с теоретической лингвистикой, которая по существу всегда занималась построением абстрактных моделей *), но не имела средств для их точного описания (что немало препятствовало ее развитию), с другой — с теорией алгоритмов и теорией автоматов, где среди прочих проблем возникает вопрос о конструктивных способах задания множеств последовательностей символов, т. е. по существу тот самый вопрос, систематическое изучение которого составляет основное содержание теории грамматик; поэтому теория грамматик с самого начала смогла воспользоваться некоторыми понятиями и методами, выработанными в теории алгоритмов и теории автоматов.

*) Такие элементарные понятия «школьной» грамматики, как подлежащее, сказуемое, падеж, род и т. п., представляют собой весьма абстрактные конструкции, относящиеся к фактам языка именно как модели. Осознать их абстрактный характер трудно только из-за их привычности.

Эти понятия и методы получили в ней своеобразную интерпретацию и были развиты в новых направлениях, однако связь и переплетение теории грамматик с теорией алгоритмов и в особенности с теорией автоматов остаются чрезвычайно сильными; по существу теория грамматик может считаться, как и теория автоматов, специфическим ответвлением теории алгоритмов, причем во многих местах обе ветви срастаются. Это относится как к методам, так и к постановкам задач и характеру результатов.

Место теории грамматик внутри общей теории алгоритмов может быть — весьма грубо и приблизительно — охарактеризовано следующим образом: теория грамматик изучает в известных аспектах «внутреннее строение» рекурсивных множеств некоторых специальных классов, по преимуществу наиболее простых, и как раз настолько простых, что реально возникающие в математике конструкции не выводят за пределы этих классов, за исключением тех (как правило, диагональных) конструкций, которые специально предназначаются для доказательства существования сколь угодно сложных объектов.

Говоря о прикладном значении теории грамматик, следует прежде всего отметить ее приложения к лингвистике, которой формальные грамматики доставляют весьма удобный метаязык. Кроме того, они нашли широкое применение для описания упоминавшихся выше языков программирования. Кажется достаточно правдоподобным, что сфера приложений формальных грамматик, в частности в гуманитарных исследованиях, будет со временем расширяться *).

*) Имеются чрезвычайно интересные опыты использования конструкций, по существу представляющих собой формальные грамматики, в поэтике; см. «Морфологию сказки» В. Я. Проппа (1-е изд., Л., 1928; 2-е изд., «Наука», 1969), недавние работы А. К. Жолковского и Ю. К. Щеглова.

Основоположником теории формальных грамматик по праву считается американский лингвист Н. Хомский. Именно он указал на возможность использовать для описания естественных языков некоторые исчисления, рассматривавшиеся ранее в теории алгоритмов (но не игравшие там особенно существенной роли), придав им удобную для этого форму (с тех пор за такими исчислениями закрепилось данное Н. Хомским название «порождающих грамматик»); он же ввел в рассмотрение те специальные классы порождающих грамматик, которые представляют для лингвистических приложений наибольший интерес, и указал пути их использования в лингвистике (замечательно, что именно эти классы грамматик оказались связанными с упоминавшимися выше простыми классами рекурсивных множеств); наконец, он положил начало математическому исследованию формальных грамматик, введя некоторые важные для этого понятия (в частности, понятие машины с магазинной памятью) и доказав ряд основных фактов *).

Сделаем еще несколько замечаний, относящихся уже не столько к самой теории грамматик, сколько к ее изложению в настоящей книге.

1) Теория грамматик является в настоящее время настолько разветвленной дисциплиной, что было бы очень трудно собрать в одной книге все ее наиболее интересные результаты или хотя бы отразить все направления исследования. Такую задачу автор перед собой не ставил, и вряд ли это было бы целесообразно. Имелась в виду совсем другая цель, а именно: изложить основы этой теории, ее «классические» понятия и факты, а из направлений более специальных и более новых лишь некоторые, но достаточно полно и подробно — так, чтобы читатель мог не только составить представление об общем характере теории грамматик, но и почувствовать используемые в ней методы, ее «кухню», а при наличии желания и прочих условий также и подготовиться к самостоятельной работе в этой области.

Разумеется, при выборе специальных вопросов для изложения в книге автор в значительной мере руководствовался своими личными вкусами.

*) Весьма существен также вклад Н. Хомского в собственно лингвистику; это, однако, не относится к предмету настоящей книги.

Среди сравнительно новых разделов теории грамматик, не получивших отражения в книге, особого упоминания заслуживает «теория управления выводом», представляющаяся весьма перспективной как с теоретической, так и с прикладной точки зрения. Эта область заслуживает, впрочем, отдельной монографии. Упомянем также об окрестностных грамматиках (некоторые косвенные сведения о них имеются в настоящей книге в § 4.5 и упражнении 5.7), Δ-грамматиках (также вскользь затронутых в § 4.5), оценках сложности распознавания порождаемых грамматиками языков.

2) Весьма важное место отводится в книге сопоставлению грамматик с автоматами; это дает возможность представлять себе грамматики и классы грамматик не изолированно, а в некотором достаточно широком контексте и благодаря этому лучше понимать строение порождаемых грамматиками языков и работу самих грамматик. Поэтому в главах 1, 3, 4 и 5 рассматриваются различные классы автоматов (собственно — машин Тьюринга) и изучаются их связи с классами грамматик.

Другой вопрос, значение которого в книге сознательно подчеркивается, — оценки сложности работы грамматик (т. е. вывода в грамматиках). Эти оценки являются важнейшим средством классификации грамматик и языков; именно они позволяют составить представление о том, как грамматика «работает». Поэтому вопрос об оценках сложности вывода в произвольных грамматиках рассматривается в начале книги — он составляет содержание второй главы; оценкам сложности вывода в грамматиках специальных типов посвящена основная часть третьей главы и вся седьмая.

3) Наряду с формальными грамматиками существуют и другие математические конструкции, специально предназначаемые для описания некоторых существенных аспектов естественного языка или для описания лингвистического исследования. Вместе с теорией грамматик теории этих конструкций составляют так называемую математическую лингвистику. Подробнее о значении этого термина см. [Гладкий — Мельчук 1969, стр. 15—22]. Для нас сейчас важно лишь то, что теория грамматик — это только часть математической лингвистики, хотя и занимающая в ней центральное место.

Настоящая книга посвящена только теории грамматик; однако автор счел целесообразным изложить в ней в виде приложений два специальных вопроса математической лингвистики, собственно к теории грамматик не относящихся: способы описания синтаксической структуры предложения и «теорию замещаемости». Это вызвано, с одной стороны, тем, что некоторыми возникающими при изучении указанных вопросов понятиями (впрочем, весьма простыми) приходится пользоваться в теории грамматик, с другой — наличием тесной связи между лингвистическими интерпретациями теории грамматик и этих вопросов. Кроме того, систематического и достаточно строгого изложения способов описания синтаксической структуры предложения до сих пор вообще, кажется, не появлялось в доступной широкому читателю литературе *).

По стилю оба приложения несколько отличаются от основного текста книги — в них уделено сравнительно много внимания лингвистическим приложениям. Это обусловлено самим характером материала, состоящего в значительной степени из определений понятий, вызванных к жизни не столько математическими, сколько лингвистическими соображениями.

Для чтения приложения I достаточно знакомства с § 1.1 (или даже с его началом). Знакомство с материалом § III.1 необходимо для чтения глав 3—7; остальные два параграфа приложения I используются только в § 6.3 (не считая одного несущественного замечания в § 6.1). Для чтения приложения II достаточно знакомства с §§ 1.1 и 1.2; материал этого приложения, именно, его второго параграфа, использован только в §§ 5.2 и 8.4 **).

4) В настоящей книге слово «язык» всюду означает «множество последовательностей элементарных символов», что в лингвистической интерпретации могло бы означать «множество текстов», в частном случае «множество правильных текстов». Такое словоупотребление является обычным в математической лингвистике (в последнее время оно проникло и в теорию автоматов), но расходится с принятым в собственно лингвистике (языковедении), где под языком понимают, как правило, некоторую с и с т е м у, «работающее устройство», включающее в себя грамматику *), что же касается совокупности текстов на данном языке, т. е. результата его работы, то эту совокупность лингвисты называют обычно «речью» (такое употребление терминов «язык» и «речь» восходит к Ф. де Соссюру). Однако использование слова «язык» в смысле «множество последовательностей символов» настолько уже укоренилось в теории грамматик и других разделах математической лингвистики, что отказаться от него не представляется возможным, тем более, что не видно, чем его заменить (во всяком случае, лингвистический термин «речь» здесь не годится).

5) Формальные грамматики могут использоваться для описания различных уровней естественного языка, например фонологического, морфологического или синтаксического. В любом случае задача грамматики состоит в том, чтобы описать, как строятся некоторые более сложные объекты из более простых: для фонологического уровня — морфы **) из фонем, для морфологического — слова (точнее словоформы, см. ниже) из морф, для синтаксического — предложения из слов. Поскольку в нашей книге лингвистическое истолкование привлекается только в иллюстративных целях, мы будем обращаться лишь к одной «естественно-языковой» интерпретации — синтаксической, во многих отношениях самой удобной. Таким образом, элементарные символы во всех лингвистических примерах будут интерпретироваться как слова (а «правильные последовательности»

*) Очень хорошее «нематематическое» изложение имеется в статье [Падучева 1964].

**) Читателю, интересующемуся лингвистическими вопросами, можно рекомендовать прочесть оба приложения сразу после гл. 1. Прочие читатели могут ознакомиться с § III. 1 перед чтением гл. 3, с §§III. 2, III. 3 перед чтением гл. 6, а из приложения II посмотреть только определение взаимозамещаемости и лемму III. 1, дойдя до теоремы 5.7, и определение конфигурации, дойдя до стр. 277.

*) Так же обычно понимается слово «язык» и в математике; например, «языком узкого исчисления предикатов» называют не множество правильно построенных формул этого исчисления, а совокупность правил образования таких формул (а иногда и правил преобразования).

**) Морфами называются наименьшие осмысленные единицы языка (корни, суффиксы и т. п.).

элементарных символов — как грамматически правильные предложения *)). Однако понятие «слово» весьма расплывчато и нуждается в уточнении. В лингвистике такое уточнение производится несколькими различными способами; это приводит к нескольким различным понятиям, из которых для наших целей следует отметить два: а) сегмент — единица внешней стороны языка; для письменной формы языка сегмент можно понимать просто как последовательность букв от пробела до пробела; б) словоформа — единица языка, рассматриваемая одновременно в плане выражения и в плане содержания; она может пониматься как сегмент, снабженный совокупностью значений, одно из которых — лексическое (это приблизительно то же самое, что понимается под «значением слова» в толковых словарях), а остальные — грамматические («мужской род», «дательный падеж» и т. п.)**). Например, в предложении *Топи печь, пора пироги печь* два вхождения сегмента *печь*, отвечающие разным словоформам; каждое из предложений *Судно на море догнал* и *Судно весело бежит* содержит вхождение сегмента *судно*, и опять-таки этим вхождениям отвечают разные словоформы (в первом случае различаются как лексические, так и грамматические значения, во втором — только грамматические). Во всех лингвистических замечаниях и примерах, содержащихся в основном тексте и в приложении I, «слова» могут трактоваться либо как сегменты, либо как словоформы, но всякий раз нужно четко представлять себе различие между этими концепциями. Иная интерпретация понадобится нам лишь в приложении II.

*) Для естественных языков грамматическую правильность следует отличать от осмысленности. Так, предложение *Ехала деревня мимо ямщика* бессмысленно, но грамматически правильно. Для простых формализованных языков объемы понятий грамматической правильности и осмысленности обычно совпадают; например, всякое арифметическое выражение, правильным образом составленное из символов натуральных чисел, знаков сложения и умножения и скобок, имеет смысл, т. е. представляет некоторое натуральное число.

**) В трактовке понятий сегмента и словоформы мы следуем А. А. Зализняку [Зализняк 1967, стр. 19—20].

ГЛАВА 1

ОСНОВНЫЕ ПОНЯТИЯ

§ 1.0. Некоторые предварительные пояснения

Поясним некоторые термины и обозначения — общематематические и из теории графов, — употребляемые в литературе не всегда одинаковым образом.

Отношение «быть подмножеством» обозначается символом $\subseteq$. Запись $A \subset B$ означает $A \subseteq B \,\&\, A \neq B$.

Пустое множество обозначается $\varnothing$.

Множество всевозможных упорядоченных пар элементов множества M обозначается M^2. Множество всех подмножеств M обозначается 2^M.

Мощность множества M обозначается $\mu(M)$.

Операции алгебры логики и кванторы обозначаются $\&$, $\vee$, $\neg$, $\supset$, $\equiv$, $\forall$, $\exists$ соответственно.

Если ρ — отношение эквивалентности на множестве M, то множество классов, на которые ρ разбивает M, называется фактор-множеством множества M по отношению ρ и обозначается M/ρ. Мощность M/ρ называется индексом отношения ρ.

Множество M с заданными на нем предикатами $P_1, \ldots, P_n$ обозначается $\langle M; P_1, \ldots, P_n\rangle$. Во всех остальных случаях упорядоченная система объектов $\mathfrak{A}_1, \ldots, \mathfrak{A}_k$ обозначается альтернативно через $(\mathfrak{A}_1, \ldots, \mathfrak{A}_k)$ или $\langle \mathfrak{A}_1, \ldots, \mathfrak{A}_k\rangle$.

Множество M с заданным на нем двуместным предикатом (бинарным отношением) R называется графом и обозначается, в соответствии с вышесказанным, $\langle M; R\rangle$. Вместо $R(a, b)$ пишем aRb. Элементы множества M

называются узлами графа $\langle M; R\rangle$ *). Пара узлов (a, b), для которой имеет место aRb, называется дугой графа; a есть начало этой дуги, b — ее конец. Вместо «пара (a, b) является дугой» говорим «из a в b идет дуга» или «a подчиняет b». Последовательность узлов графа $a_0, a_1, \ldots, a_n$ $(n \geqslant 0)$ называется путем (идущим) из a_0 в a_n в этом графе, если для каждого $i = 1, \ldots, n$ из a_{i-1} в a_i идет дуга; a_0 есть начало пути, a_n — его конец; число n есть длина пути. Если из a в b идет путь, говорим, что b зависит от a. (В частности, всякий узел зависит от себя.)

Узлы графа, не являющиеся началами никаких дуг, называются висячими, узлы, не являющиеся ни началами, ни концами никаких дуг, — изолированными.

Конечный граф называется деревом, если: а) в нем существует единственный узел (называемый корнем), который не является концом никакой дуги; б) всякий его узел, отличный от корня, является концом точно одной дуги; в) в нем нет замкнутых путей (т. е. путей, концы которых совпадают с началами) ненулевой длины. Легко видеть, что в каждый узел дерева идет в точности один путь из корня.

Множество всех узлов дерева, подчиненных одному узлу a, называется кустом этого узла, а множество всех узлов, зависящих от a, — группой зависимости узла a. Число элементов куста называется его шириной. Наибольшее значение ширины куста узла дерева называется шириной дерева.

Дерево называется бинарным, если ширина куста каждого его невисячего узла равна двум.

Высотой, рангом и степенью узла дерева называются соответственно длина пути из корня в данный узел, наибольшая длина пути из данного узла в висячий и наименьшая длина пути из данного узла в висячий. Наибольшее значение высоты узла дерева (совпадающее, очевидно, с наибольшим значением ранга и с ранг

*) Этот термин обычен в лингвистических работах и часто встречается в работах по математической лингвистике. В теории графов более принят термин вершина; мы избегаем его потому, что во многих математико-лингвистических работах «вершиной дерева» называют корень (см. ниже).

гом корня) называется высотой дерева. Наибольшее значение степени узла дерева называется степенью дерева.

Дерево, состоящее из одного лишь корня, называется единичным.

Множество M с заданными на нем двумя бинарными отношениями R_1 и R_2 называется графом (и обозначается $\langle M; R_1, R_2\rangle$). Нагруженным графом называется упорядоченная система $\langle M; R_1, R_2, U, g\rangle$, где $\langle M; R_1, R_2\rangle$ — граф, U — конечное множество и g — отображение M в U.

Упорядоченная система $\langle M; R, \varphi\rangle$, где M и R — множества и φ — отображение R в множество всевозможных упорядоченных пар элементов M, называется мультиграфом. Элементы M называются узлами мультиграфа, элементы R — его дугами. Если $(a, b) = \varphi(\alpha)$, говорим, что дуга α идет из a в b; a есть начало α; b есть конец α. Путь в мультиграфе определяется так же, как в графе. (Граф можно было бы определить как мультиграф с разнозначным φ.)

§ 1.1. Цепочки и языки

Мы будем рассматривать непустое конечное множество V, которое будем называть словарем (или алфавитом), а его элементы — элементарными символами (или просто символами, а также буквами). Произвольную конечную последовательность элементов V, т. е. отображение в V некоторого (конечного) начального отрезка натурального ряда, будем называть цепочкой *) в словаре V. Отображаемый отрезок натурального ряда может быть и пустым; тогда мы говорим о пустой цепочке. Непустую цепочку мы, как правило, будем записывать в виде $\omega = b_1 \ldots b_n$, где b_i — образ числа i. Пустую цепочку будем обозначать символом Λ. Мощность отображаемого отрезка натурального ряда будет называться длиной цепочки; в частности, длина пустой цепочки равна нулю. Длина цепочки ω обозначается $|\omega|$.

*) Мы не пользуемся более распространенным в математических работах термином слово во избежание ненужной омонимии в лингвистических приложениях.

Например, *алкилалкоксисилан* — цепочка длины 1 в словаре V, состоящем из 32 русских букв, а такж в словаре $V' = \{а, и, к, л, н, о, с\}$ и в любом словаре, содержащем V'. Иногда мы будем изображать цепочку

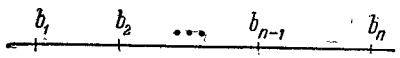


Рис. 1.

$b_1 \ldots b_n$ на отрезке горизонтальной прямой так, как по казано на рис. 1.

Множество всех цепочек в словаре V будет обозначаться через V^*, множество всех непустых цепочек в V — через V^+.

Конкатенацией (или результатом конкатенации) непустых цепочек $b_1 \ldots b_n$ и $c_1 \ldots c_p$ называется цепочка $d_1 \ldots d_{n+p}$, где $d_1 = b_1, \ldots, d_n = b_n, d_{n+1} = c_1, \ldots, d_{n+p} = c_p$. Кроме того, конкатенация цепочек ω и Λ, равно как и Λ и ω, где ω — произвольная цепочка, считается по определению равной ω. Конкатенацию цепочек ω и φ мы будем обозначать $\omega\varphi$.

Заметим, что слово «конкатенация» обозначает у нас и операцию и ее результат. Иногда эту операцию называют также умножением, а ее результат — произведением цепочек.

Операция конкатенации, очевидно, ассоциативна, но не коммутативна *).

Вместо $\underbrace{\omega\omega \ldots \omega}_{n \text{ раз}}$ мы будем обычно писать ω^n; ω^1 будет означать ω; ω^0 будет означать Λ.

Если $\omega = \varphi\psi$, мы будем называть φ началом ω и ψ концом ω. В частности, всякая цепочка есть начало и конец самой себя.

Мы введем также две частичные операции, обратные для конкатенации: левое деление и правое деление. Именно, левым (правым) частным от деления цепочки ω на цепочку φ называется цепочка ψ такая, что $\varphi\psi = \omega$ (соответственно $\psi\varphi = \omega$). Левое частное от деления ω на φ будет обозначаться через $\varphi \backslash \omega$, правое — через ω / φ.

*) За исключением случая, когда V состоит из одного элемента.

Например, цепочка *рококо* есть конкатенация цепочек *рок* и *око*, цепочка *окорок* — конкатенация цепочек *око* и *рок*; *рок* = *око* $\backslash$ *окорок* = *рококо* / *око*; *око* = *рок* $\backslash$ $\backslash$ *рококо* = *окорок* / *рок*; частных *рок* $\backslash$ *окорок*, *око* $\backslash$ $\backslash$ *рококо*, *окорок* / *око*, *рококо* / *рок* не существует.

Если для каких-либо цепочек ω, φ, η_1, η_2 в словаре V имеет место равенство $\omega = \eta_1\varphi\eta_2$, мы будем называть цепочку $\eta_1 * \varphi * \eta_2$, где символ $*$ не принадлежит V, вхождением цепочки φ в цепочку ω. Если существует вхождение φ в ω, мы будем называть φ подцепочкой ω. В случае, когда длина цепочки φ равна 1, т. е. φ состоит из одного символа b, будем говорить о вхождении символа b в цепочку ω. Вхождения символов в цепочку мы нередко будем называть ее точками.

Число вхождений символа α в цепочку ω будет обозначаться $|\omega|_\alpha$.

Если $\alpha = \eta_1 * b * \eta_2$ и $\beta = \xi_1 * c * \xi_2$ — точки одной и той же цепочки $\omega = \eta_1 b \eta_2 = \xi_1 c \xi_2$, и если при этом $|\eta_1| < |\xi_1|$, мы будем писать $\alpha < \beta$ или $\beta > \alpha$ и говорить, что α лежит (или расположена) левее β, а β — правее α. Если $\alpha < \beta < \gamma$ или $\gamma < \beta < \alpha$, говорим, что β лежит (или расположена) между α и γ. Для любых двух точек α, β цепочки ω таких, что $\alpha \leqslant \beta$, мы будем называть множество точек ξ, удовлетворяющих неравенствам $\alpha \leqslant \xi \leqslant \beta$, отрезком цепочки ω и обозначать $[\alpha, \beta]$, а множество точек, удовлетворяющих неравенствам $\alpha < \xi < \beta$, — интервалом цепочки ω и обозначать (α, β). Иногда нам будут нужны также несобственные интервалы $(-, \alpha)$ и $(\alpha, -)$ — множества точек, удовлетворяющих соответственно неравенствам $\xi < \alpha$ и $\alpha < \xi$. Интервал, в отличие от отрезка, может быть пуст. Между отрезками цепочки и вхождениями в нее непустых подцепочек имеется очевидное взаимно однозначное соответствие. Иногда для упрощения изложения отрезки будут отождествляться с соответствующими подцепочками (только в тех случаях, когда это не ведет к недоразумениям).

Пары точек α, β и γ, δ, по определению, разделяют друг друга, если одна из точек γ, δ лежит, а другая не лежит между α и β (или, что то же самое,

одна из точек α, β лежит, а другая не лежит между γ и δ).

Пример. Цепочка *рококо* содержит два вхождения цепочки *око*: $p * око * ко$ и $рок * око *$, одно вхождение символа p: $* p * окококо$, 7 вхождений пустой цепочки: $* * рококо$, $р * * окококо$ и т. д. Если обозначить вхождения символа $о$ (слева направо) через α, β, γ, то $\alpha < \beta < \gamma$; отрезкам $[\alpha, \beta]$ и $[\beta, \gamma]$ отвечают разные вхождения одной и той же подцепочки *око*.

Иногда мы будем фиксировать некоторый пересчет словаря V: $a_1, \ldots, a_k$ и связывать с этим пересчетом следующее отношение «предшествования» на V^*: более короткая цепочка предшествует более длинной, а для цепочек равной длины предшествование определяется лексикографически (т. е. цепочка $a_{i_1} \ldots a_{i_{s-1}} a_{i_s} \ldots a_{i_n}$ предшествует цепочке $a_{i_1} \ldots a_{i_{s-1}} a_{j_s} \ldots a_{j_n}$, если $i_s < j_s$). Это отношение является линейным порядком; более того, существует взаимно однозначное отображение ν: $V^* \xrightarrow{\text{на}} N$ ($N = \{0, 1, \ldots\}$), сохраняющее порядок (упражнение 1.3). Число $\nu(\omega)$ мы будем называть стандартным номером цепочки ω, а определенное только что отношение на V^* — стандартным порядком.

Произвольное множество цепочек в словаре V мы будем называть языком в этом словаре. Рассматривается, в частности, и пустой язык; обозначаться он будет, в согласии с § 1.0, символом $\varnothing$.

Над языками можно производить обычные теоретико-множественные операции — объединение (обозначается $\cup$), пересечение (обозначается $\cap$), вычитание (обозначается —), взятие дополнения (дополнение языка L до множества V^* будет обозначаться через $C_V L$ или, если недоразумение невозможно, через CL; дополнение до множества V^+ — соответственно через $C'_V L$ или $C'L$).

Введем теперь некоторые специфические операции над языками. Умножение (иначе, прямое умножение или конкатенация) определяется как операция, ставящая в соответствие двум языкам L_1 и L_2 язык $L = \{\varphi\psi \mid \varphi \in L_1, \psi \in L_2\}$. Этот язык обозначается $L_1 L_2$ и называется (прямым) произведением L_1 на L_2 (или конкатенацией L_1 и L_2). Для экономии скобок мы будем считать, что умножение связывает теснее, чем объединение, так что, например, $L_1 \cup L_2 L_3$ будет означать $L_1 \cup (L_2 L_3)$.

Операция умножения ассоциативна, но не коммутативна. Ввиду ее ассоциативности ясен смысл записи $L_1 \ldots L_n$. Кроме того, она дистрибутивна относительно объединения: $L(L_1 \cup L_2) = LL_1 \cup LL_2$; $(L_1 \cup L_2)L = L_1 L \cup L_2 L$ (упражнение 1.4, а)).

Итерацией (или результатом итерации) языка L называется язык $\bigcup_{n=0}^{\infty} L^n$, где $L^0 = \{\Lambda\}$, $L^1 = L$ и $L^n = \underbrace{L \ldots L}_{n \text{ раз}}$ при $n > 1$. Этот язык обозначается L^*. Иногда нам будет удобно рассматривать также усеченную итерацию языка L, определяемую как $\bigcup_{n=1}^{\infty} L^n$; она будет обозначаться L^+.

Операции левого и правого деления языка на цепочку определяются и обозначаются соответственно следующим образом:

$$\omega \backslash L = \{\varphi \mid \omega\varphi \in L\}; \quad L / \omega = \{\varphi \mid \varphi\omega \in L\}.$$

Наконец, подстановка языков $L_1, \ldots, L_n$ в язык L вместо символов $a_1, \ldots, a_n$ есть операция, сопоставляющая языку L в словаре $V = \{a_1, \ldots, a_n\}$ и языкам $L_1, \ldots, L_n$ в словарях $V_1, \ldots, V_n$ соответственно следующий язык в словаре $V_1 \cup \ldots \cup V_n$:

$$\{\omega_{i_1} \ldots \omega_{i_k} \mid a_{i_1} \ldots a_{i_k} \in L, \ \omega_{i_1} \in L_{i_1}, \ldots, \omega_{i_k} \in L_{i_k}\} \cup L',$$

где $L' = \{\Lambda\}$, если $\Lambda \in L$, и $L' = \varnothing$, если $\Lambda \notin L$. Этот язык будет обозначаться $S(L; a_1, \ldots, a_n \mid L_1, \ldots, L_n)$. В частном случае, когда языки $L_1, \ldots, L_n$ одноэлементны, подстановка называется гомоморфным отображением, или гомоморфизмом, языка L. Если L — язык в словаре $V = \{a_1, \ldots, a_n\}$ и $U \subseteq V$, то гомоморфное отображение $S(L; a_1, \ldots, a_n \mid L'_1, \ldots, L'_n)$, где L'_i есть $\{a_i\}$ при $a_i \in U$ и $\{\Lambda\}$ при $a_i \notin U$, называется проектированием L на U, а его результат —

проекцией L на U. Образ цепочки при проектировании также называется ее проекцией. Проекцию цепочки ω и языка L на словарь U мы будем обозначать соответственно $\mathrm{Пр}_U\omega$ и $\mathrm{Пр}_U L$.

Если $\{\omega\}$ — язык, состоящий из одной цепочки ω, то мы будем вместо $L\{\omega\}$, $\{\omega\}L$, $\{\omega\}^*$ и $\{\omega\}^+$ писать соответственно $L\omega$, ωL, ω^* и ω^+.

Пример. Если $V = \{0, 1\}$, L — язык, состоящий из всевозможных двоичных записей целых положительных чисел (т. е. цепочек, начинающихся символом 1), и L_1 — язык, состоящий из всевозможных двоичных записей нечетных положительных чисел (т. е. цепочек, начинающихся и оканчивающихся символом 1), то $L = 1V^*$, $L_1 = \{1\} \cup 1V^*1$, $LL_1 = L1 = L_1 - \{1\}$, $L_1L = L \cap \cap (V^*11V^*)$; $L^+ = L$; $\omega \backslash L = V^*$, если ω начинается единицей; $\omega \backslash L = \varnothing$, если ω начинается нулем; $L_1/\omega = L_1 \cup \{\Lambda\}$, если ω оканчивается единицей; $L_1/\omega = \varnothing$ если ω оканчивается нулем; $S(L_1; 0, 1 | L_1, L) = L$.

Пусть $\xi_1, \ldots, \xi_k$ — абстрактные символы и $B_1, \ldots, B_p$ — языки (точнее, символы, обозначающие языки). Выражение, составленное из $\xi_1, \ldots, \xi_k, B_1, \ldots, B_p$ с помощью знаков объединения и умножения *), соответственно знаков объединения, умножения и итерации, называется многочленом, соответственно регулярным выражением от $\xi_1, \ldots, \xi_k$ с коэффициентами $B_1, \ldots, B_p$. Если интерпретировать $\xi_1, \ldots, \xi_k$ как переменные, пробегающие какие-либо множества языков, то многочлен (регулярное выражение) становится функцией от этих переменных; значение этой функции, получаемое при подстановке на место переменных $\xi_1, \ldots, \xi_k$ конкретных языков $L_1, \ldots, L_k$, есть также некоторый язык; мы будем говорить, что этот язык представляется (или задается) данным многочленом (регулярным выражением) при $\xi_1 = L_1, \ldots, \xi_k = L_k$. Например, язык, состоящий из всевозможных цепочек в словаре $\{a, b\}$, содержащих вхождение фиксированной цепочки ω, задается регулярным выражением $\Phi(\xi) = \{a, b\}^*\xi\{a, b\}^*$ при $\xi = \{\omega\}$.

Многочлен (регулярное выражение), не содержащий (не содержащее) переменных, называется замкнутым; замкнутый многочлен (регулярное выражение) представляет единственный язык.

*) Знаком умножения можно считать пробел между буквами.

Два многочлена (регулярных выражения) от переменных $\xi_1, \ldots, \xi_k$ тождественны (или тождественно равны), если они отвечают одной и той же функции, т. е. при любой подстановке языков вместо переменных представляют один и тот же язык.

В силу упомянутого выше дистрибутивного закона любой многочлен от $\xi_1, \ldots, \xi_k$ тождественно равен некоторому многочлену вида $\bigcup_{j=1}^{s} \alpha_{j1} \ldots \alpha_{jr_j}$, где $\alpha_{j1}, \ldots, \alpha_{jr_j}$ — переменные или коэффициенты.

§ 1.2. Грамматики

В самом широком смысле формальными грамматиками, или просто грамматиками, называют любые «автоматические устройства» (т. е. исчисления или алгоритмы), позволяющие «задавать» языки. При этом «задание» может осуществляться по-разному: оно означает либо возможность для каждой цепочки данного языка подобрать такой режим работы устройства, чтобы к концу работы получить («породить») эту цепочку (причем, разумеется, ни одна цепочка, не принадлежащая языку, не должна «порождаться»), либо возможность «перечислить» язык, т. е. организовать работу устройства так, чтобы оно выдавало цепочки языка одну за другой и могло бы выдать любую из них, работая достаточно долго, либо, наконец, возможность для произвольной цепочки (в соответствующем словаре) получить от устройства ответ на вопрос, принадлежит ли эта цепочка языку. Из трех указанных подходов мы выберем первый. Как станет нам ясно впоследствии (а читателю, достаточно хорошо знающему теорию алгоритмов, должно быть ясно уже сейчас), это не сужает класса описываемых языков — любой язык, допускающий задание первым способом, допускает также задание вторым или третьим. В то же время именно такой подход лучше всего моделирует ситуацию, имеющую место при пользовании языком (естественным или

искусственным) *) — основная задача там состоит в порождении предложения, обладающего заданным смыслом. Правда, эта ситуация моделируется далеко не в полной мере — в модели происходит порождение не предложений с заданным смыслом, а любых правильных предложений (понятие смысла в этой модели вообще не присутствует в сколько-нибудь отчетливой форме). Тем не менее такая модель позволяет значительно приблизиться к пониманию того, как смысл преобразуется в текст, поскольку преобразования, с помощью которых в ней происходит процесс порождения правильного предложения, могут рассматриваться как прообраз — хотя и достаточно грубый — тех преобразований, с помощью которых осуществляется переход от смысла к тексту.

Итак, мы останавливаемся на понимании грамматики как «устройства для порождения цепочек» и впредь будем говорить о «порождающих грамматиках». Но порождение цепочек может быть организовано различными способами. В частности, возможны такие способы, при которых все возникающие в процессе порождения промежуточные объекты будут тоже цепочками, и такие, при которых эти объекты могут иметь иную природу. Нас здесь будут интересовать исключительно способы первого типа **); таких способов тоже, как легко понять, может быть много, и нам придется выбрать какой-то один (или в крайнем случае несколько). Мы выберем способ, предложенный Н. Хомским (впервые в [Chomsky 1956, 1957]) и не уступающий по силе, как мы увидим в § 1.4, никакому другому возможному эффективному способу ***); в то же время он обладает тем

*) В этой и предыдущей фразах слово «язык» употреблено в разных значениях (см. введение, стр. 14—15).

**) Для более точного моделирования перехода от смысла к тексту более адекватны как раз такие грамматики, в которых промежуточные объекты имеют более сложную природу, например являются деревьями (см. [Гладкий — Мельчук 1971]). Однако теория таких грамматик еще недостаточно разработана; во всяком случае, — это для нас здесь наиболее важно — теория «древесных» грамматик не может не опираться существенным образом на теорию проще устроенных «цепочечных» грамматик, которые и являются предметом настоящей книги.

***) Разумеется, это утверждение не носит характера математической теоремы. Его статус будет разъяснен в конце § 1.4.

преимуществом, что по крайней мере для наиболее важных частных случаев позволяет сопоставлять порождаемым предложениям весьма естественные описания их структуры (см. ниже, § 3.1). Еще одно весьма важное достоинство данного способа состоит в том, что он хорошо вписывается в некоторую более общую систему понятий математической логики, теории алгоритмов и теории автоматов *).

Итак, мы переходим к определению порождающих грамматик в смысле Н. Хомского; до тех пор, пока не будут введены в рассмотрение другие типы грамматик, — а по большей части и после этого (когда невозможно недоразумение) — мы будем в качестве синонима для выражения «порождающая грамматика» употреблять просто слово «грамматика».

(Порождающая) грамматика — это упорядоченная четверка $\Gamma = \langle V, W, I, R \rangle$, где: 1) V и 2) W — непересекающиеся непустые конечные множества; 3) I — некоторый элемент W; 4) R — конечное множество цепочек вида $\varphi \rightarrow \psi$, где φ и ψ — различные цепочки в словаре $V \cup W$ и $\rightarrow$ — символ, не входящий в $V \cup W$. Множества V и W называются соответственно основным и вспомогательным словарями (алфавитами) грамматики Γ, а их элементы — соответственно основными и вспомогательными символами Γ **); I называется начальным символом Γ, R — схемой Γ и элементы R — правилами Γ. Цепочки φ и ψ называются соответственно левой и правой частями правила $\varphi \rightarrow \psi$. Объединение $V \cup W$ мы будем иногда называть полным словарем грамматики Γ.

Пусть $r = \varphi \rightarrow \psi$ — правило грамматики Γ и $\xi_1 * \varphi * \xi_2$ — вхождение φ в цепочку $\omega = \xi_1 \varphi \xi_2$ в словаре $V \cup W$. В этом случае мы будем говорить, что цепочка

*) Заметим, что эта система понятий возникла еще до исследований Н. Хомского, и то обстоятельство, что его концепция сразу нашла готовую формальную базу, очень сильно способствовало успешному развитию этой концепции. Таким образом, и здесь теория опередила потребности приложений, как бывает чаще всего, вопреки распространенному среди «широкой публики» ходячему мнению.

) Вместо слов «основной» и «вспомогательный» нередко употребляются соответственно слова **терминальный и **нетерминальный**.

$\eta = \xi_1\psi\xi_2$ получается из ω применением правила r к вхождению $\xi_1 * \varphi * \xi_2$ цепочки φ. Если цепочка η получается из цепочки ω применением какого-либо правила Г, будем говорить, что η непосредственно выводима из ω в Г, и писать $\omega \models_\Gamma \eta$ или просто $\omega \models \eta$.

Последовательность цепочек $D = (\omega_0, \omega_1, \ldots, \omega_n)$ $(n \geqslant 1)$ называется выводом ω_n из ω_0 в грамматике Г, если для каждого i, $1 \leqslant i \leqslant n$, имеет место $\omega_{i-1} \models \omega_i$. Число n называется длиной вывода D. Если существует вывод η из ω в Г, то мы будем говорить, что η выводима из ω в Г, и писать $\omega \vdash_\Gamma \eta$ или $\omega \vdash \eta$.

Вывод $(\omega_0, \omega_1, \ldots, \omega_n)$ называется полным, если $\omega_0 = I$ и ω_n — цепочка в словаре V.

Если $D = (\omega_0, \omega_1, \ldots, \omega_n)$ — вывод в грамматике Г и для некоторого $i = 1, \ldots, n$ цепочка ω_i получается из ω_{i-1} применением правила $\varphi \to \psi$ к вхождению $\xi_i * \varphi * \eta_i$ цепочки φ в ω_{i-1}, то мы будем говорить, что это правило применяется на i-м шаге вывода D к вхождению $\xi_i * \varphi * \eta_i$ и что данное вхождение φ в ω заменяется на i-м шаге вывода D. Размеченным выводом, соответствующим выводу D, мы будем называть последовательность $(\omega_0', \omega_1', \ldots, \omega_{n-1}', \omega_n)$, где ω_i' $(i = 0, \ldots, n-1)$ — вхождение, заменяемое на i-м шаге вывода D. Одному выводу может, вообще говоря, соответствовать много размеченных выводов (см. упражнение 1.9).

Множество цепочек в основном словаре грамматики Г, выводимых из ее начального символа (иначе — множество последних цепочек всевозможных полных выводов в Г), называется языком, порождаемым грамматикой Г, и обозначается $L(\Gamma)$.

Из сформулированных определений видно, что существенным свойством процесса работы порождающей грамматики является его недетерминированность: если оборвать вывод на каком-либо шаге, то продолжение, вообще говоря, не восстанавливается однозначно по оставшейся части и может быть выполнено разными способами. Таким образом, грамматика — это исчисление, т. е. разрешение производить некоторые операции [Марков 1954, стр. 203] — в данном случае подстановки

в цепочки одних подцепочек вместо других (а не алгоритм, т. е. не предписание производить какие-то операции).

Пример. Пусть $\Gamma = \langle \{a, b, c\}, \{A, B, C, D\}, A, \{A \to BCA, BCB \to D, BC \to bc, DC \to a, cA \to c, aA \to a\}\rangle$. Пример полного вывода в Г: $(A, BCA, BCBCA, BCBCBCA, BCDCA, BCDCBCA, BCDCbcA, BCabcA, bcabcA, bcabc)$. Соответствующий размеченный вывод (в данном случае единственный): $(*A*, BC*A*, BCBC*A*, BC*BCB*CA, BCDC*A*, BCDC*BC*A, BC*DC*bcA, *BC*abcA, bcab*cA*, bcabc)$. Легко видеть, что $L(\Gamma) = \{a, bc\}^+$.

Грамматика $\Gamma = \langle V, W, I, R\rangle$ называется грамматикой составляющих, или НС-грамматикой*), если каждое ее правило имеет вид $\xi_1 A \xi_2 \to \xi_1 \theta \xi_2$, где ξ_1, ξ_2 — произвольные цепочки в словаре $V \cup W$, $A \in W$ и θ — произвольная непустая цепочка в $V \cup W$.

Правила вида $\xi_1 A \xi_2 \to \xi_1 \theta \xi_2$, где ξ_1, ξ_2, A и θ имеют указанный только что смысл, иногда называют НС-правилами; цепочка ξ_1, цепочка ξ_2 и пара цепочек (ξ_1, ξ_2) называются соответственно левым контекстом, правым контекстом и контекстом НС-правила $\xi_1 A \xi_2 \to \xi_1 \theta \xi_2$. При применении НС-правила фактически заменяется только одно вхождение символа A, но возможность замены зависит от наличия нужного контекста. Если $\xi_1 = \xi_2 = \Lambda$, то правило называется бесконтекстным (или контекстно-свободным), сокращенно Б-правилом (или КС-правилом).

НС-грамматика называется бесконтекстной (или контекстно-свободной), сокращенно Б-грамматикой (или КС-грамматикой), если все ее правила бесконтекстные.

(Б-)грамматика называется автоматной, сокращенно А-грамматикой**), если каждое ее правило

*) Вместо «грамматика составляющих» часто говорят «грамматика непосредственно (или непосредственных) составляющих»; отсюда и сокращение «НС-грамматика». Мы опускаем слово «непосредственно», так как никаких других составляющих не бывает (ср. [Гладкий — Мельчук 1969], стр. 185), но сохраняем сокращение, ставшее уже общеупотребительным.

**) Иногда используется термин «грамматика с конечным числом состояний», по нашему мнению, неудачный (см. [Гладкий — Мельчук

имеет вид $A \to aB$ или $A \to a$, где A, B — вспомогательные символы и a — основной символ.

Языки, порождаемые НС-, Б- и А-грамматиками, называются соответственно НС-, Б- и А-языками.

Классы произвольных грамматик, НС-, Б- и А-грамматик мы будем иногда обозначать соответственно через Г (так же мы часто обозначаем конкретные грамматики, однако эта омонимия всегда устраняется контекстом), НС, Б и А. Кроме того, для произвольного класса грамматик $\mathscr{T}$ через $\mathscr{L}(\mathscr{T})$ будет обозначаться класс языков, порождаемых грамматиками класса $\mathscr{T}$. Примеры НС-, Б- и А-грамматик будут приведены в § 1.3.

Введем еще некоторые понятия и отметим несколько полезных для дальнейшего простых фактов.

Правило $\varphi \to \psi$ грамматики Г называется заключительным, если $\psi \neq \Lambda$ и никакой символ, входящий в ψ не может входить в левую часть какого-либо правила Г.

Перестройкой вывода $D = (\omega_0, \omega_1, \ldots, \omega_n)$ в грамматике Г мы будем называть (вообще говоря многозначную) операцию, сопоставляющую выводу D какой-либо другой вывод цепочки ω_n из ω_0 в Г, в котором применяются в точности те же правила, что и в D, но в другом порядке.

Лемма 1.1. *Всякий вывод в произвольной грамматике можно перестроить так, чтобы ни одно заключительное правило не применялось раньше какого-либ незаключительного.* Доказательство очевидно.

Пусть $(\omega'_0, \omega'_1, \ldots, \omega'_{n-1}, \omega_n)$, где $\omega'_i = \xi_i * \varphi_i * \eta_i$ $(0 \leqslant i < n)$, — размеченный вывод в некоторой грамматике, и пусть правило, применяемое на i-м шаг $(i = 1, \ldots, n)$, есть $\varphi_{i-1} \to \psi_{i-1}$. Тогда для каждого $i = 1, \ldots, n-1$ имеет место равенство $\xi_i\varphi_i\eta_i = \xi_{i-1}\psi_{i-1}\eta_{i-1}$; при этом выделенное вхождение φ_i может находиться либо целиком левее выделенного вхождения ψ_{i-1}, либо целиком правее, либо пересекаться с ним. В этих случаях мы будем говорить соответственно, что $i+1$-й шаг данного размеченного вывода происходит левее i-го, происходит правее i-го, зацеплен с i-м.

Назовем размеченный вывод упорядоченным, если никакой следующий шаг не происходит в нем левее предыдущего. Вывод, становящийся при подходящей разметке упорядоченным, назовем упорядочиваемым.

Лемма 1.2. *Всякий вывод в произвольной грамматике можно перестроить в упорядочиваемый.*

Доказательство. Пусть $D' = (\omega'_0, \ldots, \omega'_{n-1}, \omega_n)$, где $\omega'_i = \xi_i * \varphi_i * \eta_i$ $(0 \leqslant i < n)$, — неупорядоченный размеченный вывод в грамматике Г и $i_0 = i_0(D')$ — наименьшее из чисел i, для которых $i+1$-й шаг в D' происходит левее i-го. Очевидно, $|\xi_{i_0-1}| \geqslant |\xi_{i_0}\varphi_{i_0}|$. Пусть j_0 — наименьшее число, обладающее тем свойством, что для каждого j, удовлетворяющего условию $j_0 \leqslant j \leqslant i_0$, выполняется неравенство $|\xi_{j-1}| \geqslant |\xi_{i_0}\varphi_{i_0}|$. Для каждого $j = j_0, j_0+1, \ldots, i_0$ имеем $\xi_{j-1}\varphi_{j-1}\eta_{j-1} = \xi_{i_0}\varphi_{i_0}\xi'_{j-1}\varphi_{j-1}\eta_{j-1}$. Преобразуем D' следующим образом: шаги до $j_0 - 1$-го включительно оставим без изменения, затем в полученной таким образом цепочке $\xi_{i_0}\varphi_{i_0}\xi'_{j_0-1}\varphi_{j_0-1}\eta_{j_0-1}$ применим к выделенному вхождению φ_{i_0} правило, применявшееся в D' на i_0+1-м шаге, затем будем производить все замены, как на шагах D' от j_0-го до i_0-го включительно, — после чего, очевидно, получится цепочка $\xi_{i_0}\varphi_{i_0}\eta_{i_0}$ — и, наконец, преобразуем эту цепочку в ω_n, как в D'. Ясно, что такое преобразование дает новый размеченный вывод D'' в Г, и если он не упорядочен, то $i_0(D'') > i_0(D')$. Повторяя это преобразование нужное число раз, получим упорядоченный размеченный вывод.

Два вывода в грамматике Г назовем равносильными, если их первые и последние цепочки соответственно совпадают.

Вывод, в котором никакая цепочка не встречается более одного раза, назовем бесповторным.

Лемма 1.3. *Для произвольного вывода в любой грамматике существует равносильный ему бесповторный вывод, который можно получить из исходного выбрасыванием некоторых цепочек.* Доказательство очевидно.

Грамматики Г и Г′ называются эквивалентными, если $L(\Gamma) = L(\Gamma')$.

1969], стр. 184). Лучше всего было бы, пожалуй, говорить «конечно-автоматная грамматика», поскольку А-грамматики тесно связаны с конечными автоматами (см. ниже, § 5.1), но мы не будем этого делать, чтобы не увеличивать и без того слишком большое число терминологических вариантов

Лемма 1.4. *Для произвольной грамматики* Γ *жет быть построена* *) *эквивалентная ей грамматика* Γ, *левые части правил которой не содержат вхождени основных символов. Если при этом* Γ — НС-*грамматик то и* Γ' *может быть сделана* НС-*грамматикой.*

Доказательство. Пусть $\Gamma = \langle V, W, I, R\rangle$. Сопоставим каждому символу $a \in V$ «двойника» — новы символ $\bar{a} \notin V \cup W$ — так, чтобы разным символам от вечали разные «двойники». Положим $\bar{V} = \{\bar{a} \mid a \in V\}$ $\Gamma' = \langle V, W \cup \bar{V}, I, \bar{R} \cup \tilde{R}\rangle$, где $\tilde{R} = \{\bar{a} \to a \mid a \in V\}$ и получается из R заменой в каждом правиле все вхождений основных символов вхождениями их «двойников». Легко видеть, что Γ' и есть нужная грамматика.

В грамматике, удовлетворяющей условию леммы 1.4, каждое правило, правая часть которого непуста и состоит из основных символов, является заключительным.

Пусть $\mathscr{T}$ — некоторый класс грамматик и φ — k-местная операция над языками. Допуская языковую вольность, будем говорить, что класс $\mathscr{L}(\mathscr{T})$ эффективно замкнут относительно φ, если по любым k грамматикам $\Gamma_1, \ldots, \Gamma_k \in \mathscr{T}$ можно построить такую грамматику $\Gamma \in \mathscr{T}$, что $L(\Gamma) = \varphi(L(\Gamma_1), \ldots, L(\Gamma_k))$ **).

Теорема 1.1. *Класс языков, порождаемых всевозможными грамматиками, эффективно замкнут относительно операций объединения, пересечения, умножения, итерации, усеченной итерации, левого и правого деления и подстановки.*

Доказательство. 1) Пусть $\Gamma' = \langle V', W', I', R'\rangle$ и $\Gamma'' = \langle V'', W'', I'', R''\rangle$ — произвольные грамматики. В силу леммы 1.4 мы можем считать, что левые части правил Γ' и Γ'' не содержат вхождений основных символов. Будем считать, кроме того, что $W' \cap W'' = \varnothing$; если это не так, переименуем символы, входящие в W'', что не изменит языка $L(\Gamma'')$. При выполнении этих условий очевидно, что грамматики $\langle V, W, I, R \cup R'' \cup \{I \to I', I \to I''\}\rangle$ и $\langle V, W, I, R' \cup R'' \cup \{I \to I'I''\}\rangle$, где $V = V' \cup V''$, $W = W' \cup W'' \cup \{I\}$, $I \notin V' \cup V'' \cup W' \cup W''$, порождают соответственно языки $L(\Gamma') \cup L(\Gamma'')$ и $L(\Gamma')L(\Gamma'')$. Чтобы построить грамматику, порождающую пересечение, сопоставим взаимно однозначным образом каждому символу $a \in V'$ новый символ $\bar{a}$ и каждому символу $b \in V''$ новый символ $\bar{\bar{b}}$. Положим $\bar{V}' = \{\bar{a} \mid a \in V'\}$, $\bar{\bar{V}}'' = \{\bar{\bar{b}} \mid b \in V''\}$ $(\bar{V}' \cap \bar{\bar{V}}'' = \varnothing)$. Обозначим через $\bar{R}'$, соответственно через $\bar{\bar{R}}''$, множество правил, полученное из R' (из R'') заменой каждого вхождения каждого основного символа a в каждое правило вхождением символа $\bar{a}$ ($\bar{\bar{a}}$). Пусть теперь $I \notin V' \cup V'' \cup \bar{V}' \cup \bar{\bar{V}}'' \cup W' \cup W''$ и

$$\Gamma = \langle V' \cap V'', W' \cup W'' \cup \bar{V}' \cup \bar{\bar{V}}'' \cup \{I\}, I, \bar{R}' \cup \bar{\bar{R}}'' \cup \{I \to I'I''\} \cup R_1 \cup R_2\rangle,$$

где $R_1 = \{\bar{a}\bar{\bar{b}} \to \bar{\bar{b}}\bar{a} \mid a \in V', b \in V''\}$, $R_2 = \{\bar{a}\bar{\bar{a}} \to a \mid a \in V' \cap V''\}$. Нетрудно видеть, что $L(\Gamma) = L(\Gamma') \cap L(\Gamma'')$.

2) Пусть $\Gamma = \langle V, W, I, R\rangle$ — произвольная грамматика. Как и выше, считаем, что левые части ее правил не содержат вхождений основных символов. Пусть x — произвольная цепочка из V^*. Положим $W' = W \cup \{I', \#\}$, где $I', \# \notin V \cup W$, $R' = \{I' \to \# I \#, \# \# \to \# I \#, \# \# \to \Lambda\} \cup \{\# a \to a \# \mid a \in V\}$, $R'_x = \{I' \to \# I, \# x \to \Lambda\}$. Легко видеть, что грамматики $\hat{\Gamma} = \langle V, W', I', R \cup R'\rangle$ и $\hat{\Gamma}_x = \langle V, W', I', R \cup R'_x\rangle$ порождают соответственно языки $L(\Gamma)^+$ и $x \backslash L(\Gamma)$. Чтобы получить язык $L(\Gamma)^*$, достаточно добавить к схеме грамматики Γ' правило $I' \to \Lambda$. Грамматика для $L(\Gamma)/x$ строится аналогично $\hat{\Gamma}_x$.

3) Доказательство для подстановки предоставляется читателю.

Замечание. Относительно операций вычитания и взятия дополнения рассматриваемый класс языков не замкнут (см. ниже замечание в конце § 1.4).

*) «Построить» будет на протяжении всей книги означать «эффективно построить». Здесь, например, «по произвольной грамматике Γ может быть построена Γ'» означает «существует алгоритм, строящий по произвольной грамматике Γ нужную Γ'»; в дальнейшем всюду аналогично.

**) Таким образом, эффективная замкнутость есть на самом деле свойство класса $\mathscr{T}$, а не класса $\mathscr{L}(\mathscr{T})$.

§ 1.3. Примеры грамматик

Во всех приводимых здесь и далее примерах грамматик мы будем, если не оговорено противное, обозначать основные символы строчными латинскими буквами,

вспомогательные — заглавными латинскими буквами (те и другие буквы могут иметь индексы и надстрочные знаки), начальный символ — буквой I. Это позволит нам ограничиться выписыванием схем грамматик.

Пример 1. а) Любой непустой конечный язык $\{\omega_1, \ldots, \omega_n\}$, $\omega_i \neq \Lambda$, порождается Б-грамматикой со схемой $\{I \to \omega_1, \ldots, I \to \omega_n\}$.

б) Тот же язык, если $\omega_i = a_{i_1} \ldots a_{ik_i}$, порождается А-грамматикой со схемой

$$\{A_{i0} \to a_{i1}A_{i1}, \ldots, A_{i,\,k_i-2} \to a_{i,\,k_i-1}A_{i,\,k_i-1}, \quad A_{i,\,k_i-1} \to a_{i,\,k_i} \mid i = 1, \ldots, n\},$$

где $A_{10} = \ldots = A_{n0} = I$.

в) Пустой язык порождается А-грамматикой со схемой $\{I \to aI\}$.

Таким образом, любой конечный язык, не содержащий Λ, является А-языком.

Пример 2. Для любого словаря V язык V^+ порождается А-грамматикой со схемой $\{I \to aI, I \to a \mid a \in V\}$.

Будем называть простейшей записью натурального числа n цепочку $\underbrace{|\ldots|}_{n \text{ раз}}$. Множество простейших записей элементов произвольного множества натуральных чисел M будет всегда отождествляться с самим множеством M.

Пример 3. а) Каковы бы ни были числа $k = 1, 2, \ldots$ и $l = 0, 1, \ldots, k-1$, множество $P_{kl} = \{kn + l \mid n = 0, 1, \ldots\}$ (арифметическая прогрессия) порождается А-грамматикой с основным словарем $\{|\}$, начальным символом A_0 и схемой $\{A_0 \to |A_1, \ldots, A_{k-2} \to |A_{k-1}, A_{k-1} \to |A_0, A_0 \to |B_1, B_1 \to |B_2, \ldots, B_{l-2} \to |B_{l-1}, B_{l-1} \to |\}$ (так при $l > 1$; при $l = 1$ последние l правил заменяются правилом $A_0 \to |$, при $l = 0$ — правилом $A_{k-1} \to |$).

б) По аналогии с примером 1, б) легко построить А-грамматику, порождающую объединение конечного числа арифметических прогрессий и конечного множества чисел.

Пример 4. Язык a^+b^+ порождается А-грамматикой со схемой $\{I \to aI, I \to aB, B \to bB, B \to b\}$.

Пример 5. Язык $\{a^nb^n \mid n = 1, 2, \ldots\}$ порождается Б-грамматикой со схемой $\{I \to aIb, I \to ab\}$.

Пример 6. Языки $\{a^nb^na^m \mid m, n = 1, 2, \ldots\}$ и $\{a^mb^na^n \mid m, n = 1, 2, \ldots\}$ порождаются Б-грамматиками со схемами $\{I \to Ia, I \to Aa, A \to aAb, A \to ab\}$ и $\{I \to aI, I \to aA, A \to bAa, A \to ba\}$ соответственно.

Пример 7. Множество всевозможных «правильных скобочных последовательностей» порождается Б-грамматикой с основным словарем (), {} и схемой $\{I \to II, I \to (I), I \to (\,)\}$.

Пример 8. Язык L в словаре $\{a, b\}$, состоящий из всех непустых цепочек, содержащих одинаковое число вхождений a и b, порождается Б-грамматикой со схемой $\{I \to II, I \to aIb, I \to bIa, I \to ab, I \to ba\}$.

Непосредственно ясно, что все цепочки в $\{a, b\}$, порождаемые данной грамматикой, принадлежат L. Обратное доказывается возвратной индукцией по длине цепочки; базис этой индукции тривиален, а индукционный шаг основывается на следующем очевидном факте: любая цепочка $\omega \in L$ длины, большей или равной 4, представима по крайней мере в одном из видов $a\xi b$, $b\eta a$, $\zeta_1\zeta_2$, где $\xi, \eta, \zeta_1, \zeta_2 \in L$.

Для произвольной цепочки ω в словаре $V = \{a_1, \ldots, a_k\}$ назовем ее обращением и обозначим $\hat{\omega}$ цепочку $a_{i_n} \ldots a_{i_1}$, если $\omega = a_{i_1} \ldots a_{i_n}$, и Λ, если $\omega = \Lambda$.

Пример 9. а) Язык $L_з = \{x\hat{x} \mid x \in V^+\}$ ($з$ — от «зеркальное отражение») порождается Б-грамматикой со схемой $\{I \to aIa, I \to aa \mid a \in V\}$.

б) Язык $C'L_з$ порождается Б-грамматикой со схемой:

(I)	$I \to aIb$	$(a, b \in V)$
(II)	$I \to a$	$(a \in V)$
(III)	$I \to aAb$	$(a, b \in V;\ a \neq b)$
(IV)	$I \to ab$	$(a, b \in V;\ a \neq b)$
(V)	$A \to aAb$	$(a, b \in V)$
(VI)	$A \to ab$	$(a, b \in V)$

Действительно, всякий полный вывод в этой грамматике происходит по одной из трех схем: а) I^k II; б) I^k III V^l VI; в) I^k IV ($k, l = 0, 1, \ldots$); по схеме а) порождаются всевозможные цепочки нечетной длины,

по схемам б) и в) — всевозможные цепочки вида $\omega_1\omega_2$, где $|\omega_1| = |\omega_2|$ и $\omega_2 \neq \hat{\omega}_1$.

Пример 10. Язык $\{a^n b^n a^n \mid n = 1, 2, \ldots\}$ порождается грамматикой со схемой $\{I \to aIBa,\ I \to aba,\ aB \to Ba,\ bB \to bb\}$.

Действительно, каков бы ни был полный вывод в этой грамматике, в нем на одном и только на одном шаге должно применяться второе правило, и после этого цепочка будет иметь вид $a^n b\omega$, где ω содержит n вхождений a и $n-1$ вхождений B. Вывод закончится лишь тогда, когда все B превратятся в b, но для этого все B должны «собраться вместе» в середине цепочки.

Пример 11. Пусть $V = \{a_1, \ldots, a_k, b\}$. Язык $\{xbx \mid x \in \{a_1, \ldots, a_n\}^*\}$ порождается грамматикой со схемой $\{I \to IA_ia_i,\ I \to b,\ a_iA_j \to A_ja_i,\ bA_i \to a_ib \mid i, j = 1, \ldots, k\}$.

Пример 12. Пусть $V = \{a_1, \ldots, a_k, b\}$. Язык $\{x_1 b x_2 \mid x_1, x_2 \in \{a_1, \ldots, a_k\}^*,\ x_1 \neq x_2\}$ порождается Б-грамматикой со схемой:

(I) $I \to I'$
(II) $I \to I''$
(III) $I' \to I'a_i \quad (i = 1, \ldots, k)$
(IV) $I' \to A_ia_i \quad (i = 1, \ldots, k)$
(V) $A_l \to a_jA_ia_l \quad (i, j, l = 1, \ldots, k)$
(VI) $A_l \to a_jB \quad (i, j = 1, \ldots, k;\ i \neq j)$
(VII) $B \to a_jB \quad (j = 1, \ldots, k)$
(VIII) $B \to b$
(IX) $A_l \to b \quad (i = 1, \ldots, k)$
(X) $I'' \to C$
(XI) $C \to a_iCa_j \quad (i, j = 1, \ldots, k)$
(XII) $C \to a_iD \quad (i = 1, \ldots, k)$
(XIII) $D \to a_iD \quad (i = 1, \ldots, k)$
(XIV) $D \to b$

Легко видеть, что любой полный вывод в этой грамматике происходит по одной из схем:

а) $\mathrm{I\,III}^k\,\mathrm{IV\,V}^l\,\mathrm{VI\,VII}^m\,\mathrm{VIII}$; б) $\mathrm{I\,III}^k\,\mathrm{IV\,V}^l\,\mathrm{IX}$;
в) $\mathrm{II\,X\,XI}^k\,\mathrm{XII\,XIII}^l\,\mathrm{XIV}$ $(k, l, m = 0, 1, \ldots)$.

По схеме а) получаются всевозможные цепочки вида $x'a_jy'bx''a_iy''$, где $x', y', x'', y'' \in \{a_1, \ldots, a_k\}^*$, $|x'| = |x''|$, $i \neq j$; по схеме б) — всевозможные цепочки вида $z'bz''$, где $z', z'' \in \{a_1, \ldots, a_k\}^*$ и $|z'| < |z''|$; по схеме в) — то же с противоположным неравенством. Объединение этих трех множеств цепочек и есть нужный язык.

Пример 13. Множество $\{n^2 \mid n = 1, 2, \ldots\}$ порождается грамматикой со схемой:

(I) $I \to ABBDF$
(II) $BD \to DCB$
(III) $BC \to CB$
(IV) $AD \to AAE$
(V) $EC \to AE$
(VI) $EB \to BE$
(VII) $EF \to BBDF$
(VIII) $DF \to |$
(IX) $B \to |$
(X) $A \to |$
(XI) $I \to |$

Нетрудно видеть, что полный вывод в этой грамматике происходит следующим образом. Сначала из I по правилу I получается цепочка $ABBDF = A^1B^{2\cdot1}DF$, а затем выполняется произвольное число циклов, на каждом из которых цепочка $A^{n^2}B^{2n}DF$ преобразуется в $A^{(n+1)^2}B^{2(n+1)}DF$. Цикл состоит в том, что каждое вхождение B с помощью «стимулятора» D порождает «двойника» — вхождение C (правило II), которое затем переходит влево (правило III), после чего превращается в A (правило V) — причем до начала превращений C в A появляется еще одно вхождение A (правило IV), — и, наконец, появляются еще два вхождения B (правило VII); как раз в этот момент «стимулятор» снова принимает «исходное положение». После окончания каждого очередного цикла можно, вместо того чтобы начинать новый, применить правила VIII—X, превращающие $A^{n^2}B^{2n}DF$ в $(n+1)^2$. (Впрочем, правило X можно начать

применять и раньше — это ничего не изменит). Так получается любая цепочка вида $(n+1)^2$, и ясно, что только такие цепочки в словаре $\{|\}$ выводимы из I по правилам I—X. Цепочка $|=1^2$ получается по правилу XI.

Дальнейшие примеры «абстрактных» грамматик см. в упражнениях 1.14 и 1.15 (см. также упражнение 3.6).

Пример 14. Пусть $\Gamma = \langle V, W, \text{ПРЕДЛ}, R\rangle$, где

1) V состоит из словоформ русского языка;

2) W состоит из символов ПРЕДЛ, $\tilde{S}_{txyz}$, S_{txyz}, A_{xyz}, V^{Ψ}_{xy}, Loc_i, где t = одуш, неодуш; x = муж, жен, ср; y = ед, мн; z = им, род, дат, вин, твор, предл; $i = 1, 2, 3, 4, 5$; Ψ есть произвольная подпоследовательность последовательности 12345 (возможно, пустая).

Содержательная интерпретация вспомогательных символов: ПРЕДЛ — «предложение»; $\tilde{S}_{txyz}$ — «группа существительного, одушевленного или неодушевленного в зависимости от t, рода x в числе y и падеже z»; S_{txyz} — «существительное» (смысл индексов тот же); A_{xyz} — «прилагательное в роде x, числе y и падеже z»; V^{Ψ}_{xy} — «(непереходный) глагол в изъявительном наклонении, прошедшем времени, роде x и числе y вместе с обстоятельствами тех типов (см. ниже), номера которых входят в Ψ»; Loc_i — «обстоятельства различных типов, характеризующие движение», а именно Loc_1 — «путь» (*по дороге, по воздуху*), Loc_2 — «отправной пункт» (*из деревни, от реки*), Loc_3 — «пункт назначения» (*в лес, на концерт*), Loc_4 — «придорожные предметы» (*мимо деревни, вдоль опушки*), Loc_5 — «способ передвижения» (*пешком, в карете*).

3) R состоит из следующих правил (всюду, где не оговорено противное, индексы принимают все допустимые значения):

I. $\text{ПРЕДЛ} \to \tilde{S}_{txy\ \text{им}} V^{12345}_{xy}$

II. $\text{ПРЕДЛ} \to V^{12345}_{xy} \tilde{S}_{txy\ \text{им}}$

IIIа. $V^{\Psi}_{xy} \to V^{\Psi-i}_{xy} \text{Loc}_i$ (здесь предполагается, что Ψ содержит i; $\Psi - i$ означает последовательность, получаемую из Ψ выбрасыванием i)

IIIб. $V^{\Psi}_{xy} \tilde{S}_{txy\ \text{им}} \to V^{\Psi-i}_{xy} \tilde{S}_{txy\ \text{им}}$

IV. $V^{\Psi}_{xy} \to V_{xy}$

Va. $\tilde{S}_{txyz} \to A_{xyz} \tilde{S}_{txyz}$, если либо $z \neq$ вин, либо $y =$ ед и $x \neq$ муж

Vб. $\tilde{S}_{\text{одуш}\ xy\ \text{вин}} \to A_{xy\ \text{род}} \tilde{S}_{\text{одуш}\ xy\ \text{вин}}$ } если x=муж

Vв. $\tilde{S}_{\text{неодуш}\ xy\ \text{вин}} \to A_{xy\ \text{им}} \tilde{S}_{\text{неодуш}\ xy\ \text{вин}}$ } или y = мн

VI. $\tilde{S}_{txyz} \to S_{txyz}$

VIIа. $S_{\text{одуш муж}\ yz} \to \textit{ямщик}_{yz}$, *парень*$_{yz}$, ...
$S_{\text{неодуш ср}\ yz} \to \textit{поле}_{yz}$, *колесо*$_{yz}$, ...

. . . . . .

VIIб. $A_{xyz} \to \textit{молодой}_{xyz}$, *сверхскоростной*$_{xyz}$, ...

VIIв. $V_{xy} \to \textit{ехал}_{xy}$, *летел*$_{xy}$, ...

VIIг. $\text{Loc}_1 \to$ *по* $\tilde{S}_{txy\ \text{дат}}$;
$\text{Loc}_2 \to$ *из* $\tilde{S}_{txy\ \text{род}}$, *от* $\tilde{S}_{txy\ \text{род}}$;
$\text{Loc}_3 \to$ *в* $S_{txy\ \text{вин}}$, *на* $\tilde{S}_{txy\ \text{вин}}$, *к* $\tilde{S}_{txy\ \text{дат}}$;
$\text{Loc}_4 \to$ *мимо* $\tilde{S}_{txy\ \text{род}}$, *вдоль* $S_{txy\ \text{род}}$;
$\text{Loc}_5 \to$ *пешком*, *верхом*, *на* $\tilde{S}_{txy\ \text{предл}}$, *в* $\tilde{S}_{txy\ \text{предл}}$.

(Здесь *ямщик*$_{yz}$ и т. п. — сокращенная запись; например, *ямщик*$_{\text{род. мн.}}$ означает *ямщиков*.)

Читатель легко проверит, что язык $L(\Gamma)$ содержит такие, например, цепочки, как *Ехал ямщик мимо деревни*, *Ехала деревня мимо ямщика*, *Молодая очаровательная ведьма летела на межобластной шабаш на сверхскоростном турбореактивном помеле.*

§ 1.4. Машины Тьюринга

Машины Тьюринга и различные их разновидности играют в теории грамматик чрезвычайно важную роль. С одной стороны, выявление связей и параллелей между теорией машин и теорией грамматик позволяет лучше понять главные идеи последней; с другой — при изучении грамматик машины Тьюринга нередко оказываются удобным техническим средством.

Концепция машины Тьюринга, которой мы будем пользоваться, отличается от обычной, излагаемой в руководствах по теории алгоритмов (см., например, [Kleene 1952]). Поскольку наше изложение рассчитано на читателя, знакомого с основами теории алгоритмов, мы не будем в основном тексте касаться вопроса о взаимоотношении нашей концепции с обычной и отнесем это в упражнения. По той же причине мы не разбираем примеров и проводим рассуждения о машинах Тьюринга по большей части «на пальцах», на уровне объяснения руководящих идей (но достаточно подробно, чтобы знакомый с соответствующей техникой читатель мог восстановить все формальные детали; тем, кто найдет свой уровень владения этой техникой недостаточным, полезно выполнить упражнения 1.16—1.19). Само определение машины также не будет до конца формализованным; в нем будут фигурировать такие наглядные образы, как «лента», «ячейка, обозреваемая головкой» и т. п.; впрочем, в этом мы следуем традиции.

Обычная «классическая» машина Тьюринга представляет собой устройство для реализации алгоритмов. Работа алгоритма есть детерминированный процесс: каждый следующий его шаг жестко определяется предыдущим, и ход всего процесса полностью предопределен исходными данными. Для достижения такой детерминированности на машину Тьюринга в ее классическом виде накладывается требование ни в какой момент не допускать возможности выполнения двух разных команд. Нам, однако, понадобятся и такие машины Тьюринга, которые реализуют не алгоритмы, а исчисления, сходные с грамматиками, — «недетерминированные машины Тьюринга»; формально они будут отличаться от обычных только отсутствием упомянутого только что требования. Впрочем, термин «недетерминированная машина Тьюринга» неудобен: детерминированная машина оказывается частным случаем недетерминированной. Поэтому мы будем называть недетерминированные в указанном только что смысле машины просто «машинами Тьюринга», а «классические» — «детерминированными машинами Тьюринга».

Второе отличие наших машин от обычных состоит в том, что мы снабдим их двумя лентами, причем одна из них будет служить «входной»; на этой ленте записывается исходная цепочка, которую машина в процессе работы прочитывает, не изменяя ее, — и, более того, каждый символ читается только один раз; вся остальная работа производится на другой, «рабочей» ленте. Это видоизменение, в отличие от предыдущего, не имеет принципиального значения, но оно позволит нам сделать некоторые рассмотрения более прозрачными *).

Далее, обычно в определение машины Тьюринга включают требование, чтобы из любой ситуации, в которой состояние машины не заключительное, она переходила в некоторую новую ситуацию (в частном случае эта новая ситуация может совпадать со старой); мы же отказываемся от этого требования. Данная модификация также не принципиальна и служит лишь для большего удобства моделирования грамматик.

Последней особенностью наших машин будет «эластичность» рабочей ленты. Это означает, что: а) стирая символ в ячейке рабочей ленты, машина (или, если угодно, головка) уничтожает и самое ячейку, так что соседние с ней ячейки становятся соседними между собой («стягивание» ленты); б) между любыми двумя ячейками рабочей ленты машина может создать новую, сразу записав в ней что-либо («растягивание» ленты). При таком способе работы отпадает надобность в «пустом символе». Для нас «эластичность» будет удобна тем, что позволит при моделировании грамматик машинами обходиться без сдвигов всего содержимого ленты, которые иначе пришлось бы производить на каждом шаге.

Перейдем к формулировке определения.

Машина Тьюринга с эластичной рабочей лентой, сокращенно Э-машина, состоит из:

а) двух конечных **) разделенных на ячейки лент — входной и рабочей;

*) Главное преимущество введения входной ленты состоит в том, что МП-машины (см. ниже, § 4.5) становятся частным случаем машин Тьюринга.

**) Имеется в виду конечность ленты в каждый момент работы машины. В процессе работы размеры рабочей ленты могут неограниченно расти; входная лента не меняется в процессе работы, но может оказаться сколь угодно длинной при большом объеме исходных данных.

б) двух алфавитов (словарей) V и W, также называемых входным и рабочим (их элементы называются соответственно входными и рабочими символами);

в) символов $\#$, $\ddagger \notin V \cup W$, называемых соответственно левым и правым граничными маркерами (множества $V \cup \{\#, \ddagger\}$ и $W \cup \{\#\}$ будут обозначаться соответственно V' и W');

г) входной и рабочей головок, движущихся по соответствующим лентам (в любой момент работы машины каждая головка обозревает некоторую ячейку своей ленты, иначе — находится в этой ячейке);

д) конечного множества $Q = \{q_1, \ldots, q_s\}$ (внутренних) состояний, в котором выделены элемент q_1, называемый начальным состоянием, и непустое подмножество Q_0, элементы которого называются заключительными состояниями;

е) программы — конечного множества цепочек в алфавите $V' \cup W \cup Q \cup \{Л, П, \rightarrow\}$, называемых инструкциями, каждая из которых имеет один из следующих видов: (i) $q_\alpha a \rightarrow q_\beta$; (ii) $Aq_\alpha \rightarrow q_\beta$; (iii) $q_\alpha \rightarrow Aq_\beta$; (iv) $q_\alpha \rightarrow q_\beta Л$; (v) $q_\alpha \rightarrow q_\beta П$. Здесь $a \in V'$, $A \in W$, q_α, $q_\beta \in Q$.

Содержательно каждая инструкция интерпретируется как разрешение выполнить некоторые действия, зависящие от состояния, в котором находится машина, и (вообще говоря) от обозреваемых головками символов, а также от положения рабочей головки. Именно, инструкция, имеющая в левой части q_α и в правой q_β, означает, что, находясь в состоянии q_α, машина может в случае (i) при $a \neq \ddagger$ сдвинуть входную головку, если в обозреваемой ею ячейке записан символ a, на одну ячейку вправо; в случае (ii) уничтожить обозреваемую рабочую ячейку, если в ней записан символ A, и поместить рабочую головку в ячейку, примыкающую к уничтоженной слева; в случае (iii) создать непосредственно справа от обозреваемой рабочей ячейки новую ячейку, записать там символ A и переместить туда (рабочую) головку; в случаях (iv) и (v) передвинуть рабочую головку, если она не находится в крайней слева, соответственно в крайней справа ячейке, на одну ячейку влево, соответственно вправо; одновременно машина должна —

в любом из описанных случаев — перейти в состояние q_β (которое может и совпадать с q_α). Особым образом интерпретируется выполнение инструкции типа (i) при $a = \ddagger$: в этом случае происходит только перемена состояния, а входная головка остается на месте (как и рабочая).

Указанную совокупность действий мы будем называть элементарным шагом (или просто шагом) работы машины.

Вполне возможно, что в некоторый момент окажутся применимыми несколько разных инструкций, так что машина сможет выполнять любую из них «по выбору»; это и есть недетерминированность.

Полный набор сведений о том, в каком состоянии находится Э-машина, что́ записано на лентах и какие ячейки обозреваются головками, мы будем называть ситуацией данной машины. Формально ситуацию можно определить как упорядоченную систему $(q_\alpha, x', x'', X', X'')$, где $q_\alpha \in Q$, $x' \in V'^*$, $x'' \in V'^+$, $X' \in W'^*$, $X'' \in W'^+$. Здесь $x'x''$ — цепочка, записанная на входной ленте, x' — часть этой цепочки влево от головки, $X'X''$ и X' — то же для рабочей ленты. Ситуацию вида $(q_1, \Lambda, \# x \ddagger, \Lambda, \#)$, где $x \in V^*$, мы будем называть начальной; ситуацию вида $(q_f, \# x, \ddagger, \Lambda, \#)$, где $q_f \in Q_0$ и $x \in V^*$, — заключительной; будем употреблять также выражения «начальная x-ситуация», «заключительная x-ситуация», смысл которых очевиден.

Если ситуация S' может быть получена из ситуации S применением некоторой инструкции, будем говорить, что S' непосредственно достижима из S (в машине M), и писать $S \underset{M}{\succ} S'$ или $S \succ S'$. Последовательность ситуаций $C = (S_0, S_1, \ldots, S_n)$ $(n \geqslant 1)$ будем называть вычислением (машины M), если $S_{i-1} \underset{M}{\succ} S_i$ для каждого i, $1 \leqslant i \leqslant n$. Число n называется длиной вычисления C. Если существует вычисление машины M, начинающееся ситуацией S и оканчивающееся ситуацией S', мы говорим, что S' достижима из S (в машине M), и пишем $S \underset{M}{\succ\!\!\succ} S'$ или $S \succ\!\!\succ S'$. Вычисление, начинающееся начальной x-ситуацией,

назовем x-вычислением, вычисление, начинающееся начальной (x-)ситуацией и оканчивающееся заключительной (x-)ситуацией, — полным (x-)вычислением (машины M). x-вычисление, оканчивающееся ситуацией $(q_f, \#x, \ddagger, \Lambda, \#y)$, где $q_f \in Q_0$, будем называть $[x, y]$-вычислением (машины M). (Таким образом, полное x-вычисление — это то же самое, что $[x, \Lambda]$-вычисление.) Если сушествует полное x-вычисление машины M, мы будем говорить, что M допускает цепочку x. Множество цепочек, допускаемых машиной M, мы будем называть языком, допускаемым машиной M, и обозначать $L(M)$.

Заметим, что в процессе x-вычисления граничный маркер, записанный в крайней слева рабочей ячейке, никогда не уничтожается, и ни в какой другой рабочей ячейке он появиться не может. Поэтому мы будем, например, говоря «на рабочей ленте записан только символ A», «рабочая лента уничтожается» и т. п., подразумевать: «на рабочей ленте записана цепочка $\#A$», «уничтожаются все ячейки рабочей ленты, кроме той, в которой записан символ $\#$». Полное вычисление начинается и кончается при отсутствии рабочей ленты; входная цепочка к концу полного вычисления должна быть вся прочитана.

Если каждой инструкции Э-машины M взаимно однозначно сопоставлен некоторый символ c_j, то цепочка $c_{j_1} \ldots c_{j_n}$, где c_{j_l} — символ, сопоставленный инструкции, выполняемой на i-м шаге вычисления $C = (S_0, \ldots, S_n)$, будет называться шифром вычисления C. Вычисление имеет точно один шифр и вполне определяется этим шифром и исходной ситуацией (ср. упражнения 1.10 и 1.11).

Будем называть Э-машину M детерминированной (ДЭ-машиной), если ее программа удовлетворяет следующим двум условиям: (а) она не содержит двух различных инструкций с одинаковыми левыми частями; (б) если некоторое состояние содержится в левой части инструкции одного из типов (i) — (v), то оно не может содержаться в левой части инструкции другого типа.

ДЭ-машина для всякой входной цепочки x имеет не более одного полного x-вычисления.

Мы будем говорить, что ДЭ-машина, допускающая язык L, распознает этот язык, если для любой цепочки x во входном алфавите она имеет лишь конечное множество x-вычислений (попросту говоря, если, начав работать с произвольной цепочкой на входной ленте, машина в конце концов останавливается). Распознающая язык ДЭ-машина существует тогда и только тогда, когда этот язык рекурсивен (см. ниже, замечание к упражнению 1.22, б)).

Пусть M — Э-машина с входным алфавитом V и рабочим алфавитом W. Назовем Э-машину $\tilde{M}$ с рабочим алфавитом $\tilde{W} \supseteq V \cup W \cup \{d\}$ $(d \notin V \cup W)$ прямой одноленточной моделью (п. о. м.) машины M, если $[x, y]$-вычисление машины M существует тогда и только тогда, когда в машине $\tilde{M}$ из ситуации $(q_1, \Lambda, \#\ddagger, \Lambda, \#xd)$ достижима ситуация вида $(q_f, \Lambda, \#\ddagger, \#x, dy)$, где $q_f \in Q_0$.

Лемма 1.5. *Для любой Э-машины можно построить ее п. о. м. Если при этом исходная машина детерминированная, то и п. о. м. можно сделать детерминированной.*

Доказательство. Принцип работы $\tilde{M}$ может быть таким: во «входной зоне» (левее d) и «рабочей зоне» (правее d) рабочей ленты $\tilde{M}$ функционирует точно так же, как M на входной и рабочей лентах соответственно, но всякий раз, когда M переходит от «входного» шага к «рабочему» или наоборот, рабочая головка машины $\tilde{M}$ будет передвигаться из зоны в зону, предварительно поставив в обозревавшейся перед тем ячейке метку, чтобы найти эту ячейку по возвращении. Ясно, что свойство детерминированности, если машина M им обладает, при таком преобразовании сохраняется.

Теорема 1.2. *Для любой Э-машины M можно построить ДЭ-машину M_1 такую, что $L(M_1) = L(M)$.*

Доказательство. Пусть $P = \{c_1, \ldots, c_n\}$ — множество символов, находящееся во взаимно однозначном соответствии с программой машины M. Построим ДЭ-машину M_1 с входным алфавитом V и рабочим алфавитом, содержащим $V \cup W \cup P \cup \{d, e, f\}$, где V, W — соответственно входной и рабочий алфавиты машины M и d, e, $f \notin V \cup W \cup P$, работающую следующим образом. Начиная работать в начальной x-ситуации, M_1 прежде

всего переписывает x на рабочую ленту, приписывает справа dfc_1e, передвигает рабочую головку на граничный маркер и переходит в некоторое специальное состояние q'. Дальнейшая работа M_1 состоит в последовательном выполнении «макрошагов», которые могут быть описаны следующим образом. Перед началом «макрошага» машина находится в ситуации $(q', \# x, \#, \Lambda, \# xdfZe)$, где Z — некоторая цепочка в алфавите P. При выполнении «макрошага» M_1 функционирует на участке ленты между $\#$ и началом Z, как описанная в доказательстве леммы 1.5 п. о. м. машины M, однако с той разницей, что она производит при этом не любые «M-шаги», а такие, в результате которых получается «M-вычисление» с шифром Z. Для этого головка перед каждым «M-шагом» передвигается вправо до символа c_j, стоящего непосредственно вслед за f, «перетаскивает» f вправо через этот символ, затем возвращается в ячейку, где следует выполнять очередной «M-шаг» (эта ячейка, разумеется, должна быть ранее помечена), и выполняет его, следуя той инструкции, которая отвечает символу c_j, если эта инструкция применима. Если же она неприменима, то f «перетаскивается» влево, пока не окажется перед первым символом цепочки Z; эта последняя заменяется цепочкой, непосредственно следующей за ней в смысле некоторого, раз навсегда фиксированного стандартного порядка на P^*; участок ленты между d и f («рабочая зона») уничтожается; метка, имеющаяся на одном из символов цепочки x, стирается; рабочая головка становится на граничный маркер, и машина переходит в состояние q', после чего выполняется следующий «макрошаг». Если машине удалось «протащить» символ f через всю цепочку Z (так что он оказался непосредственно перед e), то различаются два случая. а) Если имитированное на данном «макрошаге» x-вычисление машины M с шифром Z является полным, то рабочая лента уничтожается и машина останавливается, перейдя в состояние, сигнализирующее об успешном окончании работы, — цепочка x допущена. б) В противном случае дальнейшие действия таковы же, как в рассмотренном выше случае столкновения с неприменимой инструкцией.

Ясно, что машина M_1 имитирует все x-вычисления машины M, и если среди них имеется хоть одно полное, то она рано или поздно это обнаружит, а в противном случае будет работать вечно. Возможность обеспечить детерминированность M_1 очевидна.

Пусть M — ДЭ-машина с входным алфавитом V и рабочим алфавитом W, причем $V \subseteq W$, и f — функция, определенная на некотором подмножестве множества V^* и принимающая значения также из V^*. Мы скажем, что машина M вычисляет функцию f, если $[x, y]$-вычисление машины M для любых x, $y \in V^*$ существует тогда и только тогда, когда $y = f(x)$.

Теорема 1.3. а) *Для всякой ДЭ-машины, вычисляющей некоторую функцию, можно построить ДЭ-машину, допускающую множество значений этой функции.* б) *Для всякой ДЭ-машины M можно построить ДЭ-машину, вычисляющую функцию с множеством значений $L(M)$.*

Доказательство. а) Пусть ДЭ-машина M с входным алфавитом V и рабочим алфавитом W вычисляет функцию f. Построим Э-машину M_1, работающую так: сначала на рабочей ленте записывается произвольная цепочка $z \in V^*$ и к ней приписывается справа символ $d \notin V \cup W$; потом M_1 работает как п. о. м. машины M, после чего на рабочей ленте оказывается записанной цепочка $zdf(z)$ (если только $f(z)$ определена; в противном случае M_1 работает вечно); затем входная цепочка x (которая до этого не читалась) сравнивается с цепочкой $f(z)$; если окажется $x = f(z)$, и только в этом случае, рабочая лента уничтожается, и машина переходит в заключительное состояние. Ясно, что M_1 допускает как раз множество значений f. Остается воспользоваться теоремой 1.2.

б) ДЭ-машина M_1, которая сначала переписывает входную цепочку на рабочую ленту, затем работает как п. о. м. данной машины M и, наконец, в случае успешного окончания работы уничтожает «разграничитель» d, будет вычислять функцию, принимающую значение x для любого $x \in L(M)$ и не определенную вне $L(M)$.

Выясним теперь взаимоотношение между Э-машинами и грамматиками.

Будем говорить, что грамматика Γ и Э-машина M эквивалентны, если $L(\Gamma) = L(M)$. Аналогично определяется эквивалентность двух Э-машин.

Теорема 1.4. а) *Для всякой Э-машины можно построить эквивалентную ей грамматику.* б) *Для всякой грамматики можно построить эквивалентную ей Э-машину.*

Доказательство. а) Пусть M — данная Э-машина с входным алфавитом V и M_1 — ее п. о. м. По M_1 легко построить машину M_2, имеющую такие два состояния q' и q'', что при $x \in V^*$ из ситуации $(q', \Lambda, \# \ddagger, \Lambda, \# x)$ тогда и только тогда достижима ситуация $(q'', \Lambda, \# \ddagger, \Lambda, \#)$, когда $x \in L(M)$. Построим теперь грамматику Γ следующим образом. 1) Основной словарь Γ есть V. 2) Вспомогательный словарь Γ есть $(W - V) \cup Q \cup \{I\}$, где W и Q — соответственно рабочий алфавит и множество состояний машины M_2 и $I \notin V \cup W \cup Q$. 3) Начальный символ Γ есть I. 4) Схема Γ состоит из правил $I \to \# q$, $\# q' \to \Lambda$ и правил, определенным образом сопоставляемых инструкциям машины M_2, а именно: каждой инструкции $\delta q_\alpha \to q_\beta$ типа (ii) сопоставляется правило $q_\beta \to \delta q_\alpha$; каждой инструкции $q_\alpha \to \delta q_\beta$ типа (iii) — правило $\delta q_\beta \to q_\alpha$; каждой инструкции $q_\alpha \to q_\beta$ Л типа (iv) — всевозможные правила вида $q_\beta \gamma \to \gamma q_\alpha$, где $\gamma \in W$, и каждой инструкции вида $q_\alpha \to q_\beta$ П типа (v) — всевозможные правила вида $\gamma q_\beta \to q_\alpha \gamma$, где $\gamma \in W$. Таким образом, схема грамматики Γ представляет собой «обращение» программы M_2; поэтому, какова бы ни была цепочка $x \in V^*$, в грамматике Γ тогда и только тогда можно вывести $\# q' x$ из $\# q''$, или, что то же самое, x из I, когда в M_2 ситуация $(q'', \Lambda, \# \ddagger, \Lambda, \#)$ достижима из $(q', \Lambda, \# \ddagger, \Lambda, \# x)$. Отсюда $L(\Gamma) = L(M)$.

б) Пусть $\Gamma = \langle V, W, I, R \rangle$ — произвольная грамматика.

Построим Э-машину M с входным алфавитом V и рабочим алфавитом $V \cup W$, которая будет сначала переписывать входную цепочку на рабочую ленту, затем производить на рабочей ленте произвольный вывод в Γ в обратном порядке, а потом в случае, когда в результате такого «обратного вывода» на рабочей ленте останется только символ I, уничтожать его и переходить

в заключительное состояние; при этом можно сделать так, чтобы ни в каком другом случае заключительное состояние не наступало. Очевидно, $L(M) = L(\Gamma)$.

Подведем итог. В силу только что доказанной теоремы класс языков, порождаемых грамматиками, совпадает с классом языков, допускаемых Э-машинами. Этот последний ввиду теорем 1.2 и 1.3 совпадает с классом множеств значений функций, вычисляемых ДЭ-машинами. Но легко видеть, что ДЭ-машинами вычисляются в точности те же функции, что и обычными однолеточными машинами Тьюринга (упражнение 1.20), т. е. частично рекурсивные функции *); множества же значений частично рекурсивных функций суть рекурсивно перечислимые множества. Итак, из теорем 1.2—1.4 вытекает

Теорема 1.5. *Класс языков, порождаемых грамматиками, совпадает с классом рекурсивно перечислимых языков.*

Теперь мы можем разъяснить сделанное в § 1.2 утверждение, что «порождающий» способ описания языков не уступает по силе никакому другому эффективному способу. Именно, в силу доказанных теорем это утверждение немедленно вытекает из тезиса Черча, согласно которому всякая функция, допускающая вычисление каким-либо эффективным в интуитивном смысле способом, частично рекурсивна. Вместе с тезисом Черча наше утверждение носит, разумеется, не математический, а естественнонаучный характер.

Замечание. Из теоремы 1.5 и известной теоремы Поста (см., например, [Kleene 1952], стр. 237 русского перевода), в силу которой множество тогда и только тогда рекурсивно, когда оно само и его дополнение рекурсивно перечислимы, следует — поскольку существуют нерекурсивные рекурсивно перечислимые множества — замечание, сделанное в конце § 1.2.

---

*) Обычно в курсах теории алгоритмов с помощью машин Тьюринга определяются только числовые частично рекурсивные функции, а «словарные» частично рекурсивные функции вводятся с помощью нумерации цепочек. Не составляет большого труда доказать равносильность такого определения используемому нами «непосредственному» (точные формулировки см. в упражнении 1.21).

Упражнения

1.1. Подсчитать число вхождений подцепочек в цепочку длины n.

1.2. Доказать, что для произвольных цепочек φ и ψ тогда и только тогда $\varphi\psi = \psi\varphi$, когда существуют цепочка ω и числа $m, n = 0, 1, \ldots$ такие, что $\varphi = \omega^m$, $\psi = \omega^n$.

1.3. Пусть $V = \{a_1, \ldots, a_k\}$ — некоторый словарь. Сопоставим каждой цепочке $\omega \in V^*$ число $\nu(\omega)$ следующим образом: $\nu(\Lambda) = 0$; $\nu(a_{i_1} \ldots a_{i_n}) = k^{n-1} \cdot i_1 + \ldots + k^2 \cdot i_{n-2} + k \cdot i_{n-1} + i_n$. (Не смешивать с обычным представлением натурального числа в системе счисления! В чем разница?)

Показать, что ν — взаимно однозначное отображение V^* на $N = \{0, 1, \ldots\}$ и что $\nu(\varphi) < \nu(\psi)$ тогда и только тогда, когда φ предшествует ψ в смысле стандартного порядка, отвечающего пересчету $a_1, \ldots, a_k$.

1.4. а) Доказать, что $L(L_1 \cup L_2) = LL_1 \cup LL_2$, $(L_1 \cup L_2)L = L_1L \cup L_2L$.

б) Имеют ли место аналогичные тождества для пересечения?

1.5. Сформулировать строгие (индуктивные) определения многочлена и регулярного выражения.

1.6. Представить с помощью бескоэффициентных регулярных выражений от языков $\{a_1\}, \ldots, \{a_n\}, \{\Lambda\}$:

а) множество цепочек, содержащих вхождения данной непустой цепочки $\omega = a_{i_1} \ldots a_{i_s}$;

б) множество цепочек, начинающихся (оканчивающихся) вхождением данной непустой цепочки $\omega = a_{i_1} \ldots a_{i_s}$;

в) множество цепочек, не содержащих вхождений цепочки $\omega = a_{i_1} \ldots a_{i_s}$;

г) множество цепочек, содержащих ровно k вхождений цепочки $a_{i_1} \ldots a_{i_s}$.

1.7. Показать, что для всякого регулярного выражения $\varphi(\xi_1, \ldots, \xi_k)$ найдется такое выражение $\psi(\xi_1, \ldots, \xi_k) = f(g_1^*(\xi_1, \ldots, \xi_k), \ldots, g_s^*(\xi_1, \ldots, \xi_k), \xi_1, \ldots, \xi_k)$ $(s \geqslant 0)$, где f, $g_1, \ldots, g_s$ — многочлены, что для любых k языков $L_1, \ldots, L_k$, каждый из которых содержит пустую цепочку, $\varphi(L_1, \ldots, L_k) = \psi(L_1, \ldots, L_k)$. При этом, если φ не содержит коэффициентов, то это же верно для $f, g_1, \ldots, g_s$.

1.8. Выразить объединение, умножение и итерацию языков через подстановку.

1.9. Для того, чтобы каждому выводу в грамматике Γ соответствовал единственный размеченный вывод, необходимо и достаточно, чтобы схема Γ не содержала правил $\varphi_1 \to \psi_1$, $\varphi_2 \to \psi_2$, удовлетворяющих одному из следующих двух условий: (i) $\exists\omega\exists\omega'(\omega\omega' \neq \Lambda \,\&\, \varphi_2 = \omega\varphi_1\omega' \,\&\, \psi_2 = \omega\psi_1\omega')$; (ii) $\exists\varphi_1'\exists\varphi_2'\exists\zeta\exists\theta[\varphi_1 = \varphi_1'\zeta \,\&\, \varphi_2 = \zeta\varphi_2' \,\&\, (\varphi_1' = \psi_1\theta \,\&\, \varphi_2' = \theta\psi_2 \vee \psi_1 = \varphi_1'\theta \,\&\, \psi_2 = \theta\varphi_2') \,\&\, \varphi_1'\varphi_2' \neq \Lambda]$. Доказать,

1.10. Назовем характеристикой вывода [Стоцкий 1967] *) последовательность применяемых в этом выводе правил; аналогично для размеченного вывода. (Размеченный вывод имеет, очевидно, единственную характеристику.) Показать, что в грамматике тогда и только тогда существуют различные размеченные выводы с одинаковыми характеристиками, отвечающие одному и тому же выводу, когда ее схема содержит правило вида $\varphi^m \to \varphi^n$ $(m, n = 0, 1, \ldots)$.

1.11. Могут ли полные выводы в одной и той же грамматике, оканчивающиеся разными цепочками, иметь одинаковые характеристики?

1.12. Назовем вывод в грамматике $\Gamma = \langle V, W, I, R\rangle$ приведенным налево, если на каждом его шаге соответствующее правило применяется к самому левому вхождению своей левой части (при подходящей разметке вывода). Множество цепочек из V^*, выводимых в Γ из I с помощью выводов, приведенных налево, обозначим $L_Л(\Gamma)$. Показать, что:

а) для произвольной грамматики Γ можно построить **) такую грамматику Γ', что $L(\Gamma) = L_Л(\Gamma')$; если при этом Γ — НС-грамматика, то и Γ' можно сделать НС-грамматикой;

б) если Γ — Б-грамматика, то $L(\Gamma) = L_Л(\Gamma)$.

1.13. Назовем грамматикой со стоп-правилами упорядоченную четверку $\Delta = \langle V, I, R, R'\rangle$, где V — конечное множество, $I \in V$, R — конечное множество правил такого же вида, как правила обычной грамматики, и R' — подмножество R («множество стоп-правил»). Вывод в Δ (вывод определяется, как для обычной грамматики) назовем 1-выводом, если в нем применяются только правила из $R - R'$ и к его последней цепочке никакое правило из R не применимо, и 2-выводом, если на всех его шагах, кроме последнего, применяются правила из $R - R'$, а на последнем — правило из R'. Множество цепочек, выводимых из I с помощью 1-(2-) выводов, обозначим $L_1(\Delta)$ $(L_2(\Delta))$. Показать, что:

а) для любой грамматики Γ можно построить такие грамматики со стоп-правилами Δ, Δ_1, Δ_2, что $L(\Gamma) = L_1(\Delta) \cup L_2(\Delta) = L_1(\Delta_1) = L_2(\Delta_2)$;

б) для любой грамматики со стоп-правилами Δ можно построить такие грамматики Γ, Γ_1, Γ_2, что $L_1(\Delta) \cup L_2(\Delta) = L(\Gamma)$, $L_1(\Delta) = L(\Gamma_1)$, $L_2(\Delta) = L(\Gamma_2)$.

[Friš 1965].

1.14. Построить А-грамматики, порождающие:

а) множество всевозможных цепочек в словаре $V = \{a_1, \ldots, a_k\}$, содержащих вхождения данной цепочки $x = a_{i_1} \ldots a_{i_s}$ (соответственно содержащих не менее n вхождений x, начинающихся вхождением x, оканчивающихся вхождением x, не содержащих вхождений x, содержащих ровно n вхождений x);

б) язык $V_n^+ = \{x \mid x \in V^+, |x| \geqslant n\}$;

в) язык $C'L_0$, где L_0 — произвольный конечный язык в V;

---

*) В этой работе используется термин цепочка вывода.

**) См. сноску на стр. 32.

г) язык $V_{sl}^{+}=\{x \mid x \in V^{+},\ |x|=si+l,\ i=0,\ 1,\ \ldots\}$ $(s,\ l \geqslant 1)$;

д) язык $\omega_1^{+}\omega_2^{+}\ \ldots\ \omega_n^{+}$, где $\omega_1,\ \ldots,\ \omega_n$ — произвольные непустые цепочки в V.

1.15. Построить Б-грамматики, порождающие:

а) множество правильно построенных формул логики высказываний в обычной «скобочной» записи (с фиксированным набором переменных);

б) то же для логики предикатов;

в) то же, что а), для бесскобочной записи;

г) то же, что б), для бесскобочной записи;

д) язык $\{a^n b^m \mid n,\ m=1,\ 2,\ \ldots;\ n<m\}$;

е) язык $\{x_1 b x_2 \mid x_1,\ x_2 \in \{a_1,\ \ldots,\ a_k\}^{*};\ |x_1|<|x_2|\}$;

ж) язык $\{a^{p+m}b^{m+n}c^{n+p} \mid m,\ n,\ p=1,\ 2,\ \ldots\}$;

з) язык $\{a^{n_1}b^{n_1},\ \ldots,\ a^{n_p}b^{n_p}c^{m_1}d^{m_1},\ \ldots,\ c^{m_p}d^{m_p} \mid p,\ m_i,\ n_i=1,\ 2,\ \ldots\}$;

и) «язык Дика» D_k — множество всех цепочек в алфавите $\{a_1,\ \ldots,\ a_k,\ \bar{a}_1,\ \ldots,\ \bar{a}_k\}$, равных единице в свободной группе со свободными образующими $a_1,\ \ldots,\ a_k$ (т. е. таких, которые могут быть преобразованы в Λ последовательным вычеркиванием вхождений подцепочек вида $a_i\bar{a}_i$ и $\bar{a}_i a_i$ (ср. ниже, § 6.5).

1.16. Для каждого из языков упражнений 1.14 и 1.15 построить допускающую его Э-машину.

1.17. Построить ДЭ-машины, распознающие следующие числовые множества (в простейшей записи):

а) множество четных чисел;

б) произвольную арифметическую прогрессию;

в) множество полных квадратов.

1.18. Построить ДЭ-машины, вычисляющие функции:

а) $f(x)=xdx$ (d — фиксированный символ);

б) $f(x)=x/x_0$ (x_0 — фиксированная цепочка).

1.19. Пусть ν — некоторая стандартная нумерация словаря V и $b \notin V$. Построить ДЭ-машины, вычисляющие следующие функции:

а) $\nu(x)$;

б) $\nu^{-1}(n)$;

в) $\nu^{-1}(\nu(x)+1)$;

г) $f(z)$, принимающую для каждой цепочки вида xby, где $x,\ y \in V^{*}$, значение $\nu^{-1}(\nu(x)+\nu(y))$;

д) $g(z)$, принимающую для каждой цепочки вида xby, где $x,\ y \in V^{*}$, значение $\nu^{-1}(\nu(x)\cdot\nu(y))$.

1.20. Машину Тьюринга с неэластичной рабочей лентой (Н-машину) можно определить так же, как Э-машину, с той только разницей, что рабочий алфавит дополняется «пустым символом» λ и инструкции видов (ii) и (iii) заменяются инструкциями вида (vi') $Aq_\alpha \to Bq_\beta$, где $q_\alpha, q_\beta \in Q$, $A,\ B \in W \cup \{\lambda\}$; такая инструкция означает замену в обозреваемой ячейке рабочей ленты символа A на символ B (без сдвига головки) и переход из состояния q_α в состояние q_β.

Одноленточная машина Тьюринга с неэластичной лентой (ОН-машина) отличается от Н-машины отсутствием входной ленты, входной головки, входного алфавита и инструкций типа (i).

Детерминированные Н- и ОН-машины (ДН-машины, ДОН-машины) определяются очевидным образом. (Машины Тьюринга, обычно рассматриваемые в курсах теории алгоритмов, — это ДОН-машины со следующим дополнительным ограничением: каково бы ни было незаключительное состояние q, программа машины либо содержит инструкцию (типа (iv) или (v)) с левой частью q, либо для каждого $A \in W$ содержит инструкцию (типа (vi)) с левой частью Aq.)

Ситуация, вычисление и x-вычисление для Н-машины определяются точно так же, как для Э-машины. Начальная и заключительная x-ситуация тоже определяются аналогично тому, как это делается для Э-машины, но последней компонентой вместо $\#$ будет $\#\lambda^k$, где k — произвольное натуральное число. x-вычисление и полное x-вычисление можно определить теперь так же, как для Э-машины; $[x,\ y]$-вычислением ($y \in W^{*}$) естественно назвать x-вычисление, оканчивающееся ситуацией вида $(q_f, \#x, \ddagger, \Lambda, \#y\lambda^s)$ $(q_f \in Q_0)$

Допускание языка и вычисление функции Н-машиной определяется так же, как для Э-машины.

Для ОН-машины ситуация определяется как тройка (q_α, x', x'') (обозначения сохраняют прежний смысл); язык $L \subseteq W^{*}$ допускается, если вычисление, начинающееся ситуацией вида $(q_1, \Lambda, x\lambda^t)$ и оканчивающееся ситуацией вида $(q_f, \Lambda, \lambda^u)$ $(q_f \in Q_0)$ существует тогда и только тогда, когда $x \in L$; ДОН-машина вычисляет функцию g: $L \to W^{*}$ $(L \subseteq W^{*})$, если вычисление, начинающееся ситуацией вида $(q_1, \Lambda, x\lambda^t)$ и оканчивающееся ситуацией вида $(q_f, \Lambda, y\lambda^u)$ $(q_f \in Q_0)$, существует тогда и только тогда, когда $y = g(x)$.

а) Перенести на Н- и ОН-машины теорему 1.2.

б) Перенести теорему 1.3 на ДН- и ДОН-машины, а также на ДОН-машины с указанным выше дополнительным ограничением.

в) Показать, что классы языков, допускаемых Э-, Н-, ДН-, ОН- и ДОН-машинами, а также ДОН-машинами с вышеуказанным ограничением, совпадают, причем для всякой машины каждого из перечисленных классов можно построить эквивалентную ей машину любого другого класса.

1.21. Назовем Э-машину числовой, если ее входной алфавит состоит из единственного символа |. Язык, допускаемый (функцию, вычисляемую) числовой Э(ДЭ)-машиной, назовем числовым рекурсивно перечислимым множеством (ч. р. п. м.), соответственно числовой частично рекурсивной функцией (ч. ч. р. ф.). Для произвольного алфавита V назовем язык $L \subseteq V^{*}$ нумерационно рекурсивно перечислимым множеством (н. р. п. м.) и функцию f: $T \to V^{*}$ $(T \subseteq V^{*})$ нумерационно частично рекурсивной функцией (н. ч. р. ф.), если существует такая стандартная нумерация ν алфавита V, что для некоторого ч. р. п. м. L' имеет место $L' = \nu(L)$ — соответственно для некоторой ч. ч. р. ф. f' имеет место $f(x) \equiv \nu^{-1}(f'(\nu(x)))$ (знак $\equiv$ означает совпадение областей определения и совпадение значений в каждой точке общей области определения).

Показать, что:

а) понятия н. р. п. м. и н. ч. р. ф. не зависят от выбора стандартной нумерации;

б) понятие н. р. п. м. не изменяется при расширении алфавита (точнее: если $L \subseteq V^*$ и $V \subseteq V'$, то свойство языка L быть н. р. п. м. не зависит от того, относительно которого из алфавитов V и V' определяются н. р. п. м.);

в) язык (функция) тогда и только тогда является н. р. п. м. (н. ч. р. ф.), когда он(а) допускается (вычисляется) некоторой Э-машиной.

1.22. Показать, что:

а) для всякой ДЭ-машины, распознающей язык L, можно построить ДЭ-машину, допускающую язык CL;

б) если имеются ДЭ-машины, допускающие языки L и CL, то по ним можно построить ДЭ-машину, распознающую L.

Замечание. Отсюда и из теоремы Поста следует, — поскольку, как уже было отмечено, ДЭ-машины допускают в точности рекурсивно перечислимые множества, — что существование распознающей язык ДЭ-машины равносильно рекурсивности этого языка.

ГЛАВА 2

СИГНАЛИЗИРУЮЩИЕ ФУНКЦИИ

§ 2.1. Сигнализирующие функции грамматик

Одним из наиболее существенных вопросов, возникающих при изучении порождающих грамматик (как в теоретическом, так и в прикладном аспекте), является вопрос об оценке сложности их работы, т. е. сложности выводов. Для оценки сложности выводов в грамматиках используются так называемые сигнализирующие функции, введением которых мы сейчас и займемся.

Пусть $f(\Gamma, D, i)$ — вычислимая функция, принимающая в качестве значений натуральные числа и определенная на всевозможных тройках (Γ, D, i), где $\Gamma = \langle V, W, I, R\rangle$ — грамматика (причем мы считаем, что основные и вспомогательные словари всех рассматриваемых грамматик содержатся в некоторых счетных множествах E и E_1 соответственно), $D = (\omega_0, \omega_1, \ldots, \omega_n)$ — вывод в Γ и $i = 0, 1, \ldots, n$. Для произвольной цепочки $x \in L(\Gamma)$ и произвольного вывода $D = (\omega_0, \omega_1, \ldots, \omega_n)$ цепочки x из I в Γ положим

$$\tilde{f}(\Gamma, D, x) = \max_{0 \leqslant i \leqslant n} f(\Gamma, D, i).$$

Далее, для любой цепочки $x \in L(\Gamma)$ положим $\tilde{F}(\Gamma, x) = \min \tilde{f}(\Gamma, D, x)$, где минимум берется по всем выводам цепочки x из I в Γ. Наконец, положим

$$F(\Gamma, n) = \max_{x \in L(\Gamma),\ |x| \leqslant n} \tilde{F}(\Gamma, x);$$

для тех n, для которых не существует таких $x \in L(\Gamma)$, что $|x| \leqslant n$, полагаем $F(\Gamma, n) = 0$.

Так определенную функцию $F(\Gamma, n)$ мы будем называть квазисигнализирующим оператором для грамматик, а функцию $F_\Gamma(n)$, получаемую из $F(\Gamma, n)$ фиксированием переменной Γ, — квазисигнализирующей функцией (или просто квазисигнализирующей) типа F грамматики Γ.

Таким образом, квазисигнализирующий оператор $F(\Gamma, n)$ сопоставляет каждой грамматике Γ числовую функцию $F_\Gamma(n)$. Если участвующая в определении оператора функция $f(\Gamma, D, i)$ выбрана так, чтобы она в каком-то смысле характеризовала «сложность» цепочки ω_i по отношению к ее «месту» в выводе D, то функцию $\bar{f}(\Gamma, D, x)$ — наибольшую «f-сложность» входящей в вывод цепочки — естественно считать мерой f-сложности вывода; тогда $\bar{F}(\Gamma, x)$ есть f-сложность самого простого вывода цепочки x, а $F(\Gamma, n)$ — наименьшее число N такое, что все принадлежащие $L(\Gamma)$ цепочки длины, не превосходящей n, можно вывести с f-сложностью, не превосходящей N. Таким образом, значение оператора F для конкретной грамматики, т. е. ее квазисигнализирующую типа F, можно считать определенной характеристикой сложности выводов в данной грамматике.

Однако действительно пригодные для произвольных грамматик характеристики сложности выводов доставляются лишь теми квазисигнализирующими операторами, которые удовлетворяют некоторому специальному условию (его значение станет нам ясно из дальнейшего). Это условие — рекурсивность графика соответствующей функции $\bar{F}(\Gamma, x)$, т. е. существование алгоритма, позволяющего по любой тройке (Γ, x, p), где Γ — грамматика, x — цепочка в ее основном словаре и p — натуральное число, узнать, выполняется ли равенство $\bar{F}(\Gamma, x) = p$. Квазисигнализирующий оператор $F(\Gamma, n)$, удовлетворяющий этому условию, мы будем называть сигнализирующим оператором для грамматик, а соответствующие квазисигнализирующие — сигнализирующими (функциями) типа F.

Может оказаться, что некоторый квазисигнализирующий оператор, не являющийся сигнализирующим в классе всех грамматик, становится таковым в некотором более частном классе. Точнее: пусть $F(\Gamma, n)$ — квазисигнализирующий оператор, $\mathscr{T}$ — некоторый класс грамматик и $F^{\mathscr{T}}(\Gamma, n)$ — функция, индуцируемая оператором F на $\mathscr{T}$. Если соответствующая функция $\bar{F}^{\mathscr{T}}$ имеет рекурсивный график, то мы будем называть $F^{\mathscr{T}}(\Gamma, n)$ относительным сигнализирующим оператором в классе $\mathscr{T}$, а функцию $F^{\mathscr{T}}_\Gamma(n)$, получаемую из $F^{\mathscr{T}}(\Gamma, n)$ фиксированием переменной Γ, — сигнализирующей (функцией) типа F грамматики Γ относительно $\mathscr{T}$.

Мы будем рассматривать следующие основные типы квазисигнализирующих.

1. Временна́я сложность (обозначение $T_\Gamma(n)$): соответствует случаю $f(\Gamma, D, i) = i$.

2. Емкость $S_\Gamma(n)$: получается при $f(\Gamma, D, i) = |\omega_i|$.

3. Левая глубина $\mathscr{Y}^{\text{л}}_\Gamma(n)$: $f(\Gamma, D, i)$ — увеличенное на единицу расстояние от самого левого вхождения в ω_i вспомогательного символа до правого конца ω_i; точнее, если $\omega_i = x_i A_i \eta_i$, где $x_i \in V^*$ и $A_i \in V_1$, то $f(\Gamma, D, i) = |\eta_i| + 1$; если $\omega_i \in V^*$, то $f(\Gamma, D, i) = 0$.

4. Правая глубина $\mathscr{Y}^{\text{п}}_\Gamma(n)$: определяется симметрично $\mathscr{Y}^{\text{л}}_\Gamma(n)$.

5. Разброс $D_\Gamma(n)$: $f(\Gamma, D, i)$ есть длина наименьшей подцепочки ω_i, содержащей все вхождения вспомогательных символов, если таковые имеются; в противном случае $f(\Gamma, D, i) = 0$.

6. Активная емкость $I_\Gamma(n)$: $f(\Gamma, D, i)$ есть число вхождений вспомогательных символов в ω_i.

Соответствующие квазисигнализирующие операторы будут обозначаться T, S, $\mathscr{Y}^{\text{л}}$, $\mathscr{Y}^{\text{п}}$, D, I*).

Содержательный смысл функций T_Γ и S_Γ ясен: первая характеризует сложность вывода числом его шагов, вторая — объемом затрачиваемой «памяти» (носителями «памяти» в выводе естественно считать его промежуточные цепочки). Легко видеть, что операторы T и S сигнализирующие. Действительно, пусть заданы грамматика Γ, цепочка x и натуральное число p. Чтобы узнать, имеет ли место равенство $\bar{T}(\Gamma, x) = p$, можно поступить следующим образом. Выпишем все полные выводы в Γ,

*) T, S, D, I — от time, space, dispersion, index (индекс — другое название активной емкости, см., например, [Стоцкий 1969]).

имеющие длину p. Если ни один из них не оканчивается цепочкой x, то заведомо $\tilde{T}(\Gamma, x) \neq p$. В противном случае выпишем все полные выводы в Г, длины которых меньше p. Если среди этих выводов найдется хотя бы один, оканчивающийся цепочкой x, то $\tilde{T}(\Gamma, x) \neq p$, а если таких выводов нет, то $\tilde{T}(\Gamma, x) = p$. Совершенно аналогично проверяется выполнение равенства $\tilde{S}(\Gamma, x) = p$; при этом вместо длины вывода $(\omega_0, \ldots, \omega_n)$ рассматривается число $\max\limits_{0 \leqslant i \leqslant n} |\omega_i|$ и берутся не произвольные выводы, а только бесповторные (иначе выводов с данным значением $\max\limits_{0 \leqslant i \leqslant n} |\omega_i|$ могло бы оказаться бесконечно много). Ограничиться бесповторными выводами можно в силу леммы 1.3 и того, что при выбрасывании из вывода каких-либо цепочек число $\max\limits_{0 \leqslant i \leqslant n} |\omega_i|$ не увеличивается.

Что касается операторов $\mathcal{Y}^{\text{Л}}$, $\mathcal{Y}^{\text{П}}$, D, I, то ни один из них сигнализирующим не является. Чтобы показать это, рассмотрим произвольно грамматику $\Gamma_0 = \langle V, W, I_0, R_0\rangle$ такую, что $V = \{|\}$ и $L(\Gamma_0) = M_0$ — нерекурсивное множество целых положительных чисел. Положим

$$\Gamma_1 = \langle \{a\} \cup V \cup W, \{I, b\}, I, R_0 \cup \{I \to I_0, | \to b, b \to aa\}\rangle,$$

где I, a, $b \notin V \cup W$. Добавим теперь к схеме грамматики Γ_1 правила $I \to I'$, $I' \to I'AA$, $I' \to BB$, $BB \to aa$, $aA \to aa$, где I', A, B — новые символы, которые будут считаться вспомогательными. Полученная грамматика Γ_2 порождает язык $L(\Gamma_1) \cup \{a^{2^n} | n > 1, 2, \ldots\}$, но при $n \in CM_0$ имеем $\tilde{\mathcal{Y}}^{\text{Л}}(\Gamma_2, a^{2^n}) = \tilde{\mathcal{Y}}^{\text{П}}(\Gamma_2, a^{2^n}) = \tilde{D}(\Gamma_2, a^{2^n}) = \tilde{I}(\Gamma_2, a^{2^n}) = 2n$, а при $n \in M_0$ ни одна из этих величин не превосходит n. Поэтому предположение о рекурсивности, например, графика функции $\tilde{I}(\Gamma, x)$ немедленно приводит к противоречию, так как из него вытекает существование алгоритма, позволяющего для любого $n = 1, 2, \ldots$ узнать, выполняется ли равенство $\tilde{I}(\Gamma_2, a^{2^n}) = 2n$, т. е. принадлежит ли число n множеству CM_0.

Впрочем, для некоторых важных классов грамматик индуцируемые операторами $\mathcal{Y}^{\text{Л}}$, $\mathcal{Y}^{\text{П}}$, D, I относительные операторы будут сигнализирующими. В частности, это

имеет место для НС-грамматик (см. § 3.3); именно ради применения к НС-грамматикам (в основном даже к Б-грамматикам, см. гл. 7) мы и рассматриваем указанные операторы.

Для произвольной грамматики $\Gamma = \langle V, W, I, R\rangle$, порождающей бесконечный язык, имеют место неравенства

$$S_\Gamma(n) \geqslant \max(n, 1); \quad T_\Gamma(n) \geqslant 1; \tag{1}$$

$$\begin{aligned} &I_\Gamma(n) \leqslant D_\Gamma(n) \leqslant \mathcal{Y}^{\text{Л}}_\Gamma(n) \leqslant S_\Gamma(n) \leqslant dT_\Gamma(n) + 1, \\ &I_\Gamma(n) \leqslant D_\Gamma(n) \leqslant \mathcal{Y}^{\text{П}}_\Gamma(n) \leqslant S_\Gamma(n) \leqslant dT_\Gamma(n) + 1; \end{aligned} \tag{2}$$

$$T_\Gamma(n) < \frac{k^{S_\Gamma(n)+1} - 1}{k - 1}. \tag{3}$$

Здесь $d = \max\limits_{\varphi \to \psi \in R} ||\psi| - |\varphi||$ и k — мощность множества $V \cup W$.

Неравенства (1) и (2) непосредственно следуют из определений (для получения последнего из неравенств (2) надо еще воспользоваться тем очевидным фактом, что ни на каком шаге вывода цепочка не может удлиниться более чем на d). (3) вытекает из леммы 1.3, поскольку для бесповторного вывода $(\omega_0, \ldots, \omega_n)$ такого, что $\max\limits_{0 \leqslant i \leqslant n} |\omega_i| = l$, число n во всяком случае меньше числа различных цепочек в словаре $V \cup W$ длины, не большей l, а это число равно $\frac{k^{l+1} - 1}{k - 1}$.

Отметим следующее свойство сигнализирующих.

Лемма 2.1. *Для произвольной грамматики* Г *следующие утверждения равносильны*: (1) *язык* $L(\Gamma)$ *рекурсивен*; (2) *любая сигнализирующая грамматики* Г *вычислима* *); (3) *любая сигнализирующая грамматики* Г *мажорируется* **) *подходящей вычислимой функцией*; (4) *хотя бы одна сигнализирующая грамматики* Г *вычислима*; (5) *хотя бы одна сигнализирующая граммати-*

*) Поскольку сигнализирующая любой грамматики есть всюду определенная функция, вычислимость здесь и далее можно заменить общей рекурсивностью.

**) Мы говорим, что функция $g(n)$ мажорируется функцией $h(n)$, если для каждого n, для которого $g(n)$ определена, $h(n)$ также определена и $g(n) \leqslant h(n)$.

ки Γ *мажорируется вычислимой функцией. При этом по алгоритму, требуемому каким-либо одним из перечисленных утверждений, можно построить алгоритмы, требуемые остальными.*

Доказательство. Достаточно доказать импликации (1) $\supset$ (2) и (5) $\supset$ (1). Пусть язык $L(\Gamma)$ рекурсивен и $F_\Gamma(n)$ — какая-либо сигнализирующая Γ. Для произвольной цепочки x в основном словаре Γ мы можем узнать, принадлежит ли она $L(\Gamma)$, и в случае положительного ответа найти значение $\bar{F}(\Gamma, x)$, последовательно проверяя равенства $\bar{F}(\Gamma, x) = 0$, $\bar{F}(\Gamma, x) = 1$ и т. д. Таким образом, функция $\bar{F}(\Gamma, x)$ (с фиксированной Γ), а следовательно и $F(\Gamma, n)$, оказывается вычислимой. Пусть теперь некоторая сигнализирующая $F_\Gamma(n)$ грамматики Γ мажорируется (всюду определенной) вычислимой функцией $h(n)$. Тогда для каждой цепочки $x \in L(\Gamma)$ имеет место $\bar{F}(\Gamma, x) \leqslant h(|x|)$; в то же время при $x \notin L(\Gamma)$ функция $\bar{F}(\Gamma, x)$ не определена. Поэтому $x \in L(\Gamma)$ тогда и только тогда, когда выполняется хотя бы одно из равенств $\bar{F}(\Gamma, x) = 0, \ldots, \bar{F}(\Gamma, x) = h(|x|)$, а они поддаются проверке в силу рекурсивности графика $\bar{F}(\Gamma, x)$.

Заметим, что если для некоторого сигнализирующего оператора $F(\Gamma, x)$, некоторой вычислимой функции $h(n)$ и некоторого класса грамматик $\mathscr{T}$ имеет место $F(\Gamma, n) \leqslant h(|x|)$, какова бы ни была грамматика $\Gamma \in \mathscr{T}$, то указанный только что метод доказательства импликации (5) $\supset$ (1) дает единый алгоритм, позволяющий для любой грамматики $\Gamma \in \mathscr{T}$ и для любой цепочки x в основном словаре этой грамматики распознать, принадлежит ли x языку $L(\Gamma)$.

В дальнейшем нам будут полезны следующие обозначения. Пусть $\mathscr{T}$ — некоторый класс грамматик, F — квазисигнализирующий оператор и $\mathfrak{F}$ — класс числовых функций. Тогда через $\mathscr{L}^F_{\mathfrak{F}}(\mathscr{T})$ мы будем обозначать класс языков, порождаемых такими грамматиками класса $\mathscr{T}$, у которых квазисигнализирующие типа F мажорируются какими-либо функциями класса $\mathfrak{F}$. Если $\mathfrak{F}$ состоит из одной функции f, будем вместо $\mathscr{L}^F_{\mathfrak{F}}(\mathscr{T})$ писать $\mathscr{L}^F_f(\mathscr{T})$. Кроме того, вместо $\mathscr{L}^F_{\mathfrak{F}}(\Gamma)$ (Γ — класс всех грамматик, см стр. 30) мы будем писать $\mathscr{L}^F_{\mathfrak{F}}$.

Наконец, $L^F_k(\Gamma)$, где Γ — грамматика, F — квазисигнализирующий оператор и k — натуральное число, будет означать множество цепочек $x \in L(\Gamma)$, удовлетворяющих условию $\bar{F}(\Gamma, x) \leqslant k$.

Замечание. Обратившись к конструкции, примененной в доказательстве леммы 1.4, легко увидеть, что для фигурирующих там грамматик Γ и Γ′ имеют место соотношения

$$S_{\Gamma'}(n) = S_\Gamma(n), \quad T_{\Gamma'}(n) \leqslant kT_\Gamma(n),$$

где k — постоянная, зависящая от Γ (и допускающая эффективное вычисление). Действительно, если D — полный вывод цепочки x в Γ и D' — соответствующий полный вывод x в Γ′, то максимальные длины цепочек в D и в D' совпадают, а длина самого вывода D' равна сумме длины D и длины x; но длина x не превосходит длины D, умноженной на максимум длин правых частей правил Γ.

Нам понадобится также

Лемма 2.2. *Для произвольной грамматики* Γ *можно построить эквивалентную ей грамматику* Γ′, *удовлетворяющую следующим условиям*:

а) Γ′ *не имеет правил с пустой левой частью.*

б) *Для любого полного вывода* D *в* Γ, *оканчивающегося цепочкой* x, *можно построить равносильный ему полный вывод* D' *в* Γ′, *отличающийся от* D *только тем, что на тех местах (кроме последнего, если* $x = \Lambda$*), на которых в* D *стоит* Λ, *в* D' *стоит цепочка из одного символа* E, *не встречающегося в остальных цепочках* D. *При этом* D' *можно разметить так, чтобы при* $x \neq \Lambda$ *правые части всех применяемых правил были непусты, а при* $x = \Lambda$ *правило с пустой правой частью применялось только на последнем шаге. Если* D — *упорядочиваемый вывод, то* D' — *также упорядочиваемый вывод. Обратно, любой вывод в* Γ′ *можно получить из некоторого вывода в* Γ *указанным способом.*

в) $S_{\Gamma'}(n) = S_\Gamma(n)$; $T_{\Gamma'}(n) = T_\Gamma(n)$.

Если $\Lambda \notin L(\Gamma)$, *то можно сверх сказанного потребовать, чтобы*

а′) Γ′ *не имела правил с пустой правой частью.*

Доказательство. Пусть $\Gamma = \langle V, W, I, R\rangle$. Каждому правилу из R вида $\Lambda \to \psi$, соответственно $\varphi \to \Lambda$,

сопоставим всевозможные правила $\alpha \to \alpha\psi$ и $\alpha \to \psi\alpha$, соответственно $\alpha\varphi \to \alpha$, $\varphi\alpha \to \alpha$, где $\alpha \in V \cup W$, а также правило $E \to \psi$, соответственно $\varphi \to E$, где E — символ, не входящий в $V \cup W$. Присоединим к W символ E и к R — все новые правила; одновременно удалим из R все правила вида $\Lambda \to \psi$. Полученная грамматика Γ' удовлетворяет условию а). Выполнение условия б) легко проверяется непосредственно; условие в), равно как и эквивалентность Γ и Γ', следует из б). Наконец, в силу а) и б) при $\Lambda \notin L(\Gamma)$ из схемы Γ' можно удалить правила вида $\varphi \to \Lambda$, что дает а').

§ 2.2. Сигнализирующие функции машин Тьюринга

Пусть $f(M, C, i)$ — вычислимая функция, принимающая в качестве значений натуральные числа и определенная на всевозможных тройках (M, C, i), где M — Э-машина, $C = (s_0, s_1, \ldots, s_n)$ — вычисление машины M и $i = 0, 1, \ldots, n$. (Алфавиты и множества состояний всех рассматриваемых машин считаем содержащимися в некотором множестве E.) Для произвольной цепочки $x \in L(M)$ и произвольного полного x-вычисления $C = (s_0, s_1, \ldots, s_n)$ машины M положим $\tilde{f}(M, C, x) = \max\limits_{0 \leqslant i \leqslant n} f(M, C, i)$. Для любой цепочки $x \in L(M)$ положим $\tilde{F}(M, x) = \min \tilde{f}(M, C, x)$, где минимум берется по всем полным x-вычислениям машины M. Наконец, положим $F(M, n) = \max\limits_{x \in L(M);\ |x| \leqslant n} \tilde{F}(M, x)$; для тех n, для которых не существует таких $x \in L(\Gamma)$, что $|x| \leqslant n$, полагаем $F(M, n) = 0$.

Так определенную функцию $F(M, n)$ мы будем называть квазисигнализирующим оператором для Э-машин, а функцию $F_M(n)$, получаемую из $F(M, n)$ фиксированием M,—квазисигнализирующей (функцией) типа F машины M.

Если M — ДЭ-машина, то можно определить $F_M(n)$ непосредственно как $\max\limits_{x \in L(M);\ |x| \leqslant n} f(M, C, x)$, поскольку в этом случае цепочка $x \in L(M)$ имеет единственное полное вычисление.

В случае, когда график функции $\tilde{F}(M, x)$ рекурсивен, мы будем называть соответствующий оператор $\tilde{F}(M, n)$ сигнализирующим оператором для Э-машин и функцию $F_M(n)$ — сигнализирующей (функцией) типа F машины M.

Нас будут интересовать два типа сигнализирующих функций Э-машин: время работы — обозначение $T_M(n)$ — и емкость — обозначение $S_M(n)$. При определении этих функций в качестве $f(M, C, i)$ берется соответственно число i и длина рабочей ленты в ситуации s_i (т. е. число ее ячеек, не считая ячейки, занятой граничным маркером). Тот факт, что соответствующие операторы действительно являются сигнализирующими, устанавливается точно так же, как для грамматик (§ 2.1).

Аналогично неравенствам (1), (2), (3) из § 2.1 нетрудно получить следующие неравенства (доказательство предоставляется читателю):

(1′) $$T_M(n) \geqslant n + 1,$$

(2′) $$S_M(n) \leqslant \tfrac{1}{2}(T_M(n) - (n+1)),$$

(3′) $$T_M(n) < r(n+1)(S_M(n)+1) \cdot \frac{k^{S_M(n)+1} - 1}{k-1},$$

где k, r — мощности рабочего алфавита и множества состояний машины M соответственно.

В главе 1 (теорема 1.2) мы доказали, что для каждой Э-машины M можно построить ДЭ-машину M_1 такую, что $L(M_1) = L(M)$. Ясно, что при «детерминизации» машины время вычисления и число используемых ячеек будут, вообще говоря, расти. Просмотр конструкции, примененной в доказательстве теоремы 1.2, дает возможность оценить этот рост. Именно, пусть $x \in L(M)$, и при этом то полное x-вычисление C машины M, имитацией которого заканчивается (единственное) полное x-вычисление C_1 машины M_1, имеет длину t, а максимальная длина рабочей ленты в вычислении C равна s. Соответствующие величины для вычисления C_1 обозначим t_1 и s_1. На каждом «макрошаге» вычисления C_1 имитируется некоторое x-вычисление машины M, длина которого t' не превосходит t; поэтому длина участка рабочей ленты между d и началом Z на данном макрошаге никогда не будет больше t (в начале макрошага она равна 0 и на каждом «M-шаге» может возрасти разве что на 1), а длина Z в точности равна t'. Отсюда

$s_1 \leqslant |x| + 2t + 3$. Далее, каждый макрошаг состоит из t' «M-шагов», каждый из которых требует не более $2(|x| + 2t' + 3 + D)$ элементарных шагов машины M_1 (D — постоянная; при выполнении «M-шага» головка должна «прогуляться» из начала цепочки длины не более $|x| + 2t' + 3$ в ее конец и обратно, выполнив «по дороге» ограниченное число специальных операций — собственно «M-шаг», перенос метки и т. п.); после выполнения последнего «M-шага» для завершения макрошага может понадобиться еще не более $|x| + 2t' + 3 + D_1$ элементарных шагов (D_1 — постоянная). Всех макрошагов будет не больше, чем различных цепочек в алфавите P длины, не превосходящей t, т. е. $\frac{h^{t+1}-1}{h-1}$; наконец, до начала первого макрошага машина должна сделать $2(|x| + 4)$ шагов. Это дает $t_1 \leqslant Ah^{t+1}t(B_1t + B_2|x| + B_3)$, где A, B_1, B_2, B_3 — постоянные.

Отсюда непосредственно следует

Лемма 2.3. *Для любой Э-машины M можно построить ДЭ-машину M_1, допускающую язык $L(M)$ и такую, что*

$$T_{M_1}(n) \leqslant Ah^{T_M(n)+1} \cdot T_M(n) \cdot (B_1T_M(n) + B_2n + B_3),$$

$$S_{M_1}(n) \leqslant 2T_M(n) + n + 3,$$

где h — число инструкций машины M и A, B_1, B_2, B_3 — постоянные, зависящие от M.

§ 2.3. Ускорение и сжатие выводов. Связь между сигнализирующими грамматик и машин

В построениях этого параграфа важную роль будет играть «лемма о сцеплении», которую мы сейчас сформулируем и докажем, введя предварительно понятие сцепленной грамматики.

Грамматика называется сцепленной, если в каждом ее размеченном полном выводе каждый следующий шаг зацеплен с предыдущим.

Лемма 2.4 (о сцеплении). *Для произвольной грамматики Γ можно построить эквивалентную ей сцепленную грамматику Γ' такую, что*

$$S_{\Gamma'}(n) = S_\Gamma(n), \quad T_{\Gamma'}(n) \leqslant kT_\Gamma(n),$$

где k — постоянная, зависящая от Γ (и допускающая эффективное вычисление).

Доказательство. Пусть $\Gamma = \langle V, W, I, R\rangle$. В силу леммы 2.2 мы можем считать, что левые части правил Γ не пусты, и рассматривать только такие выводы в Γ, в которых правило с пустой правой частью применяется разве что на последнем шаге. Сопоставим каждому символу $\alpha \in V \cup W$ взаимно однозначным образом двух «двойников» — новые символы $\bar\alpha$ и α^0; множества «двойников» обозначим соответственно $\bar Z$ и Z^0. Каждому правилу

$$\beta_1 \ldots \beta_p \to \gamma_1 \ldots \gamma_q \in R \quad (\beta_t, \gamma_j \in V \cup W)$$

сопоставим множество, состоящее из всевозможных правил вида

$$\beta_1^0 \ldots \beta_{i-1}^0 \bar\beta_i \beta_{i+1}^0 \ldots \beta_p^0 \to \bar\gamma_1 \gamma_2^0 \ldots \gamma_q^0, \qquad i = 1, \ldots, p$$

(правила 1-го рода). Каждой паре $\alpha, \beta \in V \cup W$ сопоставим правило $\bar\alpha\beta^0 \to \alpha^0\bar\beta$ (правило 2-го рода), каждому $a \in V$ — правило $\bar a \to a$ (правило 3-го рода), каждой паре $a, b \in V$ — правило $a^0b \to ab$ (правило 4-го рода). Положим $\Gamma' = \langle V, \bar Z \cup Z^0, I, R'\rangle$, где R' — совокупность всех правил 1-го—4-го рода.

Сцепленность Γ' легко проверяется непосредственно. Докажем эквивалентность Γ и Γ'. Для каждого упорядочиваемого вывода $D = (\omega_0, \ldots, \omega_n)$ в Γ легко построить вывод в Γ', состоящий из n последовательных кусков:

$$(\eta_0, \ldots, \eta_{i_1}), (\eta_{i_1+1}, \ldots, \eta_{i_2}), \ldots, (\eta_{i_{n-1}+1}, \ldots, \eta_{i_n}),$$

причем для каждого $j = 1, \ldots, n$ цепочка η_{i_j} отличается от ω_j только наличием черты над одним из вхождений символов и верхнего индекса 0 у остальных. При этом каждый кусок начинается применением правила 1-го рода, а на всех остальных шагах куска, если таковые есть, применяются правила 2-го рода. Поскольку $\omega_n \in V^*$, из η_{i_n} можно с помощью правил 2-го, 3-го и 4-го рода вывести ω_n. (Сначала следует «загнать» черточку в конец цепочки, затем применить правило 3-го рода, далее пользоваться правилами 4-го рода.)

Ясно также, что цепочку в словаре V можно вывести в Γ из I только вышеописанным образом. Ввиду леммы 1.2 из сказанного вытекает эквивалентность Γ и Γ', а также равенство $S_{\Gamma'}(n) = S_\Gamma(n)$. Остается оценить $T_{\Gamma'}$. Пусть $D' = (\eta_0, \ldots, \eta_m)$ — произвольный полный вывод в Γ', и пусть m_k $(k = 1, 2, 3, 4)$ — число шагов этого вывода, на которых применяются правила рода k (шагов рода k). Нам достаточно найти такую не зависящую от D' постоянную l, чтобы выполнялось неравенство $m_2 \leqslant l m_1$. Действительно, длина соответствующего D' вывода в Γ равна m_1, а так как $m_3 = 1$ и $m_4 = |\eta_m| - 1$, мы получим $m = m_1 + m_2 + m_3 + m_4 \leqslant (l+1) m_1 + |\eta_m| \leqslant (l + f + 1) m_1$ (f — максимум длин левых и правых частей правил Γ).

Пусть η_{i_0} — последняя цепочка вывода D', содержащая вхождение символа из $\bar{Z}$. Шаги 1-го и 2-го рода — это все шаги с номерами, не бо́льшими i_0, и только они. Для каждого $i = 0, 1, \ldots, i_0$ имеем $\eta_i = \xi_i^0 \bar{\alpha}_i \zeta_i^0$, где $\xi_i^0, \zeta_i^0 \in Z^{0*}$ и $\bar{\alpha} \in \bar{Z}$. Положим $g_i = |\xi_i^0|$, $h_i = g_i - g_{i-1}$. Если i-й шаг 2-го рода, то $h_i = 1$, а если 1-го, то $-e < h_i \leqslant 0$, где e — максимум длин левых частей правил Γ. Очевидно,

$$\sum_{i=0}^{i_0} h_i = g_{i_0} - g_0 = g_{i_0}.$$

Отсюда, обозначая через Σ_k сумму h_i по всем шагам рода k, получаем

$$m_2 = \Sigma_2 = g_{i_0} - \Sigma_1 < g_{i_0} + e m_1 < |\eta_{i_0}| + e m_1 = |\eta_m| + e m_1 \leqslant (f + e) m_1.$$

Доказательство закончено.

Теперь мы можем показать, что для произвольной грамматики имеется возможность «сжимать» и «ускорять» ее выводы равномерным образом в любом наперед заданном отношении, лишь бы не нарушились неравенства (1) (стр. 59); для этого, разумеется, нужно перейти к другим, эквивалентным грамматикам. Стоит заметить (см. ниже доказательства теорем 2.1, 2.2), что эти новые грамматики, вообще говоря, сложнее старой — имеют больше вспомогательных символов или

правил, и сами правила могут быть длиннее, — причем усложнение грамматики оказывается тем сильнее, чем больше мы захотим упростить выводы, т. е. «сжать» или «ускорить» их. Это обычная ситуация — за простоту и удобство пользования устройством приходится платить усложнением его самого.

Теорема 2.1 (о сжатии выводов). *Для любой грамматики Γ и любого положительного действительного числа c можно построить грамматику Γ', эквивалентную Γ и такую, что*

$$S_{\Gamma'}(n) \leqslant \max(n, 1, cS_\Gamma(n));$$
$$T_{\Gamma'}(n) \leqslant T_\Gamma(n) + \max(cn, 1).$$

Доказательство. Пусть $\Gamma = \langle V, W, I, R\rangle$. В силу леммы 2.2 мы можем считать, что левые части правил Γ не пусты, и рассматривать только такие выводы в Γ, в которых правило с пустой правой частью применяется разве что на последнем шаге.

Пусть l — натуральное число, большее $1/c$ и одновременно большее $\max\limits_{\varphi \to \psi \in R} |\varphi|$ и $\max\limits_{\varphi \to \psi \equiv R} ||\psi| - |\varphi||$. Сопоставим каждой цепочке $\omega \in (V \cup V_1)^*$, $|\omega| \leqslant 2l$, взаимно однозначным образом новый символ $\alpha(\omega)$. Множество таких символов обозначим Z.

Каждому правилу $\varphi \to \psi \in R$, каждой цепочке $\omega \in (V \cup W)^*$, $|\omega| \leqslant 2l$, содержащей вхождения φ, и каждому вхождению $\xi * \varphi * \eta$ цепочки φ в ω сопоставим правило $f = f(\varphi, \psi, \xi, \eta)$ следующим образом: если $|\xi\psi\eta| \leqslant 2l$, то $f = \alpha(\omega) \to \alpha(\xi\psi\eta)$; если же $|\xi\psi\eta| > 2l$, то $f = \alpha(\omega) \to \alpha(\zeta_1)\alpha(\zeta_2)$, где $\zeta_1\zeta_2 = \xi\psi\eta$ и $|\zeta_1| = l$ ($\alpha(\zeta_2)$ существует, поскольку $|\xi\psi\eta| < 3l$).

Далее, каждому правилу $\varphi \to \psi \in R$, каждой паре цепочек ω_1, $\omega_2 \in (V \cup W)^*$, $l \leqslant |\omega_1| \leqslant 2l$, $l \leqslant |\omega_2| \leqslant 2l$, такой, что $\omega_1\omega_2$ содержит вхождения φ, и каждому вхождению $\xi * \varphi * \eta$ цепочки φ в $\omega_1\omega_2$ сопоставим правило $g = g(\varphi, \psi, \xi, \eta, \omega_1, \omega_2)$ следующим образом: если $|\xi\psi\eta| < 2l$ (заметим, что в любом случае $|\xi\psi\eta| > l$), то $g = \alpha(\omega_1)\alpha(\omega_2) \to \alpha(\xi\psi\eta)$; если $2l \leqslant |\xi\psi\eta| \leqslant 4l$, то $g = \alpha(\omega_1)\alpha(\omega_2) \to \alpha(\zeta_1)\alpha(\zeta_2)$, где $\zeta_1\zeta_2 = \xi\psi\eta$ и $|\zeta_1| = \left[\frac{1}{2}|\xi\psi\eta|\right]$ ([ ] — целая часть); если, наконец, $|\xi\psi\eta| > 4l$, то $g = \alpha(\omega_1)\alpha(\omega_2) \to \alpha(\zeta_1)\alpha(\zeta_2)\alpha(\zeta_3)$, где $\zeta_1\zeta_2\zeta_3 = \xi\psi\eta$, $|\zeta_1| \models 2l$, $|\zeta_2| = l$ ($\alpha(\zeta_3)$ существует, так как $|\xi\psi\eta| < 5l$).

Сопоставим также каждой цепочке ω, $|\omega| \leqslant 2l$, правило $h = h(\omega)$: $\alpha(\omega) \to \omega$.

Положим теперь $\Gamma' = \langle V, Z, \alpha(I), R' \rangle$, где R' состоит из всех правил f, g и h. Покажем, что Γ' — нужная грамматика.

Назовем образом цепочки $\chi \in (V \cup W)^*$ любую цепочку $\theta = \alpha(\omega_1) \ldots \alpha(\omega_k)$ такую, что $\omega_1 \ldots \omega_k = \chi$ и при $k > 1$ для каждого $j = 1, \ldots, k$ имеет место $|\omega_j| \geqslant l$. Из определения правил f и g ясно, что если из цепочки χ_1 непосредственно выводима в Γ цепочка χ_2, то из каждого образа θ_1 цепочки χ_1 будет непосредственно выводим в Γ' некоторый образ цепочки χ_2 (с помощью некоторого правила g, если $|\theta_1| > 1$, и правила f, если $|\theta_1| = 1$). Поэтому для каждого полного вывода $D = (\chi_0, \ldots, \chi_n)$ в Γ существует вывод $D' = (\theta_0, \ldots, \theta_n)$ в Γ' такой, что θ_i есть образ χ_i для каждого $i = 0, \ldots, n$ (в частности, $\theta_0 = \alpha(I)$); при этом в силу определения образа $|\theta_i| \leqslant \max\left(1, \frac{1}{l}|\chi_i|\right)$. Из θ_n можно с помощью правил h вывести χ_n, причем длины промежуточных цепочек в этом выводе не будут превосходить $\max(1, |\chi_n|)$. Таким образом, мы получаем вывод χ_n из $\alpha(I)$ в Γ', в котором максимальная длина цепочки не превосходит $\max\left(|\chi_n|, 1, \frac{1}{l} \max\limits_{0 \leqslant i \leqslant n} |\chi_i|\right)$, причем число шагов этого вывода не больше $n + \max\left(\frac{1}{l}|\chi_n|, 1\right)$ (число применений правил f и g равно n; число применений правил h не превосходит $\frac{1}{l}|\chi_n|$ при $|\chi_n| \geqslant l$ и единицы при $|\chi_n| < l$). В то же время — по лемме 1.1 — всякая выводимая в Γ' цепочка в словаре V может быть выведена так, что сначала будут применяться правила f и g, а потом h; отсюда ясно, что эта цепочка может быть выведена и в Γ. Итак, Γ' есть нужная грамматика.

Следствие. *Для любой грамматики* Γ *и любого натурального числа* k *можно построить грамматику* Γ', *эквивалентную* Γ *и такую, что*

$$S_{\Gamma'}(n) \leqslant \max(n, 1, S_\Gamma(n) - k);$$
$$T_{\Gamma'}(n) \leqslant T_\Gamma(n) + \max(n, 1).$$

Теорема 2.2 (об ускорении выводов). *Для любой грамматики* Γ *и любого положительного действительного числа* c *можно построить грамматику* Γ', *эквивалентную* Γ *и такую, что*

$$T_{\Gamma'}(n) \leqslant \max(1, cT_\Gamma(n));$$
$$S_{\Gamma'}(n) \leqslant S_\Gamma(n).$$

Доказательство. Пусть $\Gamma = \langle V, W, I, R \rangle$. В силу леммы о сцеплении мы можем считать грамматику Γ сцепленной. Фиксируем натуральное число l, большее $1/c$, и рассмотрим всевозможные размеченные выводы в грамматике Γ: $D' = (\omega'_0, \ldots, \omega'_{n-1}, \omega_n)$, $\omega'_i = \xi_i * \varphi_i * \eta_i$, удовлетворяющие условиям: (α) $1 \leqslant n \leqslant 2l$; ($\beta$) существуют такие i_1 и i_2 из ряда $0, \ldots, n-1$, что $\xi_{i_1} = \eta_{i_2} = \Lambda$. Непосредственной индукцией по n легко показать, что для любого $i = 0, \ldots, n-1$ справедливо $|\xi_i \varphi_i \eta_i| \leqslant ng \leqslant 2lg$, где g — максимум длин левых и правых частей правил Γ (здесь используется сцепленность Γ!). Поэтому число размеченных выводов, удовлетворяющих условиям (α) и (β), конечно. Сопоставим каждому такому размеченному выводу правило $p = \xi_0 \varphi_0 \eta_0 \to \omega_n$ и положим $\Gamma' = \langle V, W, I, R' \rangle$, где R' состоит из всех p.

Пусть $D = (\omega_0, \ldots, \omega_m)$, $m \geqslant l$, — полный вывод в Γ. Разобьем D на куски, длина каждого из которых не меньше l и не больше $2l$; число кусков будет не больше m/l. Но каждый из этих кусков вывода можно, очевидно, выполнить с помощью одного из правил p; мы получаем, таким образом, полный вывод в Γ' длины, не большей m/l, с той же последней цепочкой ω_m. Вывод в Γ длины, меньшей l, можно заменить выводом в Γ' длины 1. С другой стороны, из определения правил p ясно, что каждый вывод в Γ' можно заменить выводом в Γ. Неравенство $S_{\Gamma'}(n) \leqslant S_\Gamma(n)$ очевидно. Итак, Γ' — нужная грамматика.

Следствие. *Для любой грамматики* Γ *и любого натурального числа* k *можно построить грамматику* Γ', *эквивалентную* Γ *и такую, что*

$$T_{\Gamma'}(n) \leqslant \max(1, T_\Gamma(n) - k);$$
$$S_{\Gamma'}(n) \leqslant S_\Gamma(n).$$

Для машин Тьюринга также имеют место теоремы о сжатии и ускорении, а именно:

Теорема 2.1′ (о сжатии вычислений). *Для любой Э-машины M и любого положительного действительного числа c можно построить Э-машину M', эквивалентную M и такую, что*

$$S_{M'}(n) \leqslant \max(1,\ cS_M(n));$$
$$T_{M'}(n) \leqslant T_M(n).$$

Если при этом M — ДЭ-машина, то и M' можно сделать ДЭ-машиной.

Теорема 2.2′ (об ускорении вычислений). *Для любой Э-машины M и любого положительного действительного числа c можно построить Э-машину M', эквивалентную M и такую, что*

$$T_{M'}(n) \leqslant (n+1) + c\,(T_M(n) - (n+1));$$
$$S_{M'}(n) \leqslant S_M(n).$$

Если при этом M — ДЭ-машина, то и M' можно сделать ДЭ-машиной.

Теорема 2.1′ доказывается совершенно аналогично теореме 2.1. Что касается теоремы 2.2′, то ее доказательство аналогично доказательству теоремы 2.2 со следующими отличиями: а) не требуется предварительной переделки машины (в случае грамматики переделка обеспечивается леммой 2.4); б) любая Э-машина при обработке цепочки длины n должна прочесть ее с начала до конца — это и приводит к выделению слагаемого $n+1$; в) поскольку никакой элементарный шаг Э-машины не затрагивает более двух ячеек, при построении новой машины придется прибегнуть к кодированию цепочек в рабочем алфавите (аналогично тому, как это делается при доказательстве теоремы о сжатии).

Займемся теперь вопросом о соотношении между сигнализирующими грамматик и машин. Обратимся сначала к доказательству пункта а) теоремы 1.4. Нетрудно видеть (см. доказательство леммы 1.5), что если машина M, работая с цепочкой длины n, затрачивает $t = n + t'$ шагов и максимальная длина рабочей ленты будет при этом равна s, то для M_1 при обработке той же цепочки (подходящим способом) максимальная длина

рабочей ленты s_1 и число шагов t_1 будут удовлетворять условиям: $s_1 = s + n + 1$, $t_1 \leqslant (s+n+1)\min(n, t')c + t$ ($\min(n, t')$ — максимально возможное число «переключений» с имитации «входной» команды на имитацию «рабочей»; c — постоянная). Соответствующие величины для M_2 будут таковы: $s_2 = s_1$, $t_2 \leqslant t_1 + n + c_1 \leqslant (s+n+1)\min(n, t')c_2 + t$ (c_1, c_2 — постоянные). Наконец, длина (подходящего) вывода той же цепочки в грамматике Γ и максимальная длина цепочки, входящей в этот вывод, будут равны $t_2 + 2$ и $s_2 + 1$ соответственно.

Обратившись теперь к доказательству пункта б) теоремы 1.4, мы увидим, что если длина вывода цепочки $x, |x| = n$, в Γ равна t, а максимальная цепочка этого вывода имеет длину s, то в M найдется x-вычисление, длина которого не превосходит $d_1 n + d_2 t \leqslant d_3 t$ (d_1, d_2, d_3 — постоянные; d_1 шагов нужно на переписывание входного символа на рабочую ленту, d_2 шагов — на имитацию применения одного правила), причем максимальная длина фигурирующей в нем рабочей ленты равна s.

Отсюда с учетом теорем 2.1, 2.2, 2.2′ и того факта, что все упоминавшиеся постоянные эффективно определимы по соответствующей грамматике (машине), получается

Теорема 2.3. а) *Для любой Э-машины M можно построить эквивалентную грамматику Γ такую, что*

$$S_\Gamma(n) \leqslant S_M(n);$$
$$T_\Gamma(n) \leqslant (S_M(n) + n) \cdot \min(n,\ T_M(n) - n) + T_M(n).$$

б) *Для любой грамматики Γ можно построить эквивалентную Э-машину M такую, что*

$$S_M(n) = S_\Gamma(n);$$
$$T_M(n) \leqslant \max((n+1),\ T_\Gamma(n)).$$

§ 2.4. Существование сколь угодно сложных рекурсивных языков

Если F — некоторый достаточно «хороший» сигнализирующий оператор для грамматик и f — вычислимая функция, то факт принадлежности какого-либо языка классу $\mathscr{L}_f^F$ может рассматриваться как определенная характеристика сложности этого языка: если $L_1 \in \mathscr{L}_f^F$ и

$L_2 \notin \mathscr{L}_f^F$, то L_2 сложнее L_1, поскольку при некотором способе порождения L_1 можно обойтись более простыми выводами, чем те, которые понадобятся при любом способе порождения L_2.

Для сигнализирующих наиболее важных типов — T и S — правомерность указанного подхода к определению сложности становится особенно наглядной, если обратить внимание на следующее: при $F = T$, $f(n) \geqslant 1$ и при $F = S$, $f(n) \geqslant n$ из $L \notin \mathscr{L}_f^F$ следует, что для любой грамматики Γ, порождающей язык L, найдется бесконечно много чисел n, удовлетворяющих неравенству $F_\Gamma(n) > f(n)$ *). Действительно, если, например, существует лишь конечное число таких n_i, $i = 1, \ldots, p$, для которых $T_\Gamma(n_i) > f(n_i)$, и при этом $f(n) \geqslant 1$, то можно, положив $k = \max\limits_{i=1, \ldots, p} (T_\Gamma(n_i) - f(n_i))$ и воспользовавшись следствием из теоремы об ускорении выводов, построить грамматику Γ_1, эквивалентную Γ и такую, что $T_{\Gamma_1}(n) \leqslant \max(1, T_\Gamma(n) - k) \leqslant f(n)$; аналогично при $F = S$, $f(n) \geqslant n$.

Возникает вопрос: сколь сложным может быть язык в этом смысле? Иначе говоря, можно ли при заданном F указать такую общерекурсивную функцию f, чтобы все рекурсивно перечислимые языки оказались в классе $\mathscr{L}_f^F$?

В такой форме этот вопрос немедленно получает отрицательный ответ; действительно, по лемме 2.1 всякий язык, для которого какая-либо сигнализирующая мажорируется вычислимой функцией, рекурсивен, так что никакой нерекурсивный язык не может принадлежать классу $\mathscr{L}_f^F$ ни для какого сигнализирующего оператора F и ни для какой вычислимой функции f. Однако наиболее важны и в теоретическом, и в прикладном аспекте как раз рекурсивные языки, и естественно поставить вопрос для них. Оказывается, что и в этом случае он решается отрицательно: имеет место

Теорема 2.4. *Для любого сигнализирующего оператора $F(\Gamma, n)$ и любой общерекурсивной функции $f(n)$ можно построить грамматику Γ_0, порождающую рекурсивный язык и такую, что $L(\Gamma_0) \notin \mathscr{L}_f^F$.*

*) Аналогичный факт имеет место и для $F = \mathscr{U}^{\text{Л}}, \mathscr{U}^{\text{П}}, D, I$ (упражнение 2.9).

Доказательство. Напомним, что рассматриваются лишь такие грамматики, основные и вспомогательные словари которых содержатся в счетных множествах E и E_1 соответственно (что не уменьшает общности). Пусть $E = \{a_1, a_3, a_5, \ldots\}$, $E_1 = \{a_2, a_4, a_6, \ldots\}$. Положим $a_1 = a$, $a_3 = b$ и введем некоторое эффективное кодирование грамматик цепочками в словаре $\{a, b\}$. [Можно, например, считать кодом цепочки $a_{i_1} a_{i_2} \ldots a_{i_t}$ цепочку $ba^{i_1+1}bba^{i_2+1}b \ldots ba^{i_t+1}b$, кодом цепочки Λ цепочку bab, кодом правила $\varphi \to \psi$ цепочку $\varkappa(\varphi) b \varkappa(\psi)$, где $\varkappa(\varphi)$, $\varkappa(\psi)$ — коды φ и ψ соответственно, и кодом грамматики $\langle V, W, I, R\rangle$ цепочку $ba^{l_1}b \ldots ba^{l_s}ba^{l}bbb\varkappa(r_1) \ldots \varkappa(r_u)$, где $\alpha_{l_1}, \ldots, \alpha_{l_s}$ и $r_1, \ldots, r_u$ — некоторые пересчеты множеств $V \cup W$ и R соответственно, $\alpha_l = I$ и $\varkappa(r_i)$ — код r_i.] Грамматику, имеющую код x, обозначим Γ_x. Пусть L_0 — множество всех цепочек x в словаре $\{a, b\}$, не удовлетворяющих конъюнкции следующих условий: а) существует грамматика Γ_x; б) $x \in L(\Gamma_x)$; в) $\tilde{F}(\Gamma_x, x) \leqslant f(|x|)$. Для всякой грамматики Γ, порождающей язык L_0, найдется число n такое, что $F(\Gamma, n) > f(n)$. В самом деле, если $L_0 = L(\Gamma)$, то для некоторой цепочки z будет $\Gamma = \Gamma_z$; при этом $z \in L(\Gamma_z) = L_0$, так как из $z \notin L(\Gamma_z)$ следовало бы по определению языка L_0, что $z \in L_0 = L(\Gamma_z)$. Но тогда $\tilde{F}(\Gamma_z, z) > f(|z|)$, откуда $F(\Gamma_z, |z|) > f(|z|)$. В то же время язык L_0 рекурсивен; действительно, $x \in L_0$ тогда и только тогда, когда не выполняется ни одно из равенств $\tilde{F}(\Gamma_x, x) = 0, \ldots, \tilde{F}(\Gamma_x, x) = f(|x|)$, каждое из которых эффективно проверяемо. Грамматику, порождающую L_0, нетрудно построить, зная алгоритмы, вычисляющие функцию $f(n)$, и график функции $\tilde{F}(\Gamma, x)$ (упражнения 2.11, 2.12).

Итак, теорема 2.4 доказана. Но справедливо и более сильное утверждение:

Теорема 2.5. *Для любого сигнализирующего оператора $F(\Gamma, n)$ и любой общерекурсивной функции $f(n)$ можно построить грамматику, порождающую рекурсивный язык и такую, что для любой эквивалентной ей грамматики Γ при всех достаточно больших n имеет место неравенство $F_\Gamma(n) > f(n)$.*

Доказательство. Закодируем грамматики, как в доказательстве теоремы 2.4. Если n — стандартный номер цепочки x в словаре $\{a, b\}$ (стр. 22), будем писать x_n вместо x и Γ_n вместо Γ_x.

Искомый язык L_0 мы будем строить индуктивно, решая последовательно для каждого $n = 0, 1, \ldots$, следует ли причислить к L_0 цепочку x_n. Одновременно будет определяться некоторая вспомогательная частичная числовая функция $h(n)$ — «функция опровержения», не принимающая никакого значения более чем один раз.

«Нулевой шаг» построения состоит в следующем. Если существует грамматика Γ_0 и $\tilde{F}(\Gamma_0, x_0) \leqslant f(|x_0|)$ (что эффективно проверяемо), то полагаем $x_0 \notin L_0$ и $h(0) = 0$. В противном случае (т. е. если грамматики Γ_0 не существует или неравенство $\tilde{F}(\Gamma_0, x_0) \leqslant f(|x_0|)$ не имеет места) полагаем $x_0 \in L_0$, а $h(0)$ остается не определенным.

Пусть произведены шаги с номерами $0, \ldots, n-1$, в результате которых для каждой из цепочек $x_0, \ldots, x_{n-1}$ решено, принадлежит ли она языку L_0, и для каких-то чисел $j_1, \ldots, j_s$ из ряда $0, \ldots, n-1$ найдены значения функции опровержения, причем все эти значения различны; мы будем говорить в этом случае, что «грамматики $\Gamma_{h(j_1)}, \ldots, \Gamma_{h(j_s)}$ опровергнуты». Тогда n-й шаг состоит в следующем. Проверяем выполнение неравенств: (0) $\tilde{F}(\Gamma_0, x_n) \leqslant f(|x_n|)$; (1) $\tilde{F}(\Gamma_1, x_n) \leqslant f(|x_n|)$; ...; $(n)\tilde{F}(\Gamma_n, x_n) \leqslant f(|x_n|)$ (точнее, тех из них, которые имеют смысл, т. е. для тех $i \leqslant n$, для которых Γ_i существуют). Если $i_1, \ldots, i_k$ — все те числа из ряда $0, \ldots, n$, для которых соответствующие неравенства справедливы, ищем среди них наименьшее — пусть это i_p, — для которого грамматика Γ_{i_p} еще не опровергнута. Если такое число нашлось, полагаем $h(n) = i_p$, $x_n \notin L_0$. Если такого числа нет, а также если ни одно из неравенств $(0), \ldots, (n)$, имеющих смысл, не выполняется (в частности, если ни одно не имеет смысла), то полагаем $x_n \in L_0$, а $h(x)$ оставляем не определенным.

Рекурсивность построенного языка L_0 очевидна, и порождающую его грамматику легко построить по алгоритмам, вычисляющим $f(n)$ и график $\tilde{F}(\Gamma, x)$. Ясно также, что L_0 не может порождаться ни одной из опровергнутых грамматик (поскольку из $\tilde{F}(\Gamma, x) \leqslant f(|x|)$ следует, во всяком случае, что $x \in L(\Gamma)$). Поэтому для завершения доказательства достаточно установить, что всякая грамматика Γ_k, для которой найдутся сколь угодно большие числа n, удовлетворяющие неравенству $\tilde{F}(\Gamma_k, n) \leqslant f(n)$, рано или поздно опровергается. Но имеет место даже более сильное утверждение: грамматика Γ_k будет опровергнута, если существует $k+1$ чисел $m_1, \ldots, m_{k+1}$, не меньших k и таких, что для каждого $i = 1, \ldots, k+1$ имеет место $\tilde{F}(\Gamma_k, |x_{m_i}|) \leqslant f(|x_{m_i}|)$

Действительно, пусть Γ_k обладает указанным свойством и никогда не опровергается. Тогда на каждом из шагов с номерами $m_1, \ldots, m_{k+1}$ должна быть опровергнута некоторая грамматика с номером, меньшим k; иначе говоря, функция опровержения для каждого из чисел $m_1, \ldots, m_{k+1}$ будет определена и примет значение, меньшее k, а это невозможно из-за разнозначности данной функции.

Упражнения

2.1. а) Найти временну́ю сложность, левую и правую глубину для каждой из грамматик примеров 6—13 из § 1.3.

б) Найти емкость грамматики примера 13 из § 1.3.

в) Найти разброс и активную емкость для каждой из грамматик примеров 7, 8, 10, 11, 13 из § 1.3.

2.2. Показать, что для любой грамматики Γ и любой общерекурсивной функции $f(m)$, удовлетворяющей условию $\forall m[m \leqslant f(m)]$, можно построить грамматику Γ', эквивалентную Γ и такую, что $S_{\Gamma'}(n) = f(S_\Gamma(n))$.

2.3. Пусть $\mathscr{T}_1$ — класс всех грамматик, у которых левые части правил не содержат основных символов. Пусть далее $\mathfrak{A}(\mathscr{T}, O, F, f(n, F_{\Gamma_1}(n), \ldots, F_{\Gamma_s}(n)))$, где $\mathscr{T}$ — класс грамматик, O — s-местная операция над языками, F — квазисигнализирующий оператор для грамматик и $f(n, m_1, \ldots, m_s)$ — общерекурсивная функция, означает: «Для любых грамматик $\Gamma_1, \ldots, \Gamma_s \in \mathscr{T}$ можно построить грамматику $\Gamma \in \mathscr{T}$, порождающую язык $O(L(\Gamma_1), \ldots, L(\Gamma_s))$ и такую, что $F_\Gamma(n) \leqslant f(n, F_{\Gamma_1}(n), \ldots, F_{\Gamma_s}(n))$».

Показать, что

а) При $F = T$, S, $\mathscr{УЛ}$, $\mathscr{УП}$, I имеет место

$$\mathfrak{A}(\mathscr{T}_1, \cup, F, \max(F_{\Gamma_1}(n), F_{\Gamma_2}(n)));$$

б) имеет место

$$\mathfrak{A}(\mathscr{T}_1, \times, T, \max_{0\leqslant k\leqslant n}(T_{\Gamma_1}(k)+T_{\Gamma_2}(n-k))),$$

$$\mathfrak{A}(\mathscr{T}_1, \times, S, \max_{0\leqslant k\leqslant n}(S_{\Gamma_1}(k), S_{\Gamma_2}(n-k), n)),$$

$$\mathfrak{A}(\mathscr{T}_1, \times, \mathscr{Y}^{\text{Ъ}}, \max(\mathscr{Y}^{\text{Ъ}}_{\Gamma_1}(n), \mathscr{Y}^{\text{Ъ}}_{\Gamma_2}(n))) \ (\text{Ъ} = \text{Л}, \text{П}),$$

$$\mathfrak{A}(\mathscr{T}_1, \times, D, \max(D_{\Gamma_1}(n), D_{\Gamma_2}(n), n)),$$

$$\mathfrak{A}(\mathscr{T}_1, \times, I, \max(I_{\Gamma_1}(n), I_{\Gamma_2}(n))+1) \ (\times - \text{знак умножения});$$

в) имеет место

$$\mathfrak{A}(\mathscr{T}_1, \cap, T, T_{\Gamma_1}(n)+T_{\Gamma_2}(n)+n^2),$$

$$\mathfrak{A}(\mathscr{T}_1, \cap, I, \max(I_{\Gamma_1}(n), I_{\Gamma_2}(n))+n),$$

и — при $F = S, \mathscr{Y}^{\text{Л}}, \mathscr{Y}^{\text{П}}, D$ —

$$\mathfrak{A}(\mathscr{T}_1, \cap, F, \max(F_{\Gamma_1}(n), F_{\Gamma_2}(n)));$$

г) при $O = +, *$ имеет место

$$\mathfrak{A}(\mathscr{T}_1, O, T, \max_{n_1+\ldots+n_k=n}(T_\Gamma(n_1)+\ldots+T_\Gamma(n_k)))\ (n_i\geqslant 1),$$

$$\mathfrak{A}(\mathscr{T}_1, O, S, \max_{0\leqslant k\leqslant n}(S_\Gamma(n-k)+k)),\ \mathfrak{A}(\mathscr{T}_1, O, \mathscr{Y}^{\text{Ъ}}, \mathscr{Y}^{\text{Ъ}}_\Gamma(n))$$

$$(\text{Ъ} = \text{Л}, \text{П}),$$

$$\mathfrak{A}(\mathscr{T}_1, O, D, \max(D_\Gamma(n), n)+1),\ \mathfrak{A}(\mathscr{T}_1, O, I, I_\Gamma(n)+1).$$

2.4. Вывести соотношения, аналогичные приведенным в упражнении 2.3:

а) для подстановки;

б) для левого и правого деления на фиксированную цепочку.

2.5. Показать, что для $F = T, S$ все соотношения упражнения 2.3 сохраняют силу при замене класса $\mathscr{T}_1$ классом Г.

2.6. Пусть $\mathscr{T}_1$ означает то же, что в упражнении 2.3, и пусть **1** — функция $f(n) = 1$ и C — класс всех функций-констант. Показать, что:

а) При $F = \mathscr{Y}^{\text{Л}}, \mathscr{Y}^{\text{П}}, D$

$$\mathscr{L}^F_1(\mathscr{T}_1) = \mathscr{L}^F_1(\mathscr{T}_1 \cap \text{НС}) = \mathscr{L}^F_1(\text{Б}) = \mathscr{L}^F_C(\mathscr{T}_1) = \mathscr{L}^F_C(\mathscr{T}_1 \cap \text{НС}) = \mathscr{L}^F_C(\text{Б})\ ^*);$$

б) $\mathscr{L}^I_1(\mathscr{T}_1) = \mathscr{L}^I_1(\mathscr{T}_1 \cap \text{НС}) = \mathscr{L}^I_1(\text{Б}) = \mathscr{L}^D_1(\mathscr{T}_1)$;

в) $\mathscr{L}^I_C(\mathscr{T}_1) = \mathscr{L}^I_C(\mathscr{T}_1 \cap \text{НС}) = \mathscr{L}^I_C(\text{Б})$.

*) При $F = \mathscr{Y}^{\text{Л}}, \mathscr{Y}^{\text{П}}$ это класс A-языков, при $F = D$ — класс линейных языков (см. ниже, § 5.3 и следствия из теорем 7.2 и 7.3).

Заметим, что для любого класса функций $\mathfrak{F}$ имеет место

$$\mathscr{L}^S_{\mathfrak{F}}(\mathscr{T}_1) = \mathscr{L}^S_{\mathfrak{F}}(\Gamma)\ \ \mathscr{L}^S_{\mathfrak{F}}(\mathscr{T}_1 \cap \text{НС}) = \mathscr{L}^S_{\mathfrak{F}}(\text{НС}),$$

а если $\mathfrak{F}$ замкнут относительно умножения на константу, то

$$\mathscr{L}^T_{\mathfrak{F}}(\mathscr{T}_1) = \mathscr{L}^T_{\mathfrak{F}}(\Gamma),\ \mathscr{L}^T_{\mathfrak{F}}(\mathscr{T}_1 \cap \text{НС}) = \mathscr{L}^T_{\mathfrak{F}}(\text{НС})$$

(см. замечание на стр. 61).

2.7. Пусть $\mathscr{T}_2$ — класс всех грамматик, у которых левая часть каждого правила содержит хоть одно вхождение вспомогательного символа, и **1**, C те же, что в упражнении 2.6. Показать, что:

а) при $F = \mathscr{Y}^{\text{Л}}, \mathscr{Y}^{\text{П}}$

$$\mathscr{L}^F_1(\mathscr{T}_2) = \mathscr{L}^F_1(\text{НС}) = \mathscr{L}^F_1(\text{Б}) = \mathscr{L}^F_C(\mathscr{T}_2) = \mathscr{L}^F_C(\text{НС}) = \mathscr{L}^F_C(\text{Б}) = \mathscr{L}^F_1(\mathscr{T}_1);$$

б) $\mathscr{L}^D_1 = \mathscr{L}^D_1(\mathscr{T}_2) = \mathscr{L}^D_C(\mathscr{T}_2) = \mathscr{L}^I_1(\mathscr{T}_2) = \mathscr{L}^D_C = \mathscr{L}^I_C(\mathscr{T}_2) = \mathscr{L}(\Gamma)$.

Заметим, что для любого класса функций $\mathfrak{F}$ имеет место $\mathscr{L}^S_{\mathfrak{F}}(\mathscr{T}_2) = \mathscr{L}^S_{\mathfrak{F}}(\Gamma)$, а если $\mathfrak{F}$ замкнут относительно умножения на константу, то $\mathscr{L}^T_{\mathfrak{F}}(\mathscr{T}_2) = \mathscr{L}^T_{\mathfrak{F}}(\Gamma)$ (ср. замечание к упражнению 2.6).

2.8. Показать, что при $F = \mathscr{Y}^{\text{Л}}, \mathscr{Y}^{\text{П}}, D$ для любой грамматики Г и любого положительного действительного числа c можно построить грамматику Г′, эквивалентную Г и такую, что $F_{\Gamma'}(n) \leqslant \max(n, 1, cF(n))$. Если при этом Г — НС-, соответственно Б-грамматика, то и Г′ можно сделать НС (Б)-грамматикой.

2.9. Пусть $F(\Gamma, n)$ — квазисигнализирующий оператор для грамматик, и участвующая в определении F функция $f(\Gamma, D, i)$ обладает следующим свойством: существует функция g, отображающая $(E \cup E_1)^*$ в натуральный ряд, обращающаяся в нуль на E^* и такая, что для любой грамматики Г и любого вывода $D = (\omega_0, \omega_1, \ldots, \omega_n)$ в Г имеет место $f(\Gamma, D, i) = g(\omega_i)$. Показать, что в этом случае для любой общерекурсивной функции $h(n)$ и любой грамматики $\Gamma = \langle V, W, I, R\rangle$, для которой $h(n) \geqslant g(I)$ и $L(\Gamma) \notin \mathscr{L}^F_h$, существует бесконечно много чисел n, удовлетворяющих неравенству $F_\Gamma(n) > h(n)$.

2.10. а) Показать, что если для класса грамматик $\mathscr{T}$ существует общерекурсивная функция $f(n)$ такая, что в любой грамматике $\Gamma \in \mathscr{T}$ длины полных выводов цепочек из $L(\Gamma)$ длины, не превосходящей n, не превосходят $f(n)$, то для любого квазисигнализирующего оператора E соответствующий относительный оператор $F^{\mathscr{T}}$ является сигнализирующим и для любой грамматики $\Gamma \in \mathscr{T}$ функция $F_\Gamma(n)$ рекурсивна.

б) Показать, что если для класса грамматик $\mathscr{T}$ существует общерекурсивная функция $f(n)$ такая, что в любой грамматике $\Gamma \in \mathscr{T}$ длины цепочек, встречающихся в полных выводах цепочек из $L(\Gamma)$

длины, не большей n, не превосходят $f(n)$, то для любого квазисигнализирующего оператора F, удовлетворяющего условию упражнения 2.9, соответствующий относительный оператор $F^{\mathscr{T}}$ является сигнализирующим и для любой грамматики $\Gamma \in \mathscr{T}$ функция $F_\Gamma(n)$ рекурсивна.

2.11. Построить грамматику, порождающую множество всевозможных цепочек из $\{a, b\}^*$, являющихся кодами грамматик при способе кодирования, примененном в доказательстве теоремы 2.4.

2.12. Указать способ построения грамматик, порождающих языки из теорем 2.4 и 2.5, по грамматике Γ_1', порождающей язык $L_1 = \{n \# f(n) \mid n = 0,1, \ldots\}$ в словаре $\{|, \#\}$, грамматике Γ_2', порождающей язык $L_2 = \{\varkappa(\Gamma) \# x \# p \mid \widetilde{F}(\Gamma, x) = p\}$ в словаре $\{a, b, |, \#\}$, где $\varkappa(\Gamma)$ и $\varkappa(x)$ — коды грамматики Γ и цепочки x (см. доказательство теоремы 2.4) и грамматике Γ_3', порождающей язык $L_3 = \{a, b, |, \#\}^* - L(\Gamma_2')$.

2.13. а) Показать, что при $F = S$ грамматика Γ_0 из теоремы 2.4 может быть построена так, чтобы имело место неравенство

$$S_{\Gamma_0}(n) = \max\left(S_1(g(n)),\ k^{S_2(g(n))},\ k^{S_3(g(n))}\right),$$

где $g(n) = 2^n + f(2^n) + 1$, k — константа и $S_i(n) = S_{\Gamma_i'(n)}$ (Γ_1', Γ_2', Γ_3' — грамматики из упражнения 2.12).

б) Получить аналогичную оценку для $T_{\Gamma_0}(n)$.

2.14. Найти временные и емкостные сигнализирующие машин, построенных при выполнении упражнений 1.16 и 1.17.

2.15. Сформулировать определения временной и емкостной сигнализирующих для Н-машин и ОН-машин (упражнение 1.20). Найти соотношения между сигнализирующими Э-машин, Н-машин и ОН-машин.

2.16. Показать, что если функция $T_\Gamma(n)$ примитивно рекурсивна, то и язык $L(\Gamma)$ примитивно рекурсивен.

ГЛАВА 3

ГРАММАТИКИ СОСТАВЛЯЮЩИХ

§ 3.1. Деревья выводов

Пусть $\Gamma = \langle V, W, I, R\rangle$ — НС-грамматика, $D = (\omega_0, \ldots, \omega_n)$ — вывод в Γ такой, что $|\omega_0| = 1$, и $D' = (\omega_0', \ldots, \omega_{n-1}', \omega_n)$, $\omega_i' = \xi_i * \varphi_i * \eta_i$, — некоторый соответствующий D размеченный вывод.

Пусть $\omega_i = \beta_{i1} \ldots \beta_{ik_i}$ ($\beta_{ij} \in V \cup W$). Если $\varphi_i \to \psi_i$ — правило, применяемое на i-м шаге D при разметке, отвечающей D', то φ_i и ψ_i можно представить (вообще говоря, не единственным образом — см. упражнение 3.2) в виде $\varphi_i = \chi_i A_i \zeta_i$, $\psi_i = \chi_i \theta_i \zeta_i$, где $A_i \in W$, χ_i, $\zeta_i \in (V \cup W)^*$, $\theta_i \in (V \cup W)^+$.

Фиксировав для каждого $i = 1, \ldots, n-1$ такое представление и обозначая точку $\beta_{i1} \ldots \beta_{i,j-1} * \beta_{ij} * \beta_{i,j+1} \ldots \beta_{ik_i}$ цепочки ω_i через B_{ij}, будем говорить, что точка $B_{i+1,j'}$ цепочки ω_{i+1} является непосредственным потомком точки B_{ij} цепочки ω_i, и писать $B_{ij} \to B_{i+1,j'}$, если либо а) $j = j' \leqslant |\xi_i|$, либо б) $j > |\xi_i| + 1$, $j' = j + |\theta_i| - 1$, либо, наконец, в) $j = |\xi_i| + 1, |\xi_i| + 1 \leqslant j' \leqslant |\xi_i| + |\theta_i|$ (т. е. либо $B_{i+1,j'}$ является «копией» B_{ij}, либо B_{ij} — точка, заменяемая на i-м шаге, а $B_{i+1,j'}$ принадлежит заменяющему ее отрезку).

Обозначим теперь через M' множество всех точек всех цепочек $\omega_0, \ldots, \omega_n$. Ясно, что граф $\langle M'; \to\rangle$ является деревом. Для $\alpha, \beta \in M'$ будем писать $\alpha < \beta$, если либо а) α и β являются точками одной и той же цепочки и $\alpha < \beta$, либо б) существуют $\alpha', \beta' \in M'$, удовлетворяющие условию а) и такие, что из них идут пути в α и в β соответственно. Отношение $<$ есть, очевидно, частичный порядок. Биграф $\langle M'; \to, <\rangle$ мы будем называть растянутым деревом вывода D.

При графическом изображении растянутого дерева вывода удобно помещать вхождения, принадлежащие одной цепочке, на горизонтальной прямой, и отношение $\prec$ понимать как «левее» (ср. § 1.1, стр. 20). Кроме того, договоримся для каждого i располагать цепочку ω_{i+1} (т. е. соответствующую прямую) ниже цепочки ω_i. При таком способе изображения можно, не опасаясь недоразумений, заменить обозначения B_{ij} символами β_{ij}.

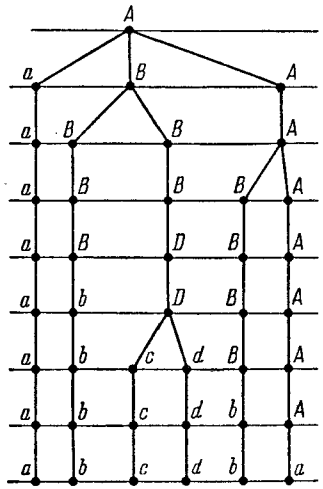


Рис. 2.

Если в $\langle M; \rightarrow \rangle$ из α в β идет путь ненулевой длины, мы будем называть β потомком α и α — предком β.

Пример. Пусть схема НС-грамматики Г содержит правила $A \rightarrow aBA$, $BA \rightarrow BBA$, $BBB \rightarrow BDB$, $B \rightarrow b$, $bDB \rightarrow bcdB$, $bA \rightarrow ba$. Тогда размеченному выводу ($*A*$, $a*BA*$, $aB*BA*$, $a*BBB*A$, $a*B*DBA$, $a*bDB*A$, $abcd*B*A$, $abcd*bA*$, $abcdba$) будет отвечать (в данном случае единственное) растянутое дерево вывода, изображенное на рис. 2.

Теперь мы «стянем в точку» каждый путь в растянутом дереве $T' = \langle M'; \rightarrow, \prec \rangle$ вывода D, не имеющий боковых ответвлений и такой, что все его узлы представляют собой вхождения одного и того же символа *). (Иначе говоря, устраняются все узлы-«копии» и остаются лишь те, которые возникают при применении правил.) Каждый узел A возникающего таким образом дерева представляет собой множество элементов M', являющихся вхождениями одного и того же символа; этот символ мы будем обозначать $g(A)$. Понятно, что если два «старых» узла B и B' оказались «стянутыми» в один «новый» узел, то для любого «старого» узла C

*) Формально эту операцию можно описать как переход к фактор-биграфу по некоторой — строящейся очевидным образом — конгруэнции.

тогда и только тогда $B \prec C$, соответственно $C \prec B$, когда $B' \prec C$ ($C \prec B'$). Поэтому на множестве M «новых» узлов естественным образом определяется отношение частичного порядка $\prec$. Полученный так объект естественно представлять себе как нагруженный биграф $T = \langle M; \rightarrow, \prec, V \cup W, g \rangle$; мы будем называть его деревом вывода D.

Выражения «непосредственный потомок», «потомок» и «предок» будут использоваться в отношении узлов дерева вывода так же, как это делалось для растянутого дерева.

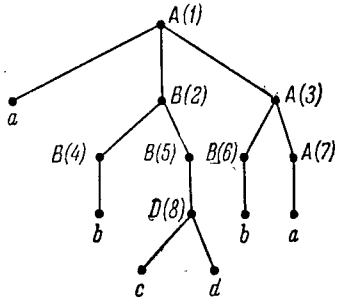


Рис. 3.

При графическом изображении дерева вывода удобно для каждой пары узлов A, B помещать A выше B, если $A \rightarrow B$, и левее B, если $A \prec B$.

Полезно заметить, что длина вывода равна числу невисячих узлов его дерева.

Пример. На рис. 3 изображено дерево вывода, соответствующее растянутому дереву рис. 2. (Цифры в скобках нужны для дальнейшего.)

Заметим еще, что отношение $\prec$ индуцирует на множестве висячих узлов дерева вывода линейный порядок, относительно которого это множество изоморфно последней цепочке соответствующего вывода *).

В дальнейшем нам придется рассматривать деревья, сходные с деревьями вывода, в общем виде. Поэтому будет полезно следующее определение.

Назовем расположенным деревом с помеченными узлами (сокращенно P_1-деревом) нагруженный биграф $\langle M; \rightarrow, \prec, U, g \rangle$ такой, что: а) $\langle M; \rightarrow \rangle$ — дерево; б) для каждого невисячего узла A

*) Это утверждение имеет следующий точный смысл: если $T = \langle M; \rightarrow, \prec, g \rangle$ — дерево вывода $(\omega_0, \ldots, \omega_n)$, $\omega_n = \beta_1 \ldots \beta_k$ ($\beta_1, \ldots, \beta_k$ — элементарные символы) и $M^в$ — множество висячих узлов T, то существует взаимно однозначное отображение τ множества $\{1, \ldots, k\}$ на $M^в$, переводящее естественный порядок в $\prec$ и такое, что $g(\tau(i)) = \beta_i$.

дерева $\langle M; \rightarrow\rangle$ отношение $\prec$ индуцирует на кусте A линейный порядок; в) $B \prec C$ тогда и только тогда, когда найдутся B_0 и C_0, подчиненные одному узлу и такие, что из них ведут пути (возможно, нулевой длины) в B и C соответственно и при этом $B_0 \prec C_0$.

Ясно, что $\prec$ есть частичный порядок. Любое дерево вывода является, очевидно, P_1-деревом.

В следующей главе нам понадобится также понятие расположенного дерева с помеченными узлами и дугами (P_2-дерева). Так мы будем называть упорядоченную систему $\langle M; \rightarrow, \prec, U, Z, g, h\rangle$, где $\langle M; \rightarrow, \prec, U, g\rangle$ — P_1-дерево, Z — конечное множество и h — отображение множества дуг дерева $\langle M; \rightarrow\rangle$ в Z.

Для каждого P_1-и P_2-дерева T отношение $\prec$ индуцирует на множестве его висячих вершин линейный порядок. Если $A_1, \ldots, A_k$ — последовательность, составленная из всех висячих узлов T в соответствии с этим порядком, то цепочка $g(A_1) \ldots g(A_k)$ будет обозначаться $u(T)$.

Говоря о пути в P_1- или P_2-дереве, мы всегда будем подразумевать $\rightarrow$-путь.

Пусть $T = \langle M; \rightarrow, \prec, g\rangle$ — дерево вывода D в НС-грамматике $\Gamma = \langle V, W, I, R\rangle$, оканчивающегося цепочкой ω. «Стянув в точки» все пути в T, не имеющие боковых ответвлений, мы получим новое P_1-дерево $T_1 = \langle M_1; \rightarrow, \prec, U_1, g_1\rangle$, где: U_1 — множество всех подмножеств $V \cup W$; каждый элемент M_1 есть множество элементов M, «стянутых в одну точку»; для каждого $A_1 \in M_1$ множество $g_1(A_1)$ имеет вид $\{g(A) \mid A \in A_1\}$. Это P_1-дерево, обладающее, очевидно, тем свойством, что всякий его невисячий узел имеет не менее двух подчиненных, мы назовем сжатым деревом вывода D. Если теперь для произвольного узла $A_1 \in M_1$ обозначить через $c(A_1)$ множество всех висячих узлов T_1, являющихся потомками A_1, и положить $C = \{c(A_1) \mid A_1 \in M_1\}$, то легко видеть, что C есть система составляющих цепочки ω (§ ПI. 1) и соответствующее дерево составляющих изоморфно дереву $\langle M_1; \rightarrow\rangle$. Более того, если сопоставить каждой составляющей $c(A_1)$ в качестве меток все элементы множества $g_1(A_1)$, то получится размеченная система составляющих цепочки ω, изоморфная «нагруженному дереву» $\langle M_1; \rightarrow, g_1\rangle$.

Пример. На рис. 4 изображено сжатое дерево вывода, отвечающее дереву рис. 3. Оно дает для цепочки $abcdba$ систему составляющих $a(b(cd))(ba)$.

Таким образом, НС-грамматика позволяет естественным образом сопоставить каждой цепочке порождаемого ею языка некоторую размеченную систему составляющих (вообще говоря, не одну — пример см. в упражнении 3.4), которая оказывается тесно связанной с «историей» цепочки, т. е. ее выводом. Это обстоятельство (объясняющее и сам термин «грамматики составляющих») делает НС-грамматики особенно важными для лингвистических приложений — они позволяют не только порождать «грамматически правильные» цепочки, но и автоматически сопоставлять каждой такой цепочке описание ее синтаксической структуры. Неединственность описания дает возможность учитывать явление синтаксической омонимии (см. § ПI.1). НС-грамматика, в которой каждая принадлежащая порождаемому ею языку цепочка имеет единственное дерево вывода, называется однозначной.

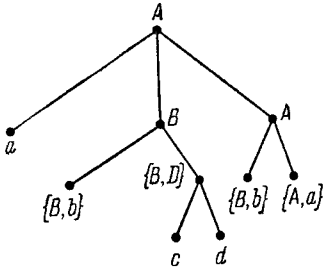


Рис. 4.

Для НС-грамматик можно определить некоторые специфические характеристики сложности выводов, весьма сходные с рассмотренными в гл. 2 сигнализирующими операторами. Именно, пусть $f(\Gamma, C, A)$ — вычислимая функция, принимающая в качестве значений натуральные числа и определенная на всевозможных тройках (Γ, C, A), где $\Gamma = \langle V, W, I, R\rangle$—НС-грамматика (причем мы считаем, что основные и вспомогательные словари всех рассматриваемых грамматик содержатся в некоторых множествах F и F_1 соответственно), C — система составляющих, отвечающая некоторому выводу в Γ, и $A \in C$. Для произвольной цепочки $x \in L(\Gamma)$ и произвольной системы составляющих C, соответствующей какому-нибудь выводу x из I в Γ, положим $\tilde{f}(\Gamma, C, x) = \max_{A \in C} f(\Gamma, C, A)$; далее определим $\tilde{F}(\Gamma, x)$ и $F(\Gamma, n)$

точно так же, как в § 2.1. Функцию $F(\Gamma, n)$ мы будем называть НС-сигнализирующим оператором для НС-грамматик, а функцию $F_\Gamma(n)$, получаемую из $F(\Gamma, n)$ фиксированием Γ, — НС-сигнализирующей (функцией) типа F грамматики Γ.

Термин «НС-сигнализирующий оператор» (а не «НС-квазисигнализирующий») оправдывается тем, что график соответствующей функции $\bar{F}(\Gamma, x)$ всегда рекурсивен; более того, она сама рекурсивна, так что рекурсивна и функция $F(\Gamma, n)$. Это следует из результата упражнения 3.3 и того очевидного факта, что длина бесповторного вывода цепочки x в НС-грамматике не может превосходить $\frac{k^{|x|+1}-1}{k-1}$, где k — мощность полного словаря грамматики.

Отметим три важных для лингвистических приложений типа НС-сигнализирующих, которые получаются если брать в качестве $f(\Gamma, C, A)$ функции $y^{\text{Л}}(A)$, $y^{\text{П}}(A)$ и $\phi(A)$ (§ П1.1). Эти НС-сигнализирующие мы будем обозначать соответственно через $\Upsilon^{\text{Л}}_\Gamma(n)$, $\Upsilon^{\text{П}}_\Gamma(n)$ и $\Phi_\Gamma(n)$. Очевидно, $\Phi_\Gamma(n) \leqslant \min(\Upsilon^{\text{Л}}_\Gamma(n), \Upsilon^{\text{П}}_\Gamma(n))$. Полезно также обратить внимание на простые соотношения между $\Upsilon^{\text{Л}}$ и $\mathscr{Y}^{\text{Л}}$ и между $\Upsilon^{\text{П}}$ и $\mathscr{Y}^{\text{П}}$ (упражнение 3.5, а)).

Можно определить также $\Phi^{\text{Л}}_\Gamma(n)$ и $\Phi^{\text{П}}_\Gamma(n)$ (ср. стр. 289).

§ 3.2. Неукорачивающие грамматики и машины Тьюринга без растяжения

Пусть $\Gamma = \langle V, W, I, R\rangle$ — произвольная грамматика. Назовем правило $\varphi \to \psi \in R$ неукорачивающим, если $|\varphi| \leqslant |\psi|$. Если все правила грамматики Γ неукорачивающие, то сама она тоже будет называться неукорачивающей.

В частности, всякая НС-грамматика неукорачивающая.

Полезно заметить, что в силу леммы 2.2 для всякой неукорачивающей грамматики Γ может быть построена грамматика Γ', не имеющая правил с пустой левой частью (причем простой просмотр доказательства леммы показывает, что Γ' может быть также сделана неукорачивающей), эквивалентная Γ и даже такая, что каждый

полный вывод в Γ является одновременно полным выводом в Γ' и обратно.

Значение неукорачивающих грамматик определяется прежде всего следующим фактом.

Теорема 3.1. *Для всякой неукорачивающей грамматики может быть построена эквивалентная ей* НС-*грамматика.*

Чтобы установить справедливость этой теоремы, мы докажем более сильное утверждение, представляющее интерес и само по себе.

Будем называть грамматику Γ, соответственно Э-машину M, грамматикой (машиной) без растяжения, если $S_\Gamma(n) \leqslant n$, соответственно $S_M(n) \leqslant n$. Очевидно, всякая неукорачивающая грамматика есть грамматика без растяжения (обратное неверно — см. упражнение 3.8).

Теорема 3.2. а) *Для всякой грамматики без растяжения можно построить эквивалентную ей Э-машину без растяжения.*

б) *Для всякой Э-машины без растяжения можно построить эквивалентную ей* НС-*грамматику.*

Из этой теоремы непосредственно следует теорема 3.1.

Доказательство теоремы 3.2. а) Простой просмотр доказательства пункта б) теоремы 1.4 показывает, что длина рабочей ленты фигурирующей там машины M при имитации вывода в грамматике Γ никогда не превышает максимальной длины встречающейся в выводе цепочки. Но если грамматика Γ без растяжения, то в выводе цепочки длины n ни одна промежуточная цепочка не может быть длиннее n.

б) Пусть M — Э-машина без растяжения с входным алфавитом V. Построим для нее машину M_2, как в доказательстве пункта а) теоремы 1.4. В нашем случае M_2 может быть простроена так, чтобы в любом ее вычислении, начинающемся с ситуации $(q', \Lambda, \#\ddagger, \Lambda, \# x)$, ни на одном шаге длина рабочей ленты не была больше $|x|$. Для этого можно воспользоваться «двухэтажной» рабочей лентой, — иначе говоря, рабочим алфавитом, состоящим из кодов пар вида (a, A), где a — входной символ машины M («символ верхнего этажа») и A — рабочий символ M («символ нижнего этажа»). В «верхнем этаже» машина M_2 будет работать, как M на входной

ленте, а в «нижнем», как M на рабочей; по окончании имитации работы M (рабочая) лента уничтожается.

Далее построим машину M_3, обладающую следующими свойствами: (α) рабочий алфавит M_3 получается из рабочего алфавита M_2 добавлением одного символа 0 («пустого символа»); (β) как и в M_2, в M_3 при $x \in V^*$ из ситуации $(q', \Lambda, \# \ddagger, \Lambda, \# x)$ тогда и только тогда достижима ситуация $(q'', \Lambda, \# \ddagger, \Lambda, \#)$, когда $x \in L(M)$; (γ) в любом вычислении машины M_3 каждому шагу, на котором выполняется инструкция вида (iii), предшествует шаг, реализующий инструкцию вида (ii).

Чтобы построить M_3, заметим, что следующие операции можно производить, выполняя перед каждым шагом вида (iii) шаг вида (ii): а) постановку метки в некоторой ячейке (непомеченная ячейка уничтожается, и на ее месте создается помеченная); б) стирание метки (аналогично); в) сдвиг символа 0 на одну ячейку влево (ячейка, содержащая 0, уничтожается, на ее месте создается ячейка, содержащая символ, стоящий в соседней ячейке слева; затем головка сдвигается — шагом вида (iv) — в соседнюю слева ячейку, эта ячейка уничтожается и взамен создается новая, содержащая 0); г) сдвиг символа 0 на одну ячейку вправо (аналогично).

После сделанного замечания ясно, что M_3 может работать так: (1) Всякий раз, когда M_2 делает шаг вида (ii), уничтожая некоторую ячейку, M_3 сначала уничтожает ту же ячейку, затем создает на ее месте новую, содержащую 0, ставит в соседней слева ячейке метку, сдвигает символ 0 влево до тех пор, пока он не окажется рядом с ячейкой, уже занятой таким же символом или символом $\#$, и, наконец, сдвигает головку вправо до помеченной ячейки и стирает там метку. (2) Когда M_2 делает шаг вида (iii), M_3 помечает обозреваемую ячейку, сдвигает головку влево до первой ячейки, занятой нулем (такая обязательно найдется!), двигает этот нуль вправо, пока он не окажется непосредственно справа от помеченной ячейки, затем (стерев предварительно метку) уничтожает его и создает на его месте нужный символ. (3) Шаги машины M_2, имеющие вид (iv) и (v), воспроизводятся машиной M_3 без изменения. (Что касается шагов вида (i), то можно, очевидно, считать, что M_2 их не делает.) Таким образом, M_3 отличается от M_2 тем, что,

уничтожая какую-либо ячейку, она создает взамен в левом конце ленты другую, занятую «пустым символом», а новую ячейку может создать только за счет одной из «пустых». M_3 способна имитировать M_2, потому что в процессе вычисления рабочая лента M_2 никогда не содержит больше ячеек, чем их было вначале.

Теперь построим по M_3 грамматику Γ_1 со схемой R_1 так, как в доказательстве пункта а) теоремы 1.4 строилась по M_2 грамматика Г. В силу свойства (γ) машины M_3 в любом полном выводе грамматики Γ_1 за каждым применением правила вида $\delta q_\beta \to q_\alpha$ следует применение правила вида $q_\alpha \to \varepsilon q_\gamma$; поэтому мы можем заменить в Γ_1 каждое правило вида $\delta q_\beta \to q_\alpha$ совокупностью всевозможных правил вида $\delta q_\beta \to \varepsilon q_\gamma$, где $q_\alpha \to \varepsilon q_\gamma \in R_1$, и полученная грамматика — обозначим ее Γ_2 — будет эквивалентна Γ_1. Остается заменить в Γ_2 каждое правило вида $\delta q_\beta \to \varepsilon q_\gamma$ правилами $\delta q_\beta \to \delta F_{\delta\varepsilon\beta\gamma}$, $\delta F_{\delta\varepsilon\beta\gamma} \to \varepsilon F_{\delta\varepsilon\beta\gamma}$, $\varepsilon F_{\delta\varepsilon\beta\gamma} \to \varepsilon q_\gamma$, каждое правило вида $q_\beta\delta \to \delta q_\beta$ — правилами $q_\beta\delta \to G_{\delta\beta}\delta$, $G_{\delta\beta}\delta \to G_{\delta\beta}H_{\delta\beta}$, $G_{\delta\beta}H_{\delta\beta} \to \delta H_{\delta\beta}$, $\delta H_{\delta\beta} \to \delta q_\beta$, и каждое правило вида $\delta q_\beta \to q_\beta\delta$ — правилами $\delta q_\beta \to \delta M_{\delta\beta}$, $\delta M_{\delta\beta} \to N_{\delta\beta}M_{\delta\beta}$, $N_{\delta\beta}M_{\delta\beta} \to N_{\delta\beta}\delta$, $N_{\delta\beta}\delta \to q_\beta\delta$, где $F_{\delta\varepsilon\beta\gamma}$, $G_{\delta\beta}$, $H_{\delta\beta}$, $M_{\delta\beta}$, $N_{\delta\beta}$ — новые символы, которые мы отнесем к вспомогательному словарю. Полученная грамматика — обозначим ее Γ_3 — является, очевидно, НС-грамматикой, и легко убедиться, что она эквивалентна Γ_2; действительно, каждое исключенное из схемы Γ_2 правило заменимо в выводе соответствующей системой правил Γ_3, и правила каждой такой системы могут применяться в полном выводе грамматики Γ_3 только подряд, так что вместо них можно применить правило из Γ_2. Итак, $L(\Gamma_1) = L(M)$.

Теорема 3.2, и тем самым теорема 3.1, доказана.

Из теоремы 3.1 вытекает

Следствие. *Класс языков, порождаемых неукорачивающими грамматиками, совпадает с* $\mathscr{L}$(НС).

Теоремы 3.1 и 3.2 легко усилить следующим образом. Назовем грамматику Г, соответственно Э-машину M, грамматикой (Э-машиной) с ограниченным растяжением, если существует такое число c, что $S_\Gamma(n) \leqslant cn (S_M(n) \leqslant cn)$. Тогда имеет место

Теорема 3.2′. а) *Для всякой грамматики с ограниченным растяжением можно построить эквивалентную ей Э-машину без растяжения.*

б) *Для всякой Э-машины с ограниченным растяжением можно построить эквивалентную ей* НС-*грамматику.*

Утверждение а) вытекает из теоремы о сжатии выводов и пункта а) теоремы 3.2; утверждение б) — из теоремы о сжатии вычислений и пункта б) теоремы 3.2.

Из теоремы 3.2′ вытекает в свою очередь

Теорема 3.1′. *Для всякой грамматики с ограниченным растяжением можно построить эквивалентную ей* НС-*грамматику.*

Следствие. Пусть $\{c \cdot n\}$ означает класс всевозможных линейных функций и n — функцию $f(n) = n$. Тогда $\mathscr{L}^S_{\{c \cdot n\}} = \mathscr{L}^S_n = \mathscr{L}(\text{НС})$.

Назовем грамматику почти неукорачивающей, если все ее незаключительные правила неукорачивающие. Справедливо следующее утверждение:

Теорема 3.3. *Для всякой почти неукорачивающей грамматики можно построить эквивалентную ей* НС-*грамматику.*

Доказательство. Пусть $\Gamma = \langle V, W, I, R\rangle$ — почти неукорачивающая грамматика и $x \in L(\Gamma)$. Ввиду леммы 1.1 существует полный вывод $D = (\omega_0, \ldots, \omega_m, \ldots, \omega_p)$ в Γ такой, что $\omega_p = x$ и на всех шагах до m — 1-го включительно применяются неукорачивающие правила, а на всех последующих — заключительные. Поскольку при каждом применении заключительного правила появляется хотя бы одно вхождение символа, сохраняющееся до конца, имеем $p - m \leqslant |x|$; поэтому для любого $i = m, \ m+1, \ldots, p$ будет $|x| \geqslant |\omega_i| - |x| \cdot g$, где $g = \max\limits_{\varphi \to \psi \in R} (|\varphi| - |\psi|)$, так что $|\omega_i| \leqslant (g+1) \cdot |x|$; в то же время при $1 \leqslant i < m$ имеем $|\omega_i| \leqslant |\omega_m|$. Следовательно, $\max\limits_{0 \leqslant i \leqslant n} |\omega_i| \leqslant (g+1)|x|$, откуда $S_\Gamma(n) \leqslant (g+1)n$. Остается применить теорему 3.1′.

Заметим, что грамматики примеров 10 и 11 из § 1.3 неукорачивающие, а грамматика примера 13 почти неукорачивающая. Поэтому соответствующие языки являются НС-языками. Дальнейшие примеры см. в упражнениях 3.6 и 3.19.

В заключение параграфа рассмотрим вопрос о свойствах замкнутости класса $\mathscr{L}$(НС).

Теорема 3.4. *Класс* $\mathscr{L}$(НС) *эффективно замкнут относительно операций объединения, пересечения, умножения, усеченной итерации и подстановки.*

Доказательство. Просмотр доказательства теоремы 1.1 показывает, что если Γ', Γ'', Γ — НС-грамматики, то соответствующие грамматики, порождающие языки $L(\Gamma') \cup L(\Gamma'')$ и $L(\Gamma')L(\Gamma'')$, будут также НС-грамматиками, грамматика, порождающая язык $L(\Gamma') \cap \cap L(\Gamma'')$, будет почти неукорачивающей, а грамматика Γ^+, порождающая $(L(\Gamma))^+$, удовлетворяет условию $S_{\Gamma^+}(n) < n + 2 \leqslant 3n$. Доказательство для подстановки предоставляется читателю (оно также не отличается от доказательства для произвольных грамматик).

Замечания. 1) По поводу операций левого и правого деления см. упражнение 3.9.

2) Вопрос о замкнутости класса $\mathscr{L}$(НС) относительно вычитания и взятия дополнения остается открытым.

§ 3.3. Сложность вывода в неукорачивающих грамматиках и НС-грамматиках

Начнем со следующего очевидного, но весьма важного замечания. Поскольку для любой неукорачивающей грамматики Γ имеет место $S_\Gamma(n) \leqslant n$, из леммы 2.1 следует, что каждый НС-язык рекурсивен. Сверх того, в силу замечания на стр. 60 существует единый (и достаточно простой) алгоритм, позволяющий по любой неукорачивающей грамматике Γ и любой цепочке x в основном словаре этой грамматики распознать, выводима ли x в Γ из начального символа *).

Более того, все НС-языки не только рекурсивны, но и примитивно рекурсивны (упражнение 2.16); при этом во всех имеющихся классификациях примитивно рекурсивных языков — или, что то же самое, предикатов — по сложности вычисления (см., например, [Ritchie 1963])

*) Это верно на самом деле и для любой цепочки в полном словаре Γ.

НС-языки занимают самые нижние ступени иерархии.

Переходя к временно́й сложности, отметим прежде всего, что для любой грамматики без растяжения Γ неравенства (2) и (3) из § 2.1 дают

$$cn \leqslant T_\Gamma(n) \leqslant \frac{k^{n+1}}{k-1},$$

где c и k — постоянные, зависящие от Γ (и эффективно определяемые по Γ). О возможности уточнения оценок временно́й сложности хотя бы для отдельных НС-языков известно немного, а то, что известно, получается довольно громоздкими рассуждениями; этот вопрос будет рассмотрен в следующем параграфе, а сейчас мы займемся сравнением временны́х сложностей НС-грамматик, неукорачивающих грамматик и грамматик без растяжения.

Теорема 3.5. *Для любой грамматики без растяжения* Γ *можно построить неукорачивающую грамматику* Γ′, *эквивалентную* Γ *и такую, что*

$$T_{\Gamma'}(n) \leqslant r \cdot [T_\Gamma(n)]^2,$$

где r *— постоянная, эффективно определяемая по* Γ.

Доказательство. Пусть $\Gamma = \langle V, W, I, R\rangle$ — грамматика без растяжения. Положим $\Gamma' = \langle V, W \cup \{I_0, E\}, I_0, R'\rangle$, где I_0, E — новые символы и R' получается из R следующим образом: а) каждое правило $\varphi \to \psi \in R$, где $|\varphi| \neq |\psi|$, заменяется правилом $\varphi E^{|\psi|-|\varphi|} \to \psi$, если $|\varphi| < |\psi|$, и правилом $\varphi \to \psi E^{|\varphi|-|\psi|}$, если $|\varphi| > |\psi|$; б) добавляются новые правила $I_0 \to I_0 E$, $I_0 \to I$, $E\alpha \to \alpha E$, $\alpha E \to E\alpha$, где $\alpha \in V \cup W$. Пусть $(I = \omega_0, \ldots, \omega_t = x)$ — полный вывод в Γ и $|x| = n$. Непосредственной индукцией по i легко показать, что для любого $i = 0, \ldots, n$ в грамматике Γ′ из цепочки IE^{n-1} выводима цепочка $\omega_i E^{n-|\omega_i|}$; при этом имитация применения одного правила $\varphi \to \psi \in R$ состоит в применении соответствующего правила из R' с предварительной «подгонкой» нужного числа символов E к месту применения правила, если $|\varphi| < |\psi|$, или последующим сдвигом вновь полученных E в конец цепочки, если $|\varphi| > |\psi|$; таким образом, на имитацию одного шага в Γ тратится не более $1 + ng$ шагов в Γ′, где $g = \max\limits_{\varphi \to \psi \in R} ||\varphi| - |\psi||$. В частности, из IE^{n-1} выводима не более чем за $t(1 + ng)$ шагов цепочка x, а поскольку IE^{n-1} выводима из I_0 за n шагов и $n \leqslant pt$ (p — постоянная), цепочка x выводима в Γ′ из I_0 не более чем за $n + t(1 + ng) \leqslant t(p + 1 + ptg) \leqslant t^2(p + 1 + pg)$ шагов. В то же время очевидно, что всякая цепочка в словаре V, выводимая в Γ′ из I_0, выводима в Γ из I. Итак, Γ′ эквивалентна Γ, и $T_{\Gamma'}(n) \leqslant (p + 1 + pg) \cdot [T_\Gamma(n)]^2$.

Замечание. Неизвестно, можно ли в теореме 3.5 заменить квадратичную функцию какой-либо медленнее растущей.

Теорема 3.6. *Для любой неукорачивающей грамматики* Γ *и любого положительного действительного числа* c *можно построить* НС-*грамматику* Γ′, *эквивалентную* Γ *и такую, что*

$$T_{\Gamma'}(n) \leqslant \max(1, cT_\Gamma(n)).$$

Доказательство. Обратившись к конструкциям, примененным для доказательства леммы 2.4 (о сцеплении) и теорем 2.1 (о сжатии выводов) и 2.2 (об ускорении выводов), мы без труда убедимся, что каждая из этих конструкций, будучи применена к неукорачивающей грамматике, приводит снова к неукорачивающей грамматике. Это дает нам возможность поступить следующим образом. Пусть Γ — данная неукорачивающая грамматика и $c > 0$ — данное действительное число. Фиксируем еще два положительных действительных числа c_1 и c_2, пока произвольные. Применив конструкцию теоремы об ускорении выводов, построим неукорачивающую грамматику Γ_1, эквивалентную Γ и такую, что $T_{\Gamma_1}(n) \leqslant \max(1, c_1 T_\Gamma(n))$. К этой грамматике применим конструкцию теоремы о сжатии выводов, причем в качестве постоянной c, фигурирующей в формулировке этой теоремы, возьмем число c_2. Полученная при этом грамматика Γ_2 эквивалентна Γ_1 и обладает следующими двумя свойствами: а) $T_{\Gamma_2}(n) \leqslant T_{\Gamma_1}(n) + \max(c_2 n, 1)$; б) длины левых частей правил f и g грамматики Γ_2 (см. доказательство теоремы о сжатии) не превосходят двух, а правых — трех.

К Γ_2 применим теперь конструкцию леммы о сцеплении со следующими видоизменениями: правила 3-го и

4-го рода будут определяться иначе, а именно: правила 3-го рода будут иметь вид $\overline{\alpha(\omega)} \to \omega$, где ω — произвольная непустая цепочка длины, не большей $2l$, в основном словаре Γ_2 (l имеет тот же смысл, что и в доказательстве теоремы о сжатии), а правила 4-го рода — вид $(\alpha(\omega))^0 a \to \omega a$, где ω — произвольная цепочка длины, не большей $2l$, в основном словаре Γ_2 и a — произвольный основной символ Γ_2; кроме того, правилам h грамматики Γ_2 не будут сопоставляться правила 1-го рода. Полученную так грамматику мы обозначим Γ_3.

Если t — длина вывода в Γ_2, m — длина соответствующего вывода в Γ_3 и m_1, m_2, m_3, m_4 имеют такой же смысл, как в доказательстве леммы о сцеплении, то, рассуждая, как в этом доказательстве, мы получаем $m_2 \leqslant (3+1)m_1 = 4m_1 \leqslant 4t$ (поскольку длины правых частей тех правил Γ_2, которым сопоставляются правила 1-го рода, не превосходят трех, а разности длин левых и правых частей этих правил — единицы; кроме того, $t = m_1 + m_3 + m_4$). Отсюда $m \leqslant 5t$. Таким образом, $T_{\Gamma_3}(n) \leqslant 5T_{\Gamma_2}(n)$. При этом каждое правило 1-го рода грамматики Γ_3 имеет один из следующих шести видов:

(i) $\bar{\beta}_1 \to \bar{\gamma}_1$;
(ii) $\bar{\beta}_1 \to \bar{\gamma}_1 \gamma_2^0$;
(iii) $\bar{\beta}_1 \beta_2^0 \to \bar{\gamma}_1 \gamma_2^0$;
(iv) $\beta_1^0 \bar{\beta}_2 \to \bar{\gamma}_1 \gamma_2^0$;
(v) $\bar{\beta}_1 \beta_2^0 \to \bar{\gamma}_1 \gamma_2^0 \gamma_3^0$;
(vi) $\beta_1^0 \bar{\beta}_2 \to \bar{\gamma}_1 \gamma_2^0 \gamma_3^0$.

Построим теперь по грамматике Γ_3 НС-грамматику Γ_4 следующим образом: а) Каждое правило 1-го рода вида (iii)—(vi) и каждое правило 2-го рода заменим некоторой системой новых правил, которую мы выпишем только для случая правил 1-го рода вида (v) и (vi) (для остальных случаев система строится совершенно аналогично). Для правила вида (v): $\bar{\beta}_1 \beta_2^0 \to F\beta_2^0$; $F\beta_2^0 \to F\gamma_2^0\gamma_3^0$; $F\gamma_2^0\gamma_3^0 \to \bar{\gamma}_1\gamma_2^0\gamma_3^0$. Для правила вида (vi): $\beta_1^0\bar{\beta}_2 \to \beta_1^0 F$; $\beta_1^0 F \to GF$; $GF \to G\gamma_2^0\gamma_3^0$; $G\gamma_2^0\gamma_3^0 \to \bar{\gamma}_1\gamma_2^0\gamma_3^0$. Здесь F, G — новые символы, разные для разных правил, которые мы присоединяем к вспомогательному словарю грамматики.

б) Правила 1-го рода вида (i) и (ii), а также правила 3-го и 4-го рода переносятся в схему Γ_4 без изменения. Грамматика Γ_4 эквивалентна Γ_3, поскольку, с одной стороны, применение каждого правила Γ_3 можно заменить применением соответствующей системы правил Γ_4, с другой — в каждом полном выводе в Γ_4 правила одной системы могут применяться только подряд (ввиду того, что цепочка, входящая в полный вывод в Γ_3, не может содержать более одного вхождения символа вида $\bar{\gamma}$), и поэтому вместо каждой такой системы можно применить соответствующее ей правило грамматики Γ_3. Очевидно также, что $T_{\Gamma_4}(n) \leqslant 4T_{\Gamma_3}(n)$.

Итак, мы имеем $T_{\Gamma_4}(n) \leqslant 20\,[T_{\Gamma_1}(n) + \max(c_2 n, 1)]$. Если положить $c_2 = \frac{1}{d_1+1}$, где d_1 — максимум разностей длин правых и левых частей Γ_1 (d_1 эффективно определяемо по Γ и c_1), то в силу последнего из неравенств (2) § 2.1 будет $c_2 n \leqslant T_{\Gamma_1}(n)$, так что $T_{\Gamma_4}(n) \leqslant 40 T_{\Gamma_1}(n) \leqslant 40 \max(1, c_1 T_\Gamma(n))$. Положив $c_1 = c/40$, получим $T_{\Gamma_4}(n) \leqslant \max(40, cT_\Gamma(n))$. Остается добавить к схеме Γ_4 всевозможные правила вида $I \to x$, где I — начальный символ Γ_4, $x \in L(\Gamma_4)$ и $|x| \leqslant 40d_4 + 1$ (d_4 — максимум разностей длин правых и левых частей правил Γ_4); поскольку среди таких цепочек содержатся все выводимые в Γ_4 не более чем за 40 шагов, имеем для полученной таким образом НС-грамматики Γ': $T_{\Gamma'}(n) \leqslant \max(1, cT_\Gamma(n))$.

Замечание. Из теоремы 3.6 вытекает, что в теореме 3.5 квантор существования по r можно заменить квантором общности.

В заключение заметим, что операторы левой и правой глубины, разброса и активной емкости (точнее, соответствующие относительные операторы) являются для класса неукорачивающих грамматик сигнализирующими и соответствующие сигнализирующие функции для любой неукорачивающей грамматики рекурсивны. (Это частный случай утверждения упражнения 2.10, б).)

§ 3.4. Оценка временной сложности некоторых НС-языков

Как отмечалось в предыдущем параграфе, временная сложность любой НС-грамматики заключена между линейной и показательной функциями. Естественно

возникает вопрос: можно ли усилить эту тривиальную оценку, т. е. найти такую функцию $f(n)$, растущую медленнее показательной, чтобы имело место равенство $\mathscr{L}_f^T(\text{НС}) = \mathscr{L}(\text{НС})$ *)? К сожалению, ответить на этот вопрос мы не можем. Мало того, неизвестно даже, верно ли соотношение $\mathscr{L}_{n^2}^T(\text{НС}) = \mathscr{L}(\text{НС})$. Единственный результат, имеющийся пока что в указанном направлении, состоит в том, что ни для какой функции $h(n)$, растущей медленнее, чем n^2, равенство $\mathscr{L}_h^T(\text{НС}) = \mathscr{L}(\text{НС})$ не имеет места. Доказательство этого и будет основной целью настоящего параграфа.

Начнем с введения некоторых вспомогательных понятий. Пусть Г — НС-грамматика, $D = (\omega_0, \ldots, \omega_n)$ — вывод в Г такой, что $|\omega_0| = 1$, T′ — какое-либо растянутое дерево вывода D и M' — множество узлов T′. Распространим отношение «быть предком» («быть потомком»), определенное на M', на множество подмножеств M', являющихся интервалами цепочек $\omega_0, \ldots, \omega_n$. Именно, если $0 \leqslant i < j \leqslant n$ и Q — интервал цепочки ω_j, мы будем называть предком Q в цепочке ω_i наибольший непустой интервал этой цепочки, для которого все потомки его точек, являющиеся точками ω_j, принадлежат Q; если же такого интервала нет, то предком Q в цепочке ω_i мы будем считать любой пустой интервал ω_i, включая несобственные интервалы $(-, \gamma)$ и $(\delta, -)$, где γ и δ — соответственно первая и последняя точки ω_i. Если Q имеет непустого предка в цепочке ω_i, то других предков Q в этой цепочке не существует. Интервал цепочки ω_j, для которого P является предком в ω_i, будет называться потомком P в цепочке ω_j.

Если $0 \leqslant i < j \leqslant n$, P — интервал цепочки ω_i и Q — множество всех точек ω_j, являющихся потомками точек P (очевидно, Q также есть интервал), мы будем называть P точным предком Q в ω_i, а Q — точным потомком P в ω_j.

Пусть далее $D = (\omega_0, \ldots, \omega_n)$ — вывод в НС-грамматике Г, $D' = (\omega_0', \ldots, \omega_{n-1}', \omega_n)$ — некоторый соответствующий D размеченный вывод и T′ — некоторое соответствующее D' растянутое дерево вывода. Если T′ фиксировано, то в каждой цепочке ω_i, $0 \leqslant i \leqslant n-1$, имеется единственная «активная точка», отвечающая «ядру» применяемого на $i+1$-м шаге правила, т. е. такая точка, из которой в дереве T′ либо выходит более одной дуги, либо выходит одна дуга, но конец ее помечен иным символом, чем начало. Пусть эта «активная точка» есть $\chi_i * \varepsilon_i * \theta_i$ (напомним, что точка — это вхождение символа). Если на i-м шаге применяется правило $\varphi_{i-1} \to \psi_{i-1}$, так что $\omega_i = \xi_{i-1}\psi_{i-1}\eta_{i-1}$ (причем выделенное вхождение ψ_{i-1} «то самое»), и если при этом $|\xi_{i-1}| \leqslant |\chi_i| < |\xi_{i-1}\psi_{i-1}|$ (т. е. «активная точка» содержится в ψ_{i-1}), мы будем говорить, что $i+1$-й шаг сильно зацеплен с i-м по отношению к размеченному выводу D' и дереву T′.

Если в НС-грамматике каждому полному выводу D отвечает единственное растянутое дерево T′ и при этом каждый следующий шаг D сильно зацеплен с предыдущим по отношению к T′ и любому соответствующему размеченному выводу D', мы будем называть эту грамматику сильно сцепленной.

Лемма 3.1 (о сильном сцеплении). *Для любой* НС-*грамматики* Г *можно построить сильно сцепленную* НС-*грамматику* Г′, *эквивалентную* Г *и такую, что* $T_{\Gamma'}(n) \leqslant T_{\Gamma}(n)$.

Чтобы убедиться в справедливости этой леммы, достаточно заметить, что построенная при доказательстве теоремы 3.6 грамматика Г′ станет сильно сцепленной, если к правилам 3-го рода фигурирующей в ее построении грамматики Γ_3 добавить всевозможные правила вида $(\alpha(\omega_1))^0\bar{\alpha}(\omega_2) \to (\alpha(\omega_1))^0\beta(\omega_2)$, $(\alpha(\omega_1))^0\beta(\omega_2) \to \gamma(\omega_1)\beta(\omega_2)$, $\gamma(\omega_1)\beta(\omega_2) \to \gamma(\omega_1)\omega_2$, где ω_1, ω_2 — непустые цепочки длины, не большей $2l$, в основном словаре Γ_2 и $\beta(\omega_2)$, $\gamma(\omega_1)$ — новые вспомогательные символы, а правила 4-го рода $(\alpha(\omega))^0 a \to \omega a$ заменить правилами вида $(\alpha(\omega_1))^0\gamma(\omega_2) \to (\alpha(\omega_1))^0\delta(\omega_2)$, $(\alpha(\omega_1))^0\delta(\omega_2) \to \varepsilon(\omega_1)\delta(\omega_2)$, $\varepsilon(\omega_1)\delta(\omega_2) \to \varepsilon(\omega_1)\omega_2$, $\varepsilon(\omega_1)\omega_2 \to \gamma(\omega_1)\omega_2$, $\varepsilon(\omega_1) \to \omega_1$, где ω_1, ω_2, γ означают то же, что и раньше, а $\delta(\omega_2)$, $\varepsilon(\omega_1)$ — новые вспомогательные символы. При этом временна́я сложность грамматики, вообще говоря, возрастает, но не более чем вчетверо, так что подбором постоянной c_2 можно добиться выполнения нужного неравенства для временно́й сложности.

*) Имеет смысл ставить вопрос лишь об усилении верхней оценки — нижнюю, очевидно, усилить нельзя.

Введем еще следующее понятие. Для произвольной цепочки ω будем называть ее сечением любой (собственный) пустой интервал (α, β), где α и β — соседние точки ω. В качестве обозначения для сечения (α, β) будет использоваться также любая пара вида (A, B), где A — отрезок ω с правым концом α и B — отрезок ω с левым концом β.

Пусть теперь Γ — сцепленная грамматика и $D = (\omega_0, \ldots, \omega_n)$ — вывод в Γ, являющийся отрезком полного вывода. Пусть $r_1, \ldots, r_m$ — некоторый пересчет правил Γ и g — максимум длин правых частей правил Γ. Пусть далее $\omega_0 = \omega_0'\omega_0''$ и ω_i', ω_i'' — соответственно точные потомки ω_0' и ω_0'' в ω_i. Обозначим через $\bar{\omega}_i'$ и $\bar{\omega}_i''$ соответственно наибольший конец ω_i' и наибольшее начало ω_i'', длины которых не превосходят g. Отрезок $\bar{\omega}_i'\bar{\omega}_i''$ назовем i-й зоной влияния сечения $(\omega_0', \omega_0'') = \Delta$.

Если $B = \chi * \beta * \zeta$ — точка цепочки $\bar{\omega}_i'$ и $C = \eta * \gamma * \theta$ — точка цепочки $\bar{\omega}_i''$, то расстоянием точки B, соответственно C, от центра зоны будет называться число $-|\beta\zeta|$, соответственно $|\eta\gamma|$ (таким образом, расстояние никогда не равно нулю).

Пусть $i_1, \ldots, i_p$ — последовательность, составленная из номеров тех шагов вывода D, на которых «активные точки» содержатся в соответствующих зонах влияния сечения Δ. Следом вывода D на сечении Δ мы назовем последовательность $((k_1, u_1), \ldots, (k_p, u_p))$, где k_j — расстояние «активной точки» цепочки ω_{i_j-1} от центра зоны и u_j — номер правила, применяемого на шаге с номером i_j. Возможен и пустой след.

Лемма 3.2 (о замещении). *Пусть* $D_1 = (\omega_0, \ldots, \omega_n)$ *и* $D_2 = (\theta_0, \ldots, \theta_m)$ — *выводы в сильно сцепленной* НС-*грамматике* Γ, *являющиеся отрезками полных выводов, и пусть* $\omega_0 = \omega_0'\omega_0''$, $\theta_0 = \theta_0'\theta_0''$, $\omega_n = \omega_n'\omega_n''$, $\theta_m = \theta_m'\theta_m''$, *где* ω_0' *и* θ_0' — *точные предки для* ω_n' *и* θ_m' *соответственно. Если при этом след вывода* D_1 *на сечении* (ω_0', ω_0'') *совпадает со следом вывода* D_2 *на сечении* (θ_0', θ_0''), *то из цепочки* $\omega_0'\theta_0''$ *выводима в* Γ *цепочка* $\omega_n'\theta_m''$.

Доказательство. Пусть $((k_1, u_1), \ldots, (k_p, u_p))$ — общий след и $(i_1, \ldots, i_p)$, $(j_1, \ldots, j_p)$ — соответствующие последовательности шагов выводов D_1 и D_2. Пусть для определенности $k_1 < 0$. Тогда на шагах вывода D_1, предшествующих i_1 (если такие имеются), может преобразовываться только цепочка ω_0'. Если при некотором $t = 1, \ldots, p-1$ числа k_t и k_{t+1} имеют разные знаки, то обязательно $i_{t+1} = i_t + 1$. Если же k_t и k_{t+1}, например, оба положительны, то между шагами i_t и i_{t+1} может преобразовываться только правая половина цепочки (т. е. потомок ω_0''). То же верно и для вывода D_2. Поэтому $\omega_n'\theta_m''$ можно вывести из $\omega_0'\theta_0''$ так: сначала преобразовывать ω_0', как в выводе D_1, до шага i_1; затем проделать шаги, входящие в след, до первого шага — с номером i_t в D_1 и j_t в D_2, — для которого числа k_t и k_{t+1} имеют одинаковые знаки; затем, если, для определенности, эти числа положительны, преобразовывать правую половину цепочки θ_{j_t}, как в выводе D_2, до шага j_{t+1} и т. д.

Теперь мы можем перейти к основному утверждению этого параграфа. Пусть V — словарь, a_1, a_2, b — элементарные символы такие, что $a_1, a_2 \in V$, $b \notin V$, и пусть ψ — функция, отображающая $\{a_1, a_2\}^*$ в bV^*b и удовлетворяющая условию $|\psi(x)| \leqslant d \cdot |x|$, где d — постоянная. Положим $L_\psi = \{x\psi(x)x \mid x \in \{a_1, a_2\}^+\}$. Функцию ψ можно подобрать так, чтобы L_ψ был НС-языком (в частности, это верно при $\psi(x) = bb$ — пример 11 из § 1.3 с тривиальной модификацией; дальнейшие примеры см. в упражнении 3.19). В то же время имеет место

Теорема 3.7. *Если* $f(n)$ — *числовая функция, по порядку меньшая* *) n^2, *то* $L_\psi \notin \mathscr{L}_f^T$(НС).

Доказательство. Допустим противное: пусть существует НС-грамматика $\Gamma = \langle V, W, I, R\rangle$, для которой $L(\Gamma) = L_\psi$ и $T_\Gamma(n) \leqslant f(n)$, так что $\lim\limits_{n\to\infty} T_\Gamma(n)/n^2 = 0$. Будем считать — ввиду леммы 3.1 мы имеем на это право, — что Γ сильно сцеплена.

Введем целочисленные параметры l, n. Число l всегда будет кратно n^2; будем полагать $l/n = r$, $l/n^2 = r/n = q$.

*) Говорят, что функция $f(n)$ по порядку меньше $g(n)$, если $\lim\limits_{n\to\infty} f(n)/g(n) = 0$.

Будем, кроме того, считать, что всегда $q \geqslant g$, где g — максимум длин правых частей правил Г.

Пусть $x \in \{a_1, a_2\}^*$, $|x| = 2l$. Положим $y = y(x) = x\psi(x)x$. Представим $y(x)$ в виде $y(x) = y_1(x) \ldots y_{2n+1}(x)$ (вместо $y_i(x)$ часто будем писать просто y_i), где $|y_1| = \ldots = |y_n| = |y_{n+2}| = \ldots = |y_{2n+1}| = r$, так что, если x' и x'' — соответственно левая и правая половины x, то $y_{n+1} = x''\psi(x)x'$.

Пусть $D_x = (I = \omega^0(x), \omega^1(x), \ldots, \omega^s(x) = y(x))$ — вывод y из I в Г, удовлетворяющий условию $s \leqslant T_\Gamma(|y|)$, откуда для любого $\varepsilon > 0$ при достаточно больших l следует

$$s < \varepsilon l^2. \tag{1}$$

Для каждого $j = 0, \ldots, s$ представим цепочку $\omega^j = \omega^j(x)$ в виде

$$\omega^j = \omega_1^j(x)\alpha_1^j(x)\omega_2^j(x)\alpha_2^j(x) \ldots \alpha_{2n}^j(x)\omega_{2n+1}^j(x),$$

где $\omega_i^j(x)$ — предок $y_i(x)$ и длина каждого $\alpha_i^j(x)$ не больше единицы.

Пусть j_0 — наименьшее из чисел j, для которых $|\omega_{n+1}^j(x)| \geqslant q$.

Будем различать два случая:

1) Найдутся сколь угодно большие n такие, что бесконечно много чисел l, кратных n^2, удовлетворяет условию: не менее половины всех цепочек x длины $2l$ таковы, что как среди отрезков $\omega_1^{j_0}(x), \ldots, \omega_n^{j_0}(x)$, так и среди отрезков $\omega_{n+2}^{j_0}(x), \ldots, \omega_{2n+1}^{j_0}(x)$ имеются отрезки длины, не меньшей q. (Соответствующие n и l будем называть допустимыми.)

2) Для каждого достаточно большого n все достаточно большие числа l, кратные n^2, таковы, что по крайней мере для половины всех цепочек x длины $2l$ либо длины всех отрезков $\omega_1^{j_0}(x), \ldots, \omega_n^{j_0}(x)$, либо длины всех отрезков $\omega_{n+2}^{j_0}(x), \ldots, \omega_{2n+1}^{j_0}(x)$ меньше q.

Случай 1. Пусть i_1, i_2 — соответственно наибольшее из чисел $i = 1, \ldots, n$ и наименьшее из чисел $i = n+2, \ldots, 2n+1$, удовлетворяющих условию $|\omega_i^{j_0}(x)| \geqslant q$ для данной цепочки x.

Пусть $\mathfrak{M}_0$ — множество всех цепочек в $\{a_1, a_2\}$ длины $2l$ и $\mathfrak{M}_1$ — наибольшее по мощности подмножество $\mathfrak{M}_0$, для любых двух цепочек которого i_1 и i_2 соответственно совпадают. При достаточно больших допустимых l будет, очевидно, $\mu(\mathfrak{M}_1) > 2^{11l/6}$.

Положим

$$\pi(x) = \omega_{i_1}^{j_0}(x)\alpha_{i_1}^{j_0}(x)\omega_{i_1+1}^{j_0}(x)\alpha_{i_1+1}^{j_0}(x) \ldots \omega_{i_2}^{j_0}(x).$$

Ясно, что $|\pi(x)| < |\omega_{i_1}^{j_0}| + |\omega_{n+1}^{j_0}| + |\omega_{i_2}^{j_0}| + 2nq + 2n$; но $|\omega_{i_1}^{j_0}|, |\omega_{i_2}^{j_0}| \leqslant r$, $|\omega_{n+1}^{j_0}| \leqslant q + g$; поскольку $q \geqslant g$ и $r = nq$, имеем

$$|\pi(x)| < 8r. \tag{2}$$

Пусть $\mathfrak{M}_2$ — наибольшее по мощности подмножество $\mathfrak{M}_1$, удовлетворяющее условию: если $x_1, x_2 \in \mathfrak{M}_2$, то $\pi(x_1) = \pi(x_2)$. В силу (2) при достаточно больших допустимых n и l будет $\mu(\mathfrak{M}_2) \geqslant 2^{10l/6}$. Общее значение $\pi(x)$ для $x \in \mathfrak{M}_2$ обозначим P.

Начиная с этого момента, число n будем считать фиксированным, так что q и r будут расти пропорционально l.

Обозначим через D'_x «хвост» вывода D_x, начинающийся с ω^{j_0}, через Δ — произвольное сечение цепочки ω^{j_0} и через ρ_Δ — длину следа вывода D'_x на Δ. Если S — множество всех сечений ω^{j_0}, то, очевидно,

$$\sum_{\Delta \in S} \rho_\Delta \leqslant 2gs. \tag{3}$$

В силу (1) и (3) для любого положительного действительного числа δ и любого целого положительного числа k при достаточно больших допустимых l в каждом из отрезков $\omega_{i_1}^{j_0}$, $\omega_{n+1}^{j_0}$, $\omega_{i_2}^{j_0}$ доля сечений, не удовлетворяющих неравенству

$$\rho_\Delta < \delta l, \tag{4}$$

будет не больше $1/k$.

Пусть u есть одно из чисел i_1, $n+1$, i_2 и S_u — множество сечений ω^{j_0}, принадлежащих $\omega_u^{j_0}$ (т. е. тех сечений ω^{j_0}, которые являются одновременно сечениями $\omega_u^{j_0}$). Для произвольного $\Delta \in S_u$ обозначим через f_Δ число

цепочек, принадлежащих $\mathfrak{M}_2$, для которых при данном Δ выполняется (4); число сечений, принадлежащих S_u, для которых (4) выполняется при фиксированной цепочке x, будет обозначаться g_x. Очевидно, $\sum_{\Delta \in S_u} f_\Delta = \sum_{x \in \mathfrak{M}_2} g_x$. Но для любой цепочки $x \in \mathfrak{M}_2$ имеем $g_x \geqslant \mu(S_u) \cdot \left(1 - \frac{1}{k}\right)$. Поэтому в S_u найдется хотя бы одно сечение Δ, для которого

$$f_\Delta \geqslant \mu(\mathfrak{M}_2) \cdot \left(1 - \frac{1}{k}\right). \tag{5}$$

Это сечение будем обозначать Δ_1 при $u = i_1$, Δ_2 при $u = n+1$ и Δ_3 при $u = n_2$.

Пусть $\mathfrak{M}_3$ — подмножество $\mathfrak{M}_2$, состоящее из всех цепочек, для которых каждое из сечений Δ_1, Δ_2, Δ_3 удовлетворяет условию (4). Из (5) следует, что при достаточно больших допустимых l будет $\mu(\mathfrak{M}_3) \geqslant 2^{9l/6}$.

Пусть теперь $\mathfrak{M}_4$ — наибольшее по мощности подмножество $\mathfrak{M}_3$, для любых двух цепочек которого x_1, x_2 следы вывода D'_{x_1} на сечениях Δ_1, Δ_2, Δ_3 соответственно совпадают со следами вывода D'_{x_2} на этих сечениях. В силу (4) при достаточно больших допустимых l будет $\mu(\mathfrak{M}_4) \geqslant 2^{8l/6}$.

Для произвольной цепочки $x \in \mathfrak{M}_4$ представим $\omega^{l_0} = \omega^{l_0}(x)$ в виде

$$\omega^{l_0} = \zeta_1(x)\,\zeta_2(x)\,\zeta_3(x)\,\zeta_4(x),$$

где $(\zeta_i(x), \zeta_{i+1}(x)) = \Delta_i$ $(i = 1, 2, 3)$. Поскольку $\mathfrak{M}_4 \subseteq \mathfrak{M}_2$ и все три сечения Δ_1, Δ_2, Δ_3 принадлежат $\pi(x)$, отрезки $\zeta_2(x)$ и $\zeta_3(x)$ для всех $x \in \mathfrak{M}_4$ соответственно совпадают; будем обозначать их ζ_2^0 и ζ_3^0.

Точный потомок отрезка $\zeta_i(x)$ $(i = 1, 2, 3, 4)$ в цепочке $y(x)$ будет обозначаться $z_i(x)$.

Пусть $\mathfrak{M}_5$ — наибольшее по мощности подмножество $\mathfrak{M}_4$, для любых двух цепочек которого x_1, x_2 при каждом $i = 1, 2, 3, 4$ длины отрезков $z_i(x_1)$ и $z_i(x_2)$ совпадают. При достаточно больших допустимых l будет $\mu(\mathfrak{M}_5) \geqslant 2^{7l/6}$. Общее значение длины $z_i(x)$ для $x \in \mathfrak{M}_5$

обозначим l_i. Поскольку $\Delta_1 \in S_{i_1}$, $\Delta_2 \in S_{n+1}$, $\Delta_3 \in S_{i_2}$, имеем

$$l_1,\ l_4 < l, \tag{6}$$

$$l_1 + l_2,\quad l_3 + l_4 > l. \tag{7}$$

Кроме того, имеет место хотя бы одно из двух неравенств:

$$l_1 + l_2 > 2l, \tag{8}$$

$$l_3 + l_4 > 2l. \tag{8'}$$

Для определенности будем считать, что выполняется (8).

Пусть теперь $\mathfrak{M}_6$ — наибольшее по мощности подмножество $\mathfrak{M}_5$, для любых двух цепочек которого x_1, x_2 отрезки $z_1(x_1)$ и $z_1(x_2)$ совпадают. Очевидно, $\mu(\mathfrak{M}_6) \geqslant 2^{7l/6 - l_1}$. Общее значение $z_1(x)$ для $x \in \mathfrak{M}_6$ обозначим z_1^0.

Покажем, что в $\mathfrak{M}_6$ найдутся две цепочки x_1 и x_2 такие, что

$$z_3(x_1)\,z_4(x_1) \neq z_3(x_2)\,z_4(x_2). \tag{9}$$

В самом деле, если x_1 и x_2 — различные цепочки из $\mathfrak{M}_6$, то равенство

$$z_3(x_1)\,z_4(x_1) = z_3(x_2)\,z_4(x_2) = \tilde{z}$$

возможно только тогда, когда $l_3 + l_4 < 2l - l_1$, так что

$$x_1 = z_1^0 w' \tilde{z},\quad x_2 = z_1^0 w'' \tilde{z},$$

и при этом $w' \neq w''$. Но различных цепочек длины $2l - l_1 - l_3 - l_4$ имеется $2^{2l - l_1 - l_3 - l_4} \leqslant 2^{l - l_1}$ (в силу (7)). Поэтому в $\mathfrak{M}_6$ найдется не менее $2^{l/6}$ цепочек x, не удовлетворяющих условию $z_3(x)\,z_4(x) = \tilde{z}$.

Пусть x_1, x_2 — цепочки из $\mathfrak{M}_6$, удовлетворяющие (9). Из цепочек

$$\omega^{l_0}(x_1) = \zeta_1(x_1)\,\zeta_2^0\zeta_3^0\zeta_4(x_1)$$

и

$$\omega^{l_0}(x_2) = \zeta_1(x_2)\,\zeta_2^0\zeta_3^0\zeta_4(x_2)$$

в Γ выводимы соответственно цепочки

$$y(x_1) = z_1^0 z_2(x_1)\,z_3(x_1)\,z_4(x_1)$$

и

$$y(x_2) = z_1^0 z_2(x_2)\,z_3(x_2)\,z_4(x_2).$$

По лемме 3.2 из $\omega^{l_0}(x_1)$, а значит и из l, должна быть выводима также цепочка $\tilde{y} = z_1^0 z_2(x_2) z_3(x_1) z_4(x_1)$. Но в силу (8) и (9) цепочки $y(x_2)$ и $\tilde{y}$ не могут одновременно принадлежать L_ψ.

Случай 2. Пусть для определенности найдутся сколь угодно большие n такие, что бесконечно много чисел l, кратных n^2, удовлетворяют условию: по крайней мере для четверти всех цепочек x длины $2l$ длины всех отрезков $\omega_1^{l_0}, \ldots, \omega_n^{l_0}$ меньше q. Рассуждая совершенно аналогично случаю 1 (с очевидными упрощениями), можно для достаточно большого l найти две цепочки x_1 и x_2 такие, что $\omega^{l_0}(x_1)$ и $\omega^{l_0}(x_2)$ представимы в виде $\omega^{l_0}(x_1) = \zeta_1^0 \zeta_2(x_1)$, $\omega^{l_0}(x_2) = \zeta_1^0 \zeta_2(x_2)$, где

а) следы «хвостов» выводов D_{x_i}, начинающихся с $\omega^{l_0}(x_i)$, на сечениях $(\zeta_1^0, \zeta_2(x_i))$ при $i = 1$ и при $i = 2$ одинаковы;

б) если $z_1(x_i)$, $z_2(x_i)$ — соответственно точные потомки отрезков ζ_1^0, $\zeta_2(x_i)$ в цепочках $y(x_i)$ $(i = 1, 2)$, то $|z_1(x_1)| = |z_1(x_2)| = l_1$, $|z_2(x_1)| = |z_2(x_2)| = l_2$, причем l_1, $l_2 > l$ и хотя бы одно из чисел l_1, l_2 больше $2l$;

в) $z_i(x_1) \neq z_i(x_2)$ $(i = 1, 2)$.

По лемме 3.2 из $\omega^{l_0}(x_1)$, а значит и из l, должна быть выводима цепочка $z_1(x_2) z_2(x_1)$, которая в силу в) не может принадлежать L_ψ одновременно с $y(x_2)$, если $l_1 > 2l$, и одновременно с $y(x_1)$, если $l_2 > 2l$.

Теорема доказана.

Замечания. 1) Среди языков, удовлетворяющих условию теоремы 3.7, имеются такие, для которых доставляемая этой теоремой нижняя оценка временной сложности является точной, т. е. в принципе не может быть улучшена. Так, легко убедиться, что для грамматики примера 11 из § 1.3 временная сложность меньше n^2, а по теореме 3.6 для порождаемого этой грамматикой языка можно построить и НС-грамматику не большей сложности. (Это верно и при упомянутой на стр. 97 модификации.) Другие примеры указаны в упражнении 3.19. Как уже отмечалось, примеры языков, для которых вытекающая из теоремы 3.7 оценка не является точной, неизвестны.

2) Если функция ψ ни для какой постоянной d не удовлетворяет условию $|\psi(x)| \leqslant d \cdot |x|$, то утверждение теоремы 3.7 может не иметь места (упражнение 3.20).

Остановимся еще на классе $\mathscr{L}_n^T$(НС) (совпадающем, в силу теоремы 3.6, с $\mathscr{L}_{\{c \cdot n\}}^T$(НС), где $\{c \cdot n\}$ — класс всех линейных функций). Ввиду теоремы 3.7 этот класс является собственной частью $\mathscr{L}$(НС); с другой стороны, он содержит все Б-языки (см. ниже, замечание к упражнению 4.1). Естественно спросить, является ли $\mathscr{L}$(Б) собственной частью $\mathscr{L}_n^T$(НС). Положительный ответ на этот вопрос вытекает из следующего примера.

Пример 1. Пусть Г — НС-грамматика со схемой

$$
\begin{gathered}
I \to bA_1b \\
bA_1 \to bA_2A_2 \\
A_2A_1 \to A_2A_2A_2 \\
A_2b \to A_3A_3b \\
A_2A_3 \to A_3A_3A_3 \\
bA_3 \to bA_4A_4 \\
A_4A_3 \to A_4A_4A_4 \\
A_4b \to A_1A_1b \\
A_4A_1 \to A_1A_1A_1 \\
A_1 \to a
\end{gathered}
$$

Очевидный подсчет показывает, что $T_\Gamma(n) \leqslant 2(n-2)$. В то же время $L(\Gamma) = \{ba^{2^{4k}}b \mid k = 0, 1, \ldots\}$; в следующей главе будет доказано утверждение (следствие из теоремы 4.5), из которого непосредственно вытекает, что этот язык не является бесконтекстным.

Рассмотрим еще один пример языка, принадлежащего разности $\mathscr{L}_n^T$(НС) — $\mathscr{L}$(Б).

Пример 2. Язык примера 10 из § 1.3 — $\{a^n b^n a^n \mid n = 1, 2, \ldots\}$ — не является Б-языком (см. ниже, § 4.3, пример 1). Покажем, как построить неукорачивающую грамматику с линейной временной сложностью, порождающую этот язык *).

*) Это построение принадлежит Р. В. Фрейвалду.

Прежде всего легко построить неукорачивающую грамматику Γ_1, порождающую язык $\{\tilde{n}\tilde{n}'\tilde{n} \mid n = 1, 2, \ldots\}$, где $\tilde{n}$ — двоичная запись числа n и $\tilde{n}'$ — тоже двоичная запись n, но состоящая из цифр $0'$ и $1'$ (чтобы можно было различить «зоны» цепочки); эта грамматика строится аналогично грамматике примера 11 из § 1.3, и, как в этом примере, Γ_1 можно построить так, чтобы цепочка $x = \tilde{n}\tilde{n}'\tilde{n}$ была выводима в Γ_1 не более чем за $c|x|^2$ шагов, где c — некоторая постоянная (ср. замечание 1) после теоремы 3.7). Поскольку $|\tilde{n}\tilde{n}'\tilde{n}| \leqslant 3(\log_2 n + 1)$, число $c|\tilde{n}\tilde{n}'\tilde{n}|^2$ мажорируется линейной функцией от $3n = |a^n b^n c^n|$. Поэтому нам достаточно построить неукорачивающую грамматику Γ_2, в которой для любого $n = 1, 2, \ldots$ из цепочки $\tilde{n}$ будет выводима не более чем за $c_1 n$ шагов, где c_1 — постоянная, цепочка a^n, и не будет выводима никакая другая цепочка в словаре $\{a\}$.

Мы не будет выписывать схему грамматики Γ_2 ввиду ее громоздкости; ограничимся описанием принципа ее работы — вполне достаточным, впрочем, для фактического построения схемы *).

Пусть $\tilde{n} = i_s i_{s-1} \ldots i_1 i_0$, где $i_l = 0$, 1 и $i_s = 1$. Вывод в Γ_2, начинающийся цепочкой $\tilde{n}$, будет состоять из $s+1$ «макрошагов». Цепочка, выведенная к началу k-го макрошага ($k = 0, \ldots, s$), будет иметь вид $i_s \ldots i_{k+1} i_k \mathcal{Y} a^{\overline{i_{k-1} \ldots i_0}}$, где $\overline{i_{k-1} \ldots i_0} = i_{k-1} \cdot 2^{k-1} + \ldots + i_0 \cdot 2^0$ и $\mathcal{Y} = A_0 A_0 A A_0 A^3 A_0 A^7 \ldots A_0 A^{2^{k-1}-1}$. Макрошаг состоит в следующем: а) Если $i_k = 0$, то i_k превращается в A_0, каждое «старое» вхождение A_0 — в $A_0 A$ и каждое вхождение A — в AA. В результате получается цепочка $i_s \ldots i_{k+1} A_0 A_0 A A_0 A^3 \ldots A_0 A^{2^k-1} a^{\overline{i_k \ldots i_0}}$ (поскольку в этом случае $\overline{i_{k-1} \ldots i_0} = \overline{i_k \ldots i_0}$). б) Если $i_k = 1$ и $k < s$, то i_k превращается в $A_0 A_0 A$, каждое «старое» вхождение A_0, кроме последнего, — в $A_0 A^3$, каждое вхождение A левее последнего вхождения A_0 — в A^4, а последнее вхождение A_0 и каждое вхождение A правее него — в aa. В результате получается опять-таки цепочка $i_s \ldots i_{k+1} A_0 A_0 A A_0 A^3 \ldots A_0 A^{2^k-1} a^{\overline{i_k \ldots i_0}}$. в) Если $k = s$, то i_s и все вхождения символов A_0 и A заменяются на a. Получается цепочка $a^{\overline{i_s \ldots i_0}}$.

*) На самом деле для осуществления приводимой ниже конструкции первый и последний символы цепочки $\tilde{n}$ должны быть предварительно как-то помечены.

Легко видеть, что схему описанной грамматики можно построить так, чтобы на каждом шаге, не входящем в последний макрошаг, цепочка удлинялась. Поэтому длина вывода цепочки a^n из $\tilde{n}$ не будет превышать $2n$ шагов.

Построение примера закончено.

Заметим еще, что любой язык из $\mathscr{L}_n^T(\text{НС})$ распознается ДЭ-машиной с ограниченным растяжением (теорема 2.3, б), лемма 2.3, упражнение 3.14), так что его дополнение является НС-языком. В то же время даже дополнение к Б-языку, как и пересечение двух Б-языков, может не принадлежать $\mathscr{L}_n^T(\text{НС})$ (упражнение 3.21).

§ 3.5. НС-грамматики с односторонним контекстом

В настоящем параграфе будет рассмотрен один специальный класс НС-грамматик, выделяемый с помощью некоторого на первый взгляд весьма сильного ограничения, но тем не менее оказывающийся эквивалентным по «порождающей силе» классу всех НС-грамматик.

Будем называть НС-грамматику левоконтекстной, соответственно правоконтекстной, если правые (левые) контексты всех ее правил пусты. Левоконтекстные и правоконтекстные НС-грамматики будем называть также НС-грамматиками с односторонним контекстом.

Довольно долго стоял вопрос о существовании НС-грамматик с односторонним контекстом, порождающих не бесконтекстные языки (см. библиографические замечания). Тем не менее имеет место

Теорема 3.8. *Для любой НС-грамматики может быть построена эквивалентная ей левоконтекстная НС-грамматика.*

Доказав эту теорему, мы, разумеется, будем вправе заключить о справедливости аналогичного факта для правоконтекстных грамматик.

Чтобы сделать основную идею доказательства теоремы 3.8 более ясной, рассмотрим сначала пример.

Пусть Г — грамматика со следующей схемой (которую мы для удобства разделим на 4 группы):

I. 1) $I \to AIB$
2) $I \to AB$
3) $I \to a_i a_i$

II. 1) $A \to C_i$
2) $C_i A \to C_i A_i$
3) $A_i A \to A_i A_i$
4) $A_i B \to A_i D_i$
5) $A_i F_j \to A_i F_{ji}$
6) $F_{ji} F_k \to F_{ji} F_{ki}$
7) $F_{ji} B \to F_{ji} D_i$

III. 1) $C_i \to E_i$
2) $E_i A_i \to E_i A$
3) $A A_i \to A A$
4) $A D_i \to A F_i$
5) $A F_{ji} \to A F_j$
6) $F_j F_{ki} \to F_j F_k$
7) $F_j D_i \to F_j F_i$

IV. 1) $E_i \to a_i$
2) $F_i \to a_i$

(Индексы i, j, k принимают значения 1, ..., n.)

Нетрудно видеть, что $L(\Gamma) = \{xx \mid x \in \{a_1, \ldots, a_n\}^+\}$. Именно, цепочка xx, где $x = a_{i_1} \ldots a_{i_s}$, $s \geqslant 2$, может быть порождена так. Сначала по правилам группы I порождаем $A^s B^s$. Затем заменяем первое вхождение A на C_{i_1} (правило строки II, 1)), с помощью II, 2) и II, 3) передаем информацию в начало правой половины цепочки и заменяем первое вхождение B на D_{i_1} (II, 4)). Это — первая половина первого цикла. Вторая половина начинается применением III, 1), после чего левая половина цепочки очищается от ненужной теперь информации (III, 2), III, 3)) и D_{i_1} заменяется на F_{i_1} (III, 4)). Следующие циклы проводятся так же, только для передачи информации используются дополнительно II, 5) и II, 6), для уничтожения ее — III, 5) и III, 6), а вместо II, 4) и III, 4) применяются соответственно II, 7) и III, 7). После s циклов получаем цепочку $E_{i_1} \ldots E_{i_s} F_{i_1} \ldots F_{i_s}$, которая затем преобразуется в xx (IV, 1), 2)). Любой цикл можно в любой момент оборвать и перейти к следующему; однако если хоть один цикл не будет доведен до конца, то мы не сможем уничтожить все вхождения вспомогательных символов. Поэтому никакая цепочка в основном словаре, не имеющая вида xx, не выводима из I в Г.

Перейдем к доказательству теоремы 3.8. Для произвольной грамматики $\Gamma = \langle V, W, I, R \rangle$ и произвольного языка $L \subseteq (V \cup W)^*$ будем называть Г-образом L множество всех тех цепочек в V, которые выводимы в Г из цепочек, принадлежащих L. Мы докажем предложение, несколько более сильное, чем теорема 3.8, а именно: для произвольной неукорачивающей грамматики Г с основным словарем V можно построить грамматику Г′ с основным словарем V и вспомогательным словарем W такую, что: 1) каждое правило Г′ имеет вид $\alpha A \to \alpha\beta$ или $A \to \beta$, где $A \in W$, $\alpha, \beta \in V \cup W$; 2) $L(\Gamma)$ является Г′-образом КС-языка $\{\#H^k G^n H^m G^n \mid k, m, n = 0, 1, \ldots\}$, где $\#$, G, H — некоторые элементы W.

Заметим прежде всего, что для данной неукорачивающей грамматики Г можно построить такую грамматику Γ_1, что в каждом ее правиле длины левой и правой частей равны и $L(\Gamma)$ есть Γ_1-образ языка $\{I_0 Q^m \mid m = 0, 1, \ldots\}$, где I_0 и Q — некоторые вспомогательные символы Γ_1. Построение Γ_1, весьма простое, предоставляется читателю.

Любой вывод цепочки $x \in V^*$ из цепочки $I_0 Q^m$ в Γ_1 (такой вывод существует тогда и только тогда, когда $x \in L(\Gamma)$ и $m = |x| - 1$) будем называть нормальным выводом x.

Построим теперь грамматику Γ_2, обладающую свойством 1) искомой грамматики Г′, а также следующим свойством: 2′) вспомогательный словарь Г′ содержит такие символы ∇, P, Q, что Γ_2-образ языка $L_0 = \{\nabla P^n Q^m P^n \mid m, n = 0, 1, \ldots\}$ состоит из всевозможных цепочек вида $\nabla z x \hat{z}$, где $x \in L(\Gamma)$ и z — двоичная запись длины некоторого нормального вывода x.

Мы не будем приводить формальное построение Γ_2 и ограничимся описанием принципа ее работы. При этом нам удобно будет представлять себе Γ_2 — поскольку она работает с цепочкой $\nabla P^n Q^m P^n$ — как некоторую машину, снабженную конечной лентой, длина которой в процессе работы не изменяется; по ленте слева направо движутся головки, которые могут возникать и исчезать. При этом можно потребовать, чтобы головки могли возникать только в крайней слева ячейке

(а исчезать — в любой) и чтобы в одной ячейке не могли одновременно находиться две головки. Можно даже потребовать, чтобы на ленте в каждый момент была только одна головка.)

В начале работы нам удобно будет считать все ячейки пустыми. Символы ∇, P и Q будет интерпретироваться как различные метки, стоящие в пустых ячейках; эти метки будут сохраняться и при заполнении ячеек, так что любая ячейка в любой момент несет информацию о том, является ли она крайней слева, и если нет, то какой из трех зон она принадлежит — левой (первые n ячеек, не считая крайней слева), средней (следующие m ячеек) или правой (последние n ячеек).

В левой и правой зонах будут записываться цепочки в словаре {0, 1}, причем цепочка v, записанная в левой зоне, всегда будет двоичной записью натурального числа, а цепочка w, записанная в правой зоне, — обращением двоичной записи натурального числа. Для простоты мы будем отождествлять v, соответственно w, с числом, двоичной записью которого служит $\hat{v}$, соответственно $\hat{w}$; будем говорить, например: «прибавим к v единицу» и т. п.

Работа машины распадается на два этапа; вплоть до конца второго этапа в левой и правой зонах вместо нулей и единиц стоят их двойники, являющиеся вспомогательными символами; мы, однако, будем для простоты пренебрегать этим.

Первый этап состоит в последовательном выполнении циклов, которые будут сейчас описаны.

1-й цикл. В последней ячейке левой зоны записывается 0, в первой ячейке средней зоны записывается I_0 и в первой ячейке правой зоны записывается 0.

$s+1$-й цикл ($s \geqslant 0$). Содержимое левой зоны увеличивается на 1, затем в средней зоне имитируется применение некоторого правила грамматики Γ_1 (причем сначала первый слева символ выбранного вхождения левой части правила заменяется первым символом правой части, затем второй и т. д.) и, наконец, увеличивается на 1 содержимое правой зоны. При этом прибавление единицы в правой зоне производится обычным способом: если в младшем (т. е. самом левом) разряде стоит нуль, то он заменяется единицей, если единица, она заменяется нулем и делается перенос. В левой

зоне единица прибавляется «наугад»: до какого-то места цепочка проходится слева направо без изменения, затем в некоторой ячейке нуль заменяется единицей (вариант — записывается единица в пустой ячейке); если следующая ячейка оказалась пустой, операция закончена, если нет, делается попытка истолковать произведенную замену нуля единицей как перенос при прибавлении единицы; если такое толкование невозможно, т. е. где-то правее есть еще нуль, то машина «ломается».

Ясно, что если $x \in L(\Gamma)$ и z — длина некоторого нормального вывода x (в двоичной записи), то цепочка $\nabla z x \hat{z}$ может быть получена описанным только что способом из подходящей цепочки языка L_0. Покажем, кроме того, что если x — цепочка в основном словаре Γ_1, то цепочка вида $\nabla z x \hat{z}$, где $z \in \{0, 1\}^*$, может быть получена на первом этапе работы машины только тогда, когда $x \in L(\Gamma)$ и z — длина некоторого нормального вывода x. Для этого заметим, что если x не выводима в Γ_1 из $I_0 Q^{|x|-1}$, то цепочка вида $\nabla v x w$, v, $w \in \{0, 1\}^*$, может появиться в результате первого этапа лишь в случае, когда некоторые циклы обрываются при прохождении средней зоны (например: вместо того, чтобы применить, скажем, правило $A_1 B_1 \to A_2 B_2$, заменяем A_1 на A_2 и после этого сразу переходим к следующему циклу). Поэтому нам, во всяком случае, достаточно установить, что при получении цепочки $\nabla z x \hat{z}$ никакой цикл не может оборваться после того, как начало изменяться содержимое левой зоны, но до того как закончилось изменение содержимого правой. Очевидно однако, что: а) при обрыве цикла в средней зоне (в частности, на границе с левой или правой) содержимое левой зоны успевает увеличиться на 1, в то время как содержимое правой не меняется; б) при обрыве внутри левой зоны после начала изменения ее содержимое увеличивается даже больше чем на 1 (например, из $101 = 5$ вместо $110 = 6$ получится $111 = 7$); в) при обрыве внутри правой зоны до окончания изменения ее содержимое уменьшается (поскольку запись в этой зоне производится справа налево; например, из $1011 = 13$ вместо $0111 = 14$ получится $0011 = 12$). Следовательно, если хоть один цикл оборвется в нежелательном месте, то цепочки вида $\nabla z x \hat{z}$ мы уже не получим.

Назначение второго этапа работы машины — проверить, совпадает ли содержимое правой зоны с обращенным содержимым левой. Это можно сделать аналогично тому, как порождалась цепочка xx в примере на стр. 106 (ясно, что этот пример нетрудно видоизменить так, чтобы получались цепочки вида $x\hat{x}$). Формальные построения мы опускаем. Можно представлять себе ячейки правой зоны «двухэтажными»; в верхних этажах записан результат первого этапа, в нижних — копия содержимого левой зоны. В тех ячейках, в которых оба этажа оказываются заняты «тезками», двухэтажные вспомогательные символы заменяются основными; основные символы возникают и в левой зоне. Этим работа машины заканчивается.

В дальнейшем будем считать, что длины цепочек языка $L(\Gamma)$ не меньше восьми. Если это не так, то для цепочек меньшей длины можно очевидным образом ввести специальные правила.

Заметим теперь, что для данной грамматики Γ существует такая эффективно по ней вычисляемая постоянная c, что для каждой цепочки $x \in L(\Gamma)$ среди цепочек, принадлежащих Γ_2-образу L_0, найдется хотя бы одна цепочка $\nabla z x \hat{z}$, для которой $|z| < c|x|$. (Это следует из того, что среди нормальных выводов цепочки x найдется хотя бы один бесповторный, длина которого мажорируется показательной функцией от $|x|$.)

Фиксируем натуральное число $t \geqslant 2(2c+1)$. Сопоставим каждой цепочке η длины, не большей $2t$, состоящей из символов, входящих в словари всех рассмотренных нами грамматик, новый символ $\alpha(\eta)$. Определим понятие образа цепочки аналогично тому, как это было сделано в доказательстве теоремы 2.1, используя число t вместо фигурировавшего там l. С помощью того же метода «блочного кодирования», который был применен в указанном доказательстве, легко преобразовать грамматику Γ_2 в грамматику $\tilde{\Gamma}_2$, обладающую свойством 1) грамматики Γ_2 и сверх того следующим свойством: $\tilde{2}$) $\tilde{\Gamma}_2$-образ языка $\{\# H^k G^n H^m G^n \mid k, m, n = 0, 1, \ldots\}$ состоит из всевозможных цепочек вида $\# R^k \alpha(z)\alpha(x)\alpha(\hat{z})$, где $x \in L(\Gamma)$, $\hat{z}$ — двоичная запись длины некоторого нормального вывода x, R — некоторый специальный символ, $k = 0, 1, \ldots$ и $\alpha(\omega)$ означает образ цепочки ω. В силу выбора числа t для каждой цепочки $x \in L(\Gamma)$ найдется такая цепочка $\chi = \# R^k \alpha(z)\alpha(x)\alpha(\hat{z})$, принадлежащая $\tilde{\Gamma}_2$-образу языка $\{\# H^k G^n H^m G^n \mid k, m, n = 0, 1, \ldots\}$, что $|\chi| = |x|$.

Теперь для завершения конструкции достаточно построить грамматику Γ_3, удовлетворяющую тому же условию 1) и такую, что а) для всякой цепочки $x \in V^*$ и всякой цепочки $\chi = \# R^k \alpha(z)\alpha(x)\alpha(\hat{z})$, где $z \in \{0, 1\}^*$ и $|\chi| = |x|$, из χ в Γ_3 выводима x; б) цепочка $x \in V^*$ выводима в Γ_3 из цепочки вида $\# R^s \alpha(z)\alpha(u)\alpha(\hat{z})$, где $u \in V^*$, $z \in \{0, 1\}^*$, лишь тогда, когда это имеет место согласно пункту а). Принцип работы Γ_3 может быть охарактеризован так. Пользуясь тем же представлением грамматики как машины, которое было использовано при построении Γ_2, будем считать ленту «четырехэтажной». Начальная информация записана в первом этаже. Работа начинается с того, что кусок первого этажа, являющийся образом x, переписывается в правый конец второго этажа. Затем в третьем этаже записывается произвольная цепочка $y \in V^*$; после этого в правом конце четвертого этажа записывается (произвольный) образ y; наконец, содержимое второго и четвертого этажей сравнивается, и в случае их совпадения уничтожаются все этажи, кроме третьего.

Доказательство теоремы закончено.

Упражнения

3.1. Построить дерево вывода и растянутое дерево вывода цепочки $ba^{16}b$ в грамматике примера 1 из § 3.4.

3.2. Показать, что одному размеченному выводу в НС-грамматике Γ может отвечать более одного растянутого дерева вывода тогда и только тогда, когда схема Γ содержит правило вида $\varphi A(\xi\eta)^n \xi B \psi \to \varphi A(\xi\eta)^{n+k} \xi B \psi$, где A, B — вспомогательные символы, $\eta \neq \Lambda$, $n = 0, 1, \ldots$, $k = 1, 2, \ldots$, причем левая часть этого правила входит хотя бы в одну цепочку, выводимую из какого-либо вспомогательного символа.

3.3. Показать, что для всякого сжатого дерева вывода T в НС-грамматике Γ найдется такой бесповторный размеченный вывод D в Γ, что одио из сжатых деревьев, отвечающих D, совпадает с T.

3.4. Подсчитать число деревьев вывода и сжатых деревьев вывода цепочки $a^{3^n} b^{3^n}$ в грамматике со схемой $\{I \to A,\ I \to B,\ A \to aAb,\ A \to ab,\ B \to a^3Bb,\ B \to aBb^3,\ B \to a^3b,\ B \to ab^3\}$.

3.5. а) Показать, что для каждой НС-грамматики Γ имеет место неравенство $\mathrm{Y}^{\text{Л}}_{\Gamma} \leqslant \mathscr{Y}^{\text{Л}}_{\Gamma}(n) + d$, где d — максимум длин цепочек

x в основном словаре Г, для которых существуют правила Г, имеющие вид $\varphi A\psi \to \varphi x\theta\psi$. Аналогично для $\mathcal{Y}_\Gamma^\Pi$ и Υ_Γ^Π.

б) Построить НС-грамматику Г такую, что $\Upsilon_\Gamma^\Pi(n) = 1$, $\mathcal{Y}_\Gamma^\Pi(n) = n$.

3.6. Построить неукорачивающие или почти неукорачивающие грамматики, порождающие следующие языки:

а) $\{xx \mid x \in \{a_1, \ldots, a_k\}^+\}$;

б) $\{a_1^n a_2^n \ldots a_k^n \mid n = 1, 2, \ldots\}$ (k — фиксированное натуральное число);

в) $\{a^n b^m a^n b^m \mid m, n = 1, 2, \ldots\}$;

г) $\{x \mid x \in \{a_1, \ldots, a_k, b_1, \ldots, b_k\}^+;\ |x|_{a_i} = |x|_{b_i}\ (i = 1, \ldots, k)\}$;

д) множество всех простых чисел в простейшей записи;

е) $\{\tilde{x} * \tilde{y} * \tilde{z} \mid xy = z\}$, где $\tilde{x}$, $\tilde{y}$, $\tilde{z}$ — двоичные записи натуральных чисел x, y, z соответственно.

3.7. Коммутативным замыканием языка $L \subseteq \subseteq \{a_1, \ldots, a_k\}^*$ называется язык $\{x \mid x \in \{a_1, \ldots, a_k\}^* \& \exists y\,(y \in \in L \& |x|_{a_i} = |y|_{a_i},\ i = 1, \ldots, k\}$. Показать, что:

а) коммутативное замыкание НС-языка есть НС-язык (причем по любой НС-грамматике Г можно построить НС-грамматику, порождающую коммутативное замыкание $L(\Gamma)$);

б) обратное неверно (более того, даже неперечислимый язык может иметь в качестве коммутативного замыкания язык $\{a_1, a_2\}^*$).

3.8. Показать, что для любой НС-грамматики можно построить эквивалентную ей грамматику без растяжения, не являющуюся почти неукорачивающей.

3.9. Показать, что для любой НС-грамматики Г и для любой цепочки x в основном словаре Г можно построить НС-грамматику Г′ такую, что $L(\Gamma') = (x \setminus L(\Gamma)) - \{\Lambda\}$. Аналогично для правого деления.

3.10. Показать, что существуют вычислимые числовые функции $f(p, q)$, обладающие следующим свойством: для любой неукорачивающей грамматики, мощность полного словаря и длины левых и правых частей правил которой не превосходят соответственно p и q, можно построить эквивалентную ей неукорачивающую грамматику, длины левых и правых частей правил которой не превосходят соответственно двух и трех, а мощность вспомогательного словаря не превосходит $f(p, q)$. Найти явное выражение одной из таких функций. [*Указание.* Проанализировать доказательство теоремы 2.1.]

3.11. Показать, что для любой неукорачивающей грамматики можно построить эквивалентную ей почти неукорачивающую грамматику с вспомогательным словарем, состоящим из трех символов.

3.12. Пусть код грамматики определяется, как в доказательстве теоремы 2.4, и пусть $\mathcal{T}$ — какой-нибудь класс грамматик, основные и вспомогательные словари которых содержатся в некоторых счетных множествах, причем объединение этих множеств не содержит символов, используемых для кодирования. Назовем грамматику Γ_0 универсальной для класса $\mathcal{T}$, если $L(\Gamma_0) = \{\varkappa(\Gamma)\, x \mid \Gamma \in \in \mathcal{T} \& x \in L(\Gamma)\}$, где $\varkappa(\Gamma)$ — код Г. Показать, что существует грамматика Γ_0, универсальная для класса грамматик без растяжения и такая, что $S_{\Gamma_0}(n) \leqslant n \log_2 n$.

3.13. Построить ДЭ-машины с ограниченным растяжением, допускающие:

а) каждый из языков упражнения 3.6;

б) множество всех простых чисел в двоичной записи;

в) множество $\{\alpha_1 \ldots \alpha_s \mid s = 1, 2, \ldots\}$, где α_i — i-й знак дробной части десятичного разложения числа $\sqrt{2}$;

г) то же для числа e.

3.14. Показать, что для всякой ДЭ-машины без растяжения, допускающей язык L, можно построить ДЭ-машину без растяжения, распознающую тот же язык L. [*Указание.* Если ДЭ-машина без растяжения не допускает цепочку x, то x-вычисление ведет либо к безрезультатной остановке, либо к уходу головки за пределы «x-зоны», либо к «зацикливанию» в этой зоне; каждое из этих трех явлений можно обнаружить, не выходя из «x-зоны».]

Замечание. Неизвестно, существуют ли языки, допускаемые Э-машинами без растяжения и не допускаемые ДЭ-машинами без растяжения. Отрицательное решение этой проблемы повлекло бы положительное решение проблемы замкнутости класса НС-языков относительно вычитания и взятия дополнения (ср. замечание 2) к теореме 3.4).

3.15. Дать верхние оценки временны́х сложностей грамматик, построенных при выполнении упражнения 3.6.

3.16. Оценить сверху возрастание временно́й сложности при построениях, использованных для доказательства теоремы 3.2. Получить из этих оценок другое доказательство теоремы 3.5.

3.17. Распространить теорему 3.6 на почти неукорачивающие грамматики.

3.18. Назовем грамматику $\Gamma = \langle V, W, I, R\rangle$ обобщенной НС-грамматикой (ОНС-грамматикой), если каждое ее правило имеет вид $\xi_1 A \xi_2 \longrightarrow \xi_1 \theta \xi_2$, где $A \in W$ и $\xi_1, \xi_2, \theta \in (V \cup W)^*$. Показать, что для произвольной грамматики можно построить эквивалентную ей обобщенную НС-грамматику и притом без увеличения временно́й сложности. [*Указание.* Рассуждать по аналогии с доказательством теоремы 3.6.]

3.19. Для каждой из следующих функций ψ_i построить неукорачивающую грамматику, порождающую язык L_{ψ_i} и такую, что ее временна́я сложность мажорируется квадратичной функцией: $\psi_1(x) = = bxb$; $\psi_2(x) = b\hat{x}b$; $\psi_3(x) = b\hat{x}x\hat{b}$.

3.20. а) Построить неукорачивающую грамматику Г, порождающую язык $\{xbx^{|x|}bx \mid x \in \{a_1a_2\}^+\}$ и такую, что $T_\Gamma(n) \leqslant n\sqrt{n}$.

б) Показать, что для любой монотонной числовой функции $h(m)$, по порядку не меньшей m и такой, что множество $\{h(m) \mid m = 1, 2, \ldots\}$ является НС-языком, можно подобрать функцию ψ, отображающую V в $b\{a_1, a_2\}^* b\,(a_1, a_2 \in V,\ b \notin V)$ и удовлетворяющую условию $|\psi(x)| \leqslant h(|x|)\cdot|x|$, так, чтобы существовала НС-грамматика, порождающая язык L_ψ и имеющая временну́ю сложность, по порядку не превосходящую $n \cdot h^{-1}(n)$.

3.21. а) Пусть $\psi(x) = bb$. Построить Б-грамматику, порождающую язык $C'L_\psi$. [*Указание.* Использовать конструкцию примера 12 из § 1.3.]

б) Очевидно, при $\psi(x) = b\hat{x}b$ имеем $L_\psi = L_1 \cap L_2$, где $L_1 = \{xb\hat{x}by \mid x, y \in \{a_1, a_2\}^+\}$, $L_2 = \{xb\hat{y}by \mid x, y \in \{a_1, a_2\}^+\}$. Построить Б-грамматики, порождающие L_1 и L_2.

3.22. а) Построить грамматику Г с правилами вида $\alpha A \to \alpha\beta$ и $A \to \beta$, где A — вспомогательный символ и α, β — элементарные символы, такую, что Г-образ языка P^+ (P — некоторый символ) есть $\{a^{n+k}b^n \mid n = 1, 2, \ldots;\ k = 0, 1, \ldots\}$.

б) Показать, что для любой НС-грамматики Г можно построить грамматику Γ_1 с правилами такого же вида, как в пункте а), и грамматику Γ_2 с правилами вида $A\alpha \to \beta\alpha$ и $A \to \beta$ (смысл обозначений тот же) такие, что $L(\Gamma)$ будет Γ_2-образом Γ_1-образа языка P^+ [Haines 1969].

3.23. Построить левоконтекстные НС-грамматики, порождающие следующие языки:

а) $\{a^nb^na^n \mid n = 1, 2, \ldots\}$;

б) $\left\{x\hat{x}x \mid x \in \{a_1, \ldots, a_k\}^+\right\}$;

в) $\{a^nba^{2^n}ba^n \mid n = 1, 2, \ldots\}$.

3.24. Проанализировав доказательство теоремы 3.8, показать, что для всякой НС-грамматики Г можно построить левоконтекстную НС-грамматику Г′, эквивалентную Г и такую, что $T_{\Gamma'}(n) \leqslant [T_\Gamma(n)]^2$

ГЛАВА 4

БЕСКОНТЕКСТНЫЕ ГРАММАТИКИ И МАШИНЫ С МАГАЗИННОЙ ПАМЯТЬЮ

§ 4.1. Некоторые вспомогательные утверждения

В этом параграфе мы докажем несколько простых лемм о Б-грамматиках, которые пригодятся нам в дальнейшем.

Лемма 4.1. *Для любой* Б-*грамматики можно построить эквивалентную ей* Б-*грамматику, схема которой не содержит правил вида* $A \to B$, *где* A, B — *вспомогательные символы.*

Доказательство. Пусть $\Gamma = \langle V, W, I, R\rangle$ — Б-грамматика. Введем на W бинарное отношение ρ_Γ: $A\rho_\Gamma B$ тогда и только тогда, когда схема Г содержит правило $A \to B$. Граф $\langle W, \rho_\Gamma\rangle$ обозначим G_Γ. Если $A, B \in W$ и в G_Γ имеются пути из A в B и из B в A (допускаются и пути нулевой длины), то будем писать $A\sigma_\Gamma B$. Ясно, что σ_Γ есть отношение эквивалентности. Сопоставим каждому классу $S \in W/\sigma_\Gamma$ новый символ α_S и положим $\Gamma_1 = \langle V, W_1, \alpha_{S_1}, R_1\rangle$, где $W_1 = \{\alpha_S \mid S \in W/\sigma_\Gamma\}$, S_1 — класс, содержащий I, и R_1 получается из R заменой в левых и правых частях всех правил каждого вхождения каждого символа $A \in W$ вхождением символа $\alpha_{S(A)}$, где $S(A)$ — класс, содержащий A. Легко видеть, что Γ_1 эквивалентна Г и граф G_{Γ_1} не содержит замкнутых путей ненулевой длины.

Пусть теперь B — висячий неизолированный узел графа G_{Γ_1} и $A_1, \ldots, A_k$ — все узлы, из которых идут дуги в B. Устраним из схемы Γ_1 правила $A_1 \to B, \ldots$

..., $A_k \to B$ и одновременно добавим к ней всевозможные правила вида $A_1 \to \psi$, ..., $A_k \to \psi$, где ψ — правая часть какого-либо правила Γ_1 с левой частью B. Полученную так грамматику обозначим Γ_2. Поскольку B — висячий узел G_{Γ_1}, среди новых правил нет ни одного правила вида $A_i \to C$, где $C \in W_1$, так что граф G_{Γ_2} имеет меньше дуг, чем G_{Γ_1}. В то же время Γ_2 эквивалентна Γ_1. Продолжая этот процесс, получим в конце концов грамматику Γ_3, эквивалентную Γ_1 и такую, что все узлы графа G_{Γ_3} изолированные; это и требовалось сделать.

Лемма 4.2. *Для всякой* Б-*грамматики можно построить эквивалентную ей* Б-*грамматику, правые части правил которой не содержат начального символа. Если при этом исходная грамматика удовлетворяет требованию леммы* 4.1, *то новую грамматику можно построить так, чтобы и она ему удовлетворяла. Если исходная грамматика автоматная, то и новую можно сделать автоматной.*

Доказательство. Достаточно добавить к вспомогательному словарю данной грамматики новый символ I', в правых частях правил заменить все вхождения начального символа вхождениями I' и добавить к схеме всевозможные правила вида $I' \to \psi$, где ψ — правая часть правила, левой частью которого служит начальный символ.

Будем называть Б-грамматику Γ стандартной бинарной Б-грамматикой, если каждое правило Γ имеет вид $A \to BC$ или $A \to a$, где A, B, C — вспомогательные символы и a — основной символ.

Стандартная бинарная Б-грамматика обладает тем свойством, что для всякого ее дерева вывода соответствующая система составляющих (см. § 3.1) бинарна.

Лемма 4.3. *Для всякой* Б-*грамматики можно построить эквивалентную ей стандартную бинарную* Б-*грамматику. Если при этом исходная грамматика удовлетворяет требованию леммы* 4.2, *то новую можно построить так, чтобы и она ему удовлетворяла.*

Доказательство. Пусть $\Gamma = \langle V, W, I, R\rangle$ — Б-грамматика. В силу леммы 4.1 мы можем считать, что R не содержит правил вида $A \to B$, где A, $B \in W$. Мы можем считать также, что каждое правило либо имеет

вид $A \to a$, где $A \in W$, $a \to V$, либо его правая часть не содержит основных символов. Если это не так, преобразуем Γ следующим образом: сопоставим каждому $a \in V$ новый символ $\tilde{a}$, присоединим все такие символы к W, во всех правилах Γ, длины правых частей которых больше единицы, заменим все вхождения основных символов вхождениями их «двойников» и присоединим к схеме Γ всевозможные правила вида $\tilde{a} \to a$, где $a \in V$; полученная грамматика, очевидно, удовлетворяет нужному условию и эквивалентна Γ. Но при выполнении перечисленных условий для получения нужного результата достаточно заменить каждое правило вида $A \to B_1B_2 \ldots B_k$, $k > 2$, системой правил $\{A \to B_1F_1, F_1 \to B_2F_2, \ldots, F_{k-3} \to B_{k-2}F_{k-2}, F_{k-2} \to B_{k-1}B_k\}$, где $F_1, \ldots, F_{k-2}$ — новые символы, разные для разных правил Γ.

Займемся теперь деревьями выводов в Б-грамматиках. Заметим прежде всего, что по самому определению растянутого дерева вывода каждому размеченному выводу в Б-грамматике, начинающемуся одноэлементной цепочкой, отвечает единственное растянутое дерево, так что между такими размеченными выводами и их растянутыми деревьями имеется взаимно однозначное соответствие. Что касается собственно дерева вывода, то отвечающих ему размеченных выводов в общем случае много; мы покажем сейчас, как их можно описать.

Пусть $\Gamma = \langle V, W, I, R\rangle$ — Б-грамматика и $T = \langle M; \to, <, V \cup W, g\rangle$ — P_1-дерево, являющееся деревом вывода в Γ некоторой цепочки $\omega \in (V \cup W)^*$ из некоторого символа $B \in W$. Такое P_1-дерево мы будем называть (B, ω)-деревом (в грамматике Γ).

Расположим множество невисячих узлов T в линейную последовательность E_1, ..., E_k так, чтобы выполнялось следующее условие: если E_j зависит от E_i, то $i \leqslant j$. Всякую последовательность невисячих узлов, удовлетворяющую этому условию, будем называть допустимой. Легко видеть, что для любого $i = 1, \ldots, k$ подграф G_i дерева $\langle M; \to\rangle$, образованный узлами $E_1, \ldots, E_i$ и всеми соединяющими их дугами, будет поддеревом, корень которого совпадает с корнем $\langle M; \to\rangle$. Построим теперь последовательность цепочек $\omega_1, \ldots, \omega_k$; она будет строиться индукцией по $k - i$, причем одновременно с цепочкой ω_i будет определяться

(B, ω_i)-дерево $T_i = \langle M_i; \rightarrow_i, <_i, g_i\rangle$ такое, что его поддерево, образованное всеми невисячими узлами, совпадает с G_i и для каждого $j < i$ имеет место $g_i(E_j) = g(E_j)$. При $i < k$ будет определяться также некоторое вхождение ω'_i символа $g(E_{i+1})$ в цепочку ω_i. Индукция проводится следующим образом I. Полагаем $\omega_k = \omega$, $T_k = T$. II. Пусть определены ω_i, ω'_i и T_i $(1 < i \leqslant k)$, удовлетворяющие сформулированным условиям. В силу допустимости последовательности $E_1, \ldots, E_k$ узел E_i будет в дереве G_i висячим; поэтому все непосредственные потомки E_i в дереве T_i являются в нем висячими узлами и в соответствующей этому дереву системе составляющих для ω_i образуют некоторую составляющую (ранга 0 или 1). Цепочку, полученную из ω_i заменой этой составляющей вхождением символа $g(E_i)$, мы обозначим ω_{i-1}, само это вхождение обозначим ω'_{i-1}, а дерево, полученное из T_i выбрасыванием всех (непосредственных) потомков E_i, обозначим T_{i-1}. Выполнение нужных условий для ω_{i-1}, ω'_{i-1} и T_{i-1} очевидно. Из способа построения ω_i непосредственно ясно, что последовательность $D = (B, \omega_1, \ldots, \omega_k = \omega)$ является выводом в Г, а последовательность $D' = (*B*, \omega'_1, \ldots, \omega'_{k-1}, \omega_k)$ — размеченным выводом. Индукцией по i легко показать, что каждое T_i есть дерево вывода, отвечающее размеченному выводу $(*B*, \omega'_1, \ldots, \omega'_{i-1}, \omega_i)$, так что, в частности, T отвечает D'.

Итак, каждой допустимой последовательности невисячих узлов T отвечает размеченный вывод D'. При этом ясно, что различным допустимым последовательностям соответствуют различные размеченные выводы. Картина будет завершена, если мы покажем, что любой размеченный вывод, отвечающий данному дереву T, может быть получен таким способом. Чтобы убедиться в этом, рассмотрим размеченный вывод $D' = (\omega'_0, \omega'_1, \ldots, \omega'_{k-1}, \omega_k)$, где $\omega'_0 = *B*$, и «сожмем» соответствующее растянутое дерево T', как описано в § 3.1. Узлами полученного дерева T будут классы узлов T', причем каждый класс состоит из узлов, лежащих на одном пути, и при этом каждое вхождение ω'_i, $i = 0, \ldots, k-1$, является «младшим» (самым далеким от корня) узлом

в некотором классе, отвечающем невисячему узлу T. Если теперь обозначить через E_i тот узел T, для которого младший узел соответствующего класса есть ω'_{i-1}, то последовательность $E_1, \ldots, E_k$ будет, как легко видеть, допустимой, и размеченный вывод, полученный по этой последовательности описанным выше способом, совпадает с D'.

Полезно заметить, что среди размеченных выводов, отвечающих данному дереву вывода, всегда есть один и только один упорядоченный. С другой стороны, упорядочиваемому выводу в Б-грамматике отвечает единственный упорядоченный размеченный вывод. Таким образом, между деревьями вывода и упорядочиваемыми выводами в Б-грамматике имеется взаимно однозначное соответствие.

Пример. Пусть Б-грамматика Г содержит правила $A \to aBA$, $B \to BB$, $A \to BA$, $B \to D$, $B \to b$, $D \to cd$, $A \to a$. Тогда размеченному выводу $(*A*, a*B*A, aBB*A*, aB*B*BA, a*B*DBA, ab*D*BA, abcd*B*A, abcdb*A*, abcdba)$ будут отвечать дерево вывода, изображенное на рис. 3 (стр. 81), и растянутое дерево вывода на рис. 2; это последнее дерево соответствует следующей допустимой последовательности невисячих узлов первого (обозначенных цифрами, стоящими на рис. 3 в скобках): 1, 2, 3, 5, 4, 8, 6, 7. Другой допустимой последовательности: 1, 3, 7, 6, 2, 5, 4, 8 — отвечают растянутое дерево на рис. 5 и размеченный вывод $(*A*, aB*A*, aBB*A*, aB*B*a, a*B*ba, aB*B*ba, a*B*Dba, ab*D*ba, abcdba)$.

Приведем еще несколько простых соображений, относящихся к деревьям вывода.

Лемма 4.4. *Пусть D — вывод в* Б-*грамматике* Г, *T — его дерево, k — длина D, s — высота T и g — максимум длин правых частей правил* Г. *Тогда* $s \leqslant k \leqslant \frac{g^s - 1}{g - 1}$, *если* $g > 1$, *и* $s = k$, *если* $g = 1$.

Доказательство. Первое неравенство очевидно; второе следует из того, что k равно числу невисячих узлов T и из каждого узла T исходит не более g дуг. Равенство $s = k$ при $g = 1$ очевидно.

Будем называть (B, ω)-дерево простым, если в нем никакие два различных невисячих узла, лежащих

на одном пути, не помечены одним и тем же символом, (B, ω)-дерево, всякое собственное полное поддерево которого простое, будем называть квазипростым.

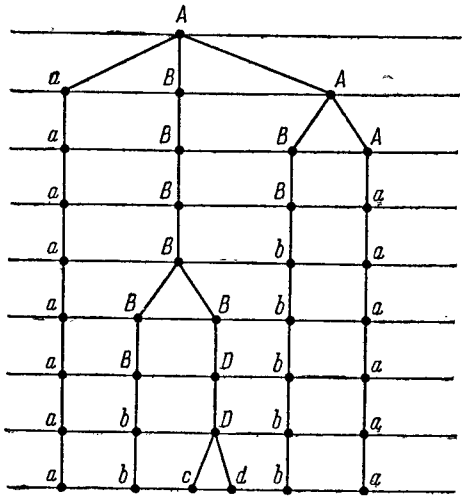


Рис. 5.

(Полное (δ-)поддерево P_1-дерева T — это P_1-поддерево T, образованное всеми узлами, в которые идут пути из данного узла δ.)

Очевидна следующая

Лемма 4.5. *Высота простого (квазипростого) (B, ω)-дерева в* Б-*грамматике* Γ *не превосходит* p *(соответственно* $p+1$*), где* p *— мощность вспомогательного словаря* Γ.

В заключение введем операцию над деревьями вывода, которая будет нам весьма полезна при изучении Б-грамматик. Пусть $T = \langle M; \rightarrow, \prec, g\rangle$ есть (A, ω)-дерево, β — какой-либо узел T, E — составляющая цепочки ω, соответствующая узлу β (в системе составляющих, определяемой деревом T), и $\xi * \varphi * \eta$ — соответствующее этой составляющей вхождение подцепочки в ω. Пусть, кроме того, $g(\beta) = B$. Обозначим через T_1 полное P_1-поддерево дерева T с корнем в β; очевидно, T_1 есть (B, φ)-дерево. Рассмотрим теперь произвольную цепочку ψ, выводимую из B, и произвольное (B, ψ)-дерево

T_2, и обозначим через $\mathrm{Sub}(T, T_1, T_2)$ P_1-дерево, полученное из T «вырезанием» поддерева T_1 и подстановкой на его место T_2. Ясно, что $\mathrm{Sub}(T, T_1, T_2)$ является $(A, \xi\psi\eta)$-деревом, и в соответствующей системе составляющих для цепочки $\xi\psi\eta$ узел β дает составляющую, отвечающую вхождению $\xi * \psi * \eta$ подцепочки ψ.

§ 4.2. Распознавание пустоты и конечности Б-языка. Проекции

Пусть $\Gamma = \langle V, W, I, R\rangle$ — Б-грамматика, $A \in W$, $\alpha \in V \cup W$. Будем говорить, что α зависит от A, если из A выводима в Γ хотя бы одна цепочка, содержащая вхождение α.

Лемма 4.6. *Существует алгоритм, позволяющий по любой* Б-*грамматике* $\Gamma = \langle V, W, I, R\rangle$ *и любым двум символам* $A \in W$, $\alpha \in V \cup W$ *распознать, зависит ли* α *от* A.

Доказательство. Пусть в Б-грамматике $\Gamma = \langle V, W, I, R\rangle$ существуют выводы вида $(A = \omega_0, \ldots, \omega_k = \xi\alpha\eta)$. Возьмем тот из этих выводов, чье дерево содержит меньше всего узлов (если таких деревьев несколько, берем любое из них), и обозначим это дерево через T_0. Среди висячих узлов T_0 хотя бы один помечен символом α. Пусть $\gamma_0, \gamma_1, \ldots, \gamma_s$ — путь из корня T_0 в этот узел. Все узлы T_0, не лежащие на этом пути, являются висячими. Действительно, в противном случае для некоторого i, $i = 0, \ldots, s-1$, какой-либо узел δ, подчиненный γ_i и отличный от γ_{i+1}, будет невисячим; но тогда дерево $\mathrm{Sub}(T_0, T', T'')$, где T' — полное δ-поддерево T_0 и T'' — дерево, состоящее из одного узла δ, будет деревом некоторого вывода, начинающегося символом A и заканчивающегося цепочкой, содержащей вхождение α, и это дерево будет содержать меньше узлов, чем T_0. Далее, при $i < j$ узлы γ_i и γ_j не могут быть помечены одним и тем же символом, иначе дерево $\mathrm{Sub}(T_0, T^i, T^j)$, где T^i и T^j — полные γ_i- и γ_j-поддеревья T_0 соответственно, также было бы деревом вывода из A цепочки, содержащей α, и имело бы меньше узлов, чем T_0. Поэтому $s \leqslant \mu(W)$; но s есть как раз число невисячих вершин T_0, т. е. длина соответствующего вывода. Теперь искомый алгоритм очевиден — достаточно

перебрать все выводы в Г, длины которых не превосходят $\mu(W)$.

Назовем вспомогательный символ Б-грамматики Г бесплодным, если из него не выводима никакая цепочка в основном словаре Г.

Лемма 4.7. *Существует алгоритм, позволяющий по любой* Б-*грамматике* Г *и любому ее вспомогательному символу* A *распознать, является ли* A *бесплодным.*

Доказательство. Пусть $\Gamma = \langle V, W, I, R\rangle$ — Б-грамматика, и для некоторого символа $A \in W$ существуют выводы, начинающиеся данным символом и заканчивающиеся цепочками в V. Возьмем тот из этих выводов, чье дерево — обозначим его T_0 — содержит меньше всего узлов (если такое дерево не одно, то берем любое). Дерево T_0 будет простым. В самом деле, если это не так, то в нем найдется путь $\delta_1, \delta_2, \ldots, \delta_r$ такой, что δ_1 и δ_r помечены одним и тем же символом и δ_r — невисячий узел; но тогда дерево $\mathrm{Sub}(T_0, T', T'')$, где T' и T'' — полные δ_1- и δ_r-поддеревья T_0 соответственно, будет содержать меньше узлов, чем T_0, и тоже будет деревом вывода, начинающегося символом A и заканчивающегося цепочкой в V. По лемме 4.5 высота T_0 не превышает $\mu(W)$, откуда по лемме 4.4 можно найти верхнюю границу для длины соответствующего вывода. Искомый алгоритм состоит, таким образом, в переборе всех выводов в Г, длины которых не превосходят этой границы.

Поскольку бесплодность начального символа Б-грамматики означает пустоту порождаемого ею языка, из леммы 4.7 вытекает

Теорема 4.1. *Существует алгоритм, позволяющий по любой* Б-*грамматике* Г *распознать, является ли язык* $L(\Gamma)$ *пустым.*

Пусть $\Gamma = \langle V, W, I, R\rangle$ — Б-грамматика и $Z \subseteq V$. Символ $A \in W$ мы будем называть Z-бесплодным, если из него не выводима никакая цепочка в словаре Z. Буквальным повторением рассуждений, примененных при доказательстве леммы 4.7, получается

Лемма 4.7′. *Существует алгоритм, позволяющий по любой* Б-*грамматике* $\Gamma = \langle V, W, I, R\rangle$, *любому словарю* $Z \subseteq V$ *и любому символу* $A \in W$ *распознать, является ли* A Z-*бесплодным.*

Назовем вспомогательный символ Б-грамматики Г неустранимым, если он а) зависит от начального символа или совпадает с ним и б) не бесплоден. В противном случае вспомогательный символ будет называться устранимым. Б-грамматику, все вспомогательные символы которой неустранимы, будем называть приведенной.

Лемма 4.8. *Для всякой* Б-*грамматики* $\Gamma = \langle V, W, I, R\rangle$, *порождающей непустой язык, можно построить эквивалентную ей приведенную* Б-*грамматику* $\Gamma = \langle V, W', I, R'\rangle$ *такую, что* $W' \subseteq W$ *и* $R' \subseteq R$.

Доказательство. Из определения ясно, что ни один узел дерева полного вывода не может быть помечен устранимым вспомогательным символом. Поэтому, выбросив из вспомогательного словаря все устранимые символы, а из схемы — все правила, содержащие эти символы в левых или правых частях, мы не изменим порождаемого грамматикой языка. Полученная так грамматика не обязана еще быть приведенной*), но в случае надобности к ней можно применить такое же преобразование, затем так же преобразовать следующую грамматику и т. д. Этот процесс когда-нибудь оборвется, так как на каждом шаге мощность вспомогательного словаря уменьшается. Поскольку язык $L(\Gamma)$ не пуст, в описанном процессе никогда не будет выброшен начальный символ, и в конце концов получится приведенная грамматика, эквивалентная Г.

Будем теперь называть вспомогательный символ Б-грамматики циклическим, если он зависит от самого себя. Имеет место

Лемма 4.9. *Приведенная* Б-*грамматика без правил вида* $A \to B$, *где* A *и* B *— вспомогательные символы, тогда и только тогда порождает бесконечный язык, когда хотя бы один ее вспомогательный символ является циклическим.*

Доказательство. 1) Пусть Г — данная грамматика и A — циклический вспомогательный символ. Тогда

*) Символ, неустранимый в старой грамматике, может стать устранимым в новой. Например, в грамматике $\Gamma = \langle\{a\}, \{I, A, B\}, I, \{I \to a, I \to AB, A \to a\}\rangle$ устранимым символом является только B, но после изъятия этого символа и правила $I \to AB$ символ A тоже становится устранимым.

существуют такие цепочки φ и ψ, что $A \underset{\Gamma}{\vdash} \varphi A \psi$, откуда для любого $n = 1, 2, \ldots$ имеет место $A \underset{\Gamma}{\vdash} \varphi^n A \psi^n$. В выводе цепочки $\varphi A \psi$ из A применяются правила, содержащие в правых частях вспомогательные символы; но длины правых частей таких правил больше единицы. Поэтому $\varphi\psi \neq \Lambda$. Поскольку грамматика Γ приведенная, при $A \neq I$ (I — начальный символ) найдутся такие цепочки ξ и η, что $I \underset{\Gamma}{\vdash} \xi A \eta$. Кроме того, найдутся такие цепочки z, u, v, x, y в основном словаре, что $A \underset{\Gamma}{\vdash} z$, $\varphi \underset{\Gamma}{\models} u$, $\psi \underset{\Gamma}{\models} v$, $\xi \underset{\Gamma}{\models} x$, $\eta \underset{\Gamma}{\models} y$ ($\underset{\Gamma}{\models}$ означает «выводима в Γ из или совпадает с»), причем $uv \neq \Lambda$. Поэтому при любом $n = 1, 2, \ldots$ из I будет выводима цепочка $w_n = xu^n z v^n y$ (так при $A \neq I$, при $A = I$ имеем вместо этого цепочку $u^n z v^n$), и при $n_1 \neq n_2$ цепочки w_{n_1} и w_{n_2} различны.

2) Если в грамматике нет циклических вспомогательных символов, то все деревья выводов просты, и их число, а тем более число выводимых из начального символа цепочек, конечно.

Теорема 4.2. *Существует алгоритм, позволяющий по любой* Б-*грамматике* Γ *распознать, является ли язык* $L(\Gamma)$ *конечным.*

Доказательство. В силу леммы 4.9 для выяснения вопроса о конечности языка $L(\Gamma)$ достаточно перестроить Γ в соответствии с леммами 4.1 и 4.8, а затем проверить все вспомогательные символы на цикличность (лемма 4.6).

Лемма 4.10. *Для любой* Б-*грамматики* $\Gamma = \langle V, W, I, R\rangle$ *и любого словаря* $Z \subseteq V$ *можно построить* Б-*грамматику, порождающую язык* $\Pi p_Z L(\Gamma) - \{\Lambda\}$.

Доказательство. В силу леммы 4.3 мы можем считать, что Γ — стандартная бинарная Б-грамматика. Назовем символ $A \in W$ исчезающим, если он не является $(V - Z)$-бесплодным. Положим $\Gamma' = \langle Z, W, I, R_1 \cup R_2\rangle$, где R_1 получается из R изъятием правил вида $A \to a$, $a \notin Z$, а R_2 состоит из всевозможных правил вида $A \to B$ таких, что для некоторого исчезающего символа C схема R содержит правило $A \to BC$ или $A \to CB$. Покажем, что Γ' — нужная грамматика.

а) Пусть $x \in L(\Gamma')$. Тогда, заменив в произвольном выводе цепочки x из I в Γ' каждое применение правила вида $A \to B$, $A, B \in W$, применением соответствующего правила $A \to BC$ или $A \to CB$, мы получим вывод в Γ, заканчивающийся некоторой цепочкой вида $x_1 C_1 x_2 C_2 \ldots$ $\ldots x_t C_t x_{t+1}$, где $C_1, \ldots, C_t$ — исчезающие символы и $x_1 \ldots x_{t+1} = x$. Из нее, в свою очередь, можно вывести в Γ цепочку вида $x_1 u_1 x_2 u_2 \ldots x_t u_t x_{t+1}$, где $u_1, \ldots, u_t \in (V - Z)^*$; но проекция такой цепочки на Z есть x. Итак, $x \in \Pi p_Z L(\Gamma)$, а поскольку $\Lambda \notin L(\Gamma')$, имеем также $x \in \Pi p_Z L(\Gamma) - \{\Lambda\}$. б) Пусть $x \in \Pi p_Z L(\Gamma) - \{\Lambda\}$. Рассмотрим цепочку $y \in L(\Gamma)$ такую, что $\Pi p_Z y = x$, и дерево T некоторого вывода цепочки y в Γ. Выбросим из T все узлы, обладающие тем свойством, что все «последние потомки» данного узла A (т. е. потомки, являющиеся висячими узлами) помечены символами из $V - Z$, но среди «последних потомков» узла, подчиняющего A, некоторые помечены символами из Z, а также всех потомков узлов, обладающих этим свойством. Ясно, что, во-первых, при этом будут выброшены все те и только те узлы, среди «последних потомков» которых ни один не помечен символом из Z, а поэтому на висячих узлах полученного дерева будет «написана» в точности цепочка x; во-вторых, полученное дерево будет деревом некоторого полного вывода в Γ'. Итак, $x \in L(\Gamma')$.

Выведем из доказанной леммы следующее любопытное утверждение.

Будем называть грамматику $\Gamma = \langle V, W, I, R\rangle$ обобщенной Б-грамматикой (ОБ-грамматикой), если каждое ее правило имеет вид $A \to \omega$, где $A \in W$, $\omega \in (V \cup W)^*$. Язык, порождаемый ОБ-грамматикой, будем называть ОБ-языком.

Теорема 4.3. *Для всякой* ОБ-*грамматики* Γ *можно построить* Б-*грамматику* Γ' *такую, что* $L(\Gamma') = L(\Gamma) - \{\Lambda\}$.

Доказательство. Добавив к основному словарю V ОБ-грамматики Γ новый символ c и заменив каждое правило вида $A \to \Lambda$ правилом $A \to c$, получим Б-грамматику Γ_c такую, что $\Pi p_V L(\Gamma_c) = L(\Gamma)$. Остается воспользоваться леммой 4.10.

(Полезно сравнить эту теорему с результатом упражнения 3.18.)

Из теоремы 4.3 вместе с леммой 4.2 получаем

Следствие. *Для всякой* ОБ-*грамматики* Γ *можно построить эквивалентную ей* ОБ-*грамматику* Γ' *такую, что правые части ее правил не содержат начального символа и всякое ее правило, левая часть которого отлична от начального символа, имеет непустую правую часть.*

Теорема 4.4. *Гомоморфный образ любого* ОБ-*языка есть* ОБ-*язык. При этом для любой* ОБ-*грамматики с основным словарем* $\{a_1, \ldots, a_n\}$ *и любых* n *цепочек* $y_1, \ldots, y_n$ *в произвольном словаре* Z *можно построить* ОБ-*грамматику, порождающую язык* $\psi(L(\Gamma))$, *где* ψ *— гомоморфное отображение, определяемое условием* $\psi(a_i) = y_i$, $i = 1, \ldots, n$.

Доказательство. Пусть $\Gamma = \langle \{a_1, \ldots, a_n\}, W, I, R\rangle$. Без ограничения общности можно считать, что $(\{a_1, \ldots, a_n\} \cup W) \cap Z = \varnothing$. Положим $\Gamma' = \langle Z, W \cup \{a_1, \ldots, a_n\}, I, R \cup \{a_1 \to y_1, \ldots, a_n \to y_n\}\rangle$. Γ' есть ОБ-грамматика, и $L(\Gamma') = \psi(L(\Gamma))$.

Полезно заметить, что понятие дерева вывода можно естественным образом распространить на случай ОБ-грамматики. Именно, рассмотрим произвольное дерево вывода цепочки y из символа B в грамматике Γ_c, использованной в доказательстве теоремы 4.3, и устраним из него все висячие узлы, помеченные символом c. Всякое полученное так P_1-дерево будет, по определению, деревом вывода цепочки $\Pi p_V y$ из B в Γ (V — основной словарь Γ). Можно было бы определить дерево вывода и прямо по выводу в ОБ-грамматике (и даже в ОНС-грамматике — см. упражнение 3.18), непосредственно обобщив определение из § 3.1; это предоставляется читателю. Висячие узлы с нетерминальными *) метками будут тогда соответствовать точкам применения правил вида $A \to \Lambda$ (для произвольных ОНС-грамматик — правил вида $\xi A \eta \to \xi\eta$).

Очевидным образом распространяется на ОБ-грамматики и понятие однозначности (см. стр. 83).

§ 4.3. Необходимые условия бесконтекстности

Когда нужно доказать, что тот или иной язык не является бесконтекстным, часто оказывается полезной следующая

*) См. сноску **) на стр. 27.

Теорема 4.5. *Для любого бесконечного* Б-*языка* L *найдутся такие натуральные числа* r *и* s, *эффективно определяемые по порождающей* L Б-*грамматике, что любая цепочка* $w \in L$, *длина которой больше* r, *может быть представлена в виде*

$$w = x_1 y_1 z y_2 x_2, \tag{1}$$

где: (а) $y_1 y_2 \neq \Lambda$; (б) $|y_1 z y_2| \leqslant s$; (в) *при любом* $n = 1, 2, \ldots$ *цепочка* $w_n = x_1 y_1^n z y_2^n x_2$ *принадлежит* L.

Доказательство. Покажем сначала, что если $\Gamma = \langle V, W, I, R\rangle$ — Б-грамматика без правил вида $A \to B$, $A, B \in W$, то для любой цепочки $w \in L(\Gamma)$ такой, что ни одно ее дерево вывода не может быть простым, имеется представление вида (1), удовлетворяющее условиям (а) и (в), а также условию: (б′) для некоторого символа $A \in W$ существует $(A, y_1 z y_2)$-дерево, высота которого не превосходит $\mu(W) + 1$. Действительно, пусть w — цепочка, обладающая указанным свойством, T — какое-либо ее дерево вывода и C — индуцируемая этим деревом система составляющих для w. В дереве T найдутся два различных узла α и β, помеченных одним и тем же вспомогательным символом A и таких, что из α в β идет путь. Обозначим составляющие системы C, происходящие от узлов α и β, — точнее, отвечающие им подцепочки w — через u и z соответственно. Для подходящих цепочек x_1, x_2, y_1, y_2 имеем, очевидно, $u = y_1 z y_2$, $w = x_1 u x_2 = x_1 y_1 z y_2 x_2$. Если U' и U'' — соответственно полные α- и β-поддеревья T, то U' есть (A, u)-дерево и U'' есть (A, z)-дерево. Положим $U_1 = U'$, $U_2 = \mathrm{Sub}(U_1, U'', U')$, $U_3 = \mathrm{Sub}(U_2, U'', U')$ и т. д. Для каждого $n = 1, 2, \ldots$ U_n является, очевидно, $(A, y_1^n z y_2^n)$-деревом. Поэтому дерево $T_n = \mathrm{Sub}(T, U', U_n)$ есть $(I, x_1 y_1^n z y_2^n x_2)$-дерево, т. е. дерево вывода цепочки $x_1 y_1^n z y_2^n x_2$ в Γ. Мы получили, таким образом, представление цепочки w в виде (1), удовлетворяющее условию (в). Выполнение условия (а) вытекает из того, что Γ не имеет правил вида $A \to B$, $A, B \in W$. Что касается условия (б′), то оно будет выполняться, если U' — квазипростое дерево (лемма 4.5). Если U' — не квазипростое дерево, то в нем найдутся два различных узла α_1 и β_1, отличных от корня, помеченных одним и тем же

вспомогательным символом и таких, что из α_1 в β_1 идет путь. Мы можем теперь заменить в нашем рассуждении α и β на α_1 и β_1, и если полное α_1-поддерево T квазипростое, то условие (б′) будет выполняться; в противном случае мы заменим α_1 и β_1 новыми узлами α_2 и β_2 и т. д.; поскольку в этом процессе высоты поддеревьев будут уменьшаться, он рано или поздно оборвется. Итак, наше вспомогательное утверждение доказано.

Пусть теперь L — бесконечный Б-язык. По лемме 4.1 существует Б-грамматика $\Gamma = \langle V, W, I, R\rangle$ без правил вида $A \to B$, $A, B \in W$, такая, что $L(\Gamma) = L$. Положим $\mu(W) = p$, $\max\limits_{\exists A\,[A \to \psi \in R]} |\psi| = g$. В силу лемм 4.4 и 4.5 длина вывода в Γ, имеющего простое дерево, не может превосходить числа $\frac{g^p - 1}{g - 1}$; в то же время никакую цепочку w нельзя вывести в Γ менее чем за $\frac{|w| - 1}{g}$ шагов. Поэтому цепочка, длина которой больше $g \cdot \frac{g^p - 1}{g - 1} + 1$, не может обладать в Γ простым деревом вывода. Точно так же заключаем, что если некоторая цепочка u для подходящего $A \in W$ имеет (A, u)-дерево высоты, не превосходящей $p + 1$, то $|u| \leqslant g \cdot \frac{g^{p+1} - 1}{g - 1} + 1$. Поэтому, положив $r = g \cdot \frac{g^p - 1}{g - 1} + 1$ и $s = g \cdot \frac{g^{p+1} - 1}{g - 1} + 1$, мы получаем для произвольной цепочки $w \in L(\Gamma)$, $|w| > r$, представление (1), удовлетворяющее условиям (а), (б), (в).

Следствие. *Множество длин цепочек* Б-*языка либо конечно, либо содержит арифметическую прогрессию.*

Из этого следствия вытекает, в частности, что языки примера 13 из § 1.3 и примера 1 из § 3.4 не бесконтекстные.

Приведем примеры применения теоремы 4.5 в случаях, когда следствием воспользоваться нельзя.

Пример 1. Пусть L — язык примера 10 из § 1.3. Если бы L был бесконтекстным, то для всякого достаточно большого k и любого $n = 1, 2, \ldots$ при подходящих t, x_1, x_2, y_1, y_2, z, $y_1 y_2 \neq \Lambda$, мы имели бы $a^k b^k a^k = x_1 y_1 z y_2 x_2$, $x_1 y_1^n z y_2^n x_2 = a^t b^t a^t$. Но если в цепочке $a^k b^k a^k$ каждая из подцепочек y_1 и y_2 состоит только из a или только из b, то ни при каком $n > 1$ цепочка $x_1 y_1^n z y_2^n x_2$ не может иметь вида $a^t b^t a^t$. Если же, например, y_1 содержит и a и b, то уже цепочка $x_1 y_1^2 z y_2^2 x_2$ будет содержать подцепочку вида $ba^s b$.

Пример 2. Пусть L — язык примера 11 из § 1.3. Для любого $k = 1, 2, \ldots$ имеем $a_1^k a_2^k b a_1^k a_2^k \in L$. Если L — Б-язык, то найдется $s > 0$ такое, что для любого достаточно большого k и любого $n = 1, 2, \ldots$ будем иметь при подходящих x_1, x_2, y_1, y_2, z

$$w_1 = a_1^k a_2^k b a_1^k a_2^k = x_1 y_1 z y_2 x_2,$$
$$w_n = x_1 y_1^n z y_2^n x_2 \in L, \quad |y_1 z y_2| \leqslant s.$$

Но если цепочка $y_1 z y_2$ не содержит вхождений b, то уже w_2 не будет делиться вхождением b пополам и, значит, не будет принадлежать L. Цепочки y_1 и y_2 не могут содержать b, иначе в w_n было бы не менее n вхождений b. Итак, вхождение b должно содержаться в z; поэтому при $k \geqslant s - 1$ цепочка y_1 должна состоять только из a_2, а y_2 — только из a_1. Отсюда $w_2 = a_1^k a_2^{k+l_1} b a_1^{k+l_2} a_2^k$, где $l_1 = |y_1|$, $l_2 = |y_2|$. Но $a_1^k a_2^{k+l_1} \neq a_1^{k+l_2} a_2^k$.

Пример 3. Пусть $L = \{a^n b^n a^n c^m \mid m, n = 1, 2, \ldots\}$. С помощью леммы 4.10 этот пример сводится к примеру 1.

Чтобы сформулировать другой критерий бесконтекстности, введем в рассмотрение n-мерные векторы, компонентами которых служат целые неотрицательные числа; для краткости будем говорить просто о векторах. Число n будет все время считаться фиксированным.

Назовем множество векторов M линейным, если существуют векторы $\alpha_1, \ldots, \alpha_s, \beta$ $(s = 0, 1, \ldots)$ такие, что $M = \{m_1\alpha_1 + \ldots + m_s\alpha_s + \beta \mid m_1, \ldots, m_s = 0, 1, \ldots\}$. (Сложение векторов и умножение на число имеют обычный смысл.) Конечную последовательность $\langle \alpha_1, \ldots, \alpha_s, \beta\rangle$ будем называть базой M. Множество векторов, являющееся объединением конечного числа

линейных множеств, называется полулинейным. База полулинейного множества есть, по определению, система баз его линейных слагаемых. Пустое множество также будем считать полулинейным (с пустой базой).

Для произвольного языка L в словаре $\{a_1, \ldots, a_n\}$ будем обозначать через $K(L)$ множество векторов

$$\{(k_1, \ldots, k_n) \mid \exists x (x \in L \,\&\, |x|_1 = k_1 \,\&\, \ldots \,\&\, |x|_n = k_n)\},$$

где $|x|_i$ означает $|x|_{a_i}$.

Теорема 4.6. *Для любого* Б-*языка* L *множество* $K(L)$ *полулинейно и его базу можно найти эффективно по* Б-*грамматике, порождающей* L.

Доказательство. Пусть $\Gamma = \langle V, W, I, R\rangle$ — Б-грамматика и T — дерево вывода в Г. Будем обозначать через $M(T)$ множество всех тех символов из W, которые встречаются в T в качестве меток при узлах. Для произвольного $M \subseteq W$ рассмотрим множества $\mathfrak{A}(M)$ и $\mathfrak{B}(M)$, состоящие соответственно из всех тех деревьев вывода T, для которых $M(T) = M$, и всех тех цепочек $x \in V^*$, для которых существуют (I, x)-деревья, принадлежащие $\mathfrak{A}(M)$. Положим также $\mathfrak{C}(M) = K(\mathfrak{B}(M))$. Нам достаточно установить, что для каждого $M \subseteq W$ множество $\mathfrak{C}(M)$ полулинейно, и указать способ нахождения его базы.

Для $M = \varnothing$ утверждение тривиально. Пусть $M \subseteq W$ не пусто и утверждение доказано для всех собственных подмножеств M; докажем его для M.

Будем называть (A, x)-дерево терминальным, если $x \in V^*$. Прямым элементарным преобразованием (п. э. п.) терминального (A, x)-дерева T назовем операцию, переводящую T в $\mathrm{Sub}(T, T', T'')$, где T' — некоторое простое терминальное полное поддерево T, а T'' — квазипростое терминальное дерево, содержащее T' в качестве собственного полного поддерева. Операцию, обратную этой, будем называть обратным элементарным преобразователем (о. э. п.). П. э. п. определяется парой деревьев T', T''; каждому п. э. п. π мы сопоставим вектор $\lambda(\pi)$, определяемый следующим образом: если T' есть (B, z)-дерево, а T'' — $(B, y_1 z y_2)$-дерево, то $\lambda(\pi) = (|y_1 y_2|_1, \ldots, |y_1 y_2|_n)$. Всевозможных п. э. п. имеется, очевидно, лишь конечное число; мы обозначим их $\pi_1, \ldots, \pi_t$.

Для произвольного терминального (I, x)-дерева T положим $\varkappa(T) = (|x|_1, \ldots, |x|_n)$.

Назовем дерево $T \in \mathfrak{A}(M)$ минимальным 1-го рода, если к нему неприменимо никакое о. э. п., и минимальным 2-го рода, если некоторое о. э. п. преобразует его в дерево, не принадлежащее $\mathfrak{A}(M)$. Очевидно, всякое $T \in \mathfrak{A}(M)$ можно получить из некоторого минимального дерева 1-го или 2-го рода последовательным применением п. э. п. Поэтому $\mathfrak{C}(M)$ представляется в виде $\{n_1\lambda(\pi_1) + \ldots + n_t\lambda(\pi_t) + \varkappa(T) \mid T$ — минимальное дерево 1-го или 2-го рода$\}$. Поскольку число минимальных деревьев 1-го рода конечно, достаточно доказать полулинейность множества $\mathfrak{C}'(M)$, отличающегося от $\mathfrak{C}(M)$ тем, что в качестве T берутся лишь минимальные деревья 2-го рода. Но очевидно, что если множество векторов $\mathfrak{D}_1$ полулинейно, то множество $\mathfrak{D}_2 = \{n_1\alpha_1 + \ldots + n_s\alpha_s + \beta \mid \beta \in \mathfrak{D}_1;\ n_1, \ldots, n_t = 0, 1, \ldots\}$, где $\alpha_1, \ldots, \alpha_t$ — заданный набор векторов, также полулинейно, причем базу $\mathfrak{D}_2$ можно найти эффективно по базе $\mathfrak{D}_1$ и векторам $\alpha_1, \ldots, \alpha_s$. Итак, $\mathfrak{C}(M)$ будет полулинейным, если таковым окажется множество $\mathfrak{C}_0(M) = \{\varkappa(T) \mid T$ — минимальное дерево 2-го рода$\}$, а поскольку все минимальные деревья 1-го рода и все п. э. п. могут быть построены эффективно, для нахождения базы $\mathfrak{C}(M)$ достаточно уметь строить базу $\mathfrak{C}_0(M)$. Но $\mathfrak{C}_0(M)$ представляется как объединение конечного числа множеств вида $\{\lambda(\pi_i) + \varkappa(T) \mid T \in M'\}$, где π_i — фиксированное п. э. п. и M' — фиксированное собственное подмножество M. Полулинейность такого множества сразу следует из полулинейности $\mathfrak{C}(M')$, и база его очевидным образом строится по базе $\mathfrak{C}(M')$.

Замечание. Если словарь одноэлементен, можно отождествлять как цепочки, так и векторы с натуральными числами. Полулинейное множество будет тогда объединением конечного числа арифметических прогрессий и конечного множества. Но такое множество есть A-язык (пример 3, б) из § 1.3). Таким образом, всякий Б-язык в одноэлементном словаре автоматный. Сверх того, по Б-грамматике с одноэлементным основным словарем эквивалентная ей A-грамматика строится эффективно.

Докажем теперь еще одно утверждение, родственное теореме 4.5; оно особенно пригодится нам в следующем параграфе.

Теорема 4.7. *Для любого бесконечного* Б*-языка* L *найдется такое натуральное число* s*, эффективно определяемое по порождающей* L Б*-грамматике, что, каковы бы ни были цепочки* x, y, z *такие, что* $xyz \in L$ *и* $|y| \geqslant s$*, цепочку* y *можно представить в виде* $y = y_1vy_2$ *так, что* $v \neq \Lambda$ *и либо* а) *цепочка* xy_1 *представима в виде* $xy_1 = x_1ux_2$*, где* $x_1u^nx_2v^ny_2z \in L$ *при любом* $n = 1, 2, \ldots$*, либо* б) *цепочка* y_2z *представима в виде* $y_2z = z_1wz_2$*, где* $xy_1v^nz_1w^nz_2 \in L$ *при любом* $n = 1, 2, \ldots$

Доказательство. Пусть Γ — Б-грамматика, порождающая L, и пусть g и p — соответственно максимум длин правых частей правил и мощность вспомогательного словаря Γ. Положим $s = 2(p+1)\cdot g^{2(p+1)}$. Рассмотрим цепочку $xyz \in L$ такую, что $|y| \geqslant s$. Пусть T — некоторое дерево вывода этой цепочки в Γ и C — соответствующая система составляющих. Поскольку ширина системы C не превосходит g, мы можем по лемме П1.1 найти такие точки $\alpha_1 \leqslant \ldots \leqslant \alpha_{p+1} < \beta_{p+1} \leqslant \ldots \leqslant \beta_1$ цепочки xyz, что $[\alpha_1, \beta_1], \ldots, [\alpha_{p+1}, \beta_{p+1}] \in C$ и либо $\alpha_1 < \ldots < \alpha_{p+1}$ и $\alpha_1, \ldots, \alpha_{p+1}$ — точки отрезка y, либо $\beta_1 > \ldots > \beta_{p+1}$ и $\beta_1, \ldots, \beta_{p+1}$ — точки отрезка y. Пусть для определенности имеет место первое. В силу выбора числа p найдутся такие i, j, $1 \leqslant i < j \leqslant p+1$, что узлы A_1 и A_2 дерева T, от которых происходят составляющие $[\alpha_i, \beta_i]$ и $[\alpha_j, \beta_j]$, помечены одним и тем же символом B. Обозначим полные A_1- и A_2-поддеревья дерева T через T_1 и T_2 соответственно. Очевидно, T_2 есть поддерево T_1; поэтому, если T_1 — (B, t_1)-дерево и T_2 — (B, t_2)-дерево, то t_2 — подцепочка t_1 и цепочка xyz представима в виде $xyz = x'vt_2wz_2$, где $vt_2w = t_1$ и отрезки v, t_2, w и z_2 начинаются в точках α_i, α_j, $\beta_j + 1$ и $\beta_i + 1$ соответственно (здесь допущена вольность в обозначениях). Поскольку $\alpha_i < \alpha_j$ и обе эти точки расположены в отрезке y, для подходящих y_1 и y_2 имеем $y = y_1vy_2$, $x' = xy_1$, $t_2wz_2 = y_2z$; при этом $v \neq \Lambda$. Если теперь положить $U_1 = T_1$, $U_{n+1} = \mathrm{Sub}(U_n, T_2, T_1)$, то при любом $n = 1, 2, \ldots$ U_n будет $(B, v^nt_2w^n)$-деревом, а $T_n = \mathrm{Sub}(T, T_1, U_n)$ будет $(I, xy_1v^nt_2w^nz_2)$-деревом, где I — начальный символ Γ, так что $xy_1v^nt_2w^nz_2 \in L$. Таким образом, положив $z_1 = t_2$, получаем искомое представление.

Пример 4. Пусть $L = \{a^nb^na^m \mid m, n = 1, 2, \ldots; m \leqslant n\}$. Теоремы 4.5 и 4.6 к языку L неприменимы; действительно, требуемое теоремой 4.5 представление цепочки $a^nb^na^m$ получается при $x_1 = a^{n-1}$, $y_1 = a$, $z = \Lambda$, $y_2 = b$, $x_2 = b^{n-1}a^m$, и $K(L) = \{p\cdot(1,1) + q\cdot(2,1) + (2,1) \mid p, q = 0, 1, \ldots\}$. В то же время теорема 4.7 позволяет установить, что L — не Б-язык. Именно, если бы цепочка $a^nb^na^m$, $m \leqslant n$, допускала требуемое этой теоремой представление при $x = a^nb^n$, $y = a^m$, $z = \Lambda$, то цепочка u или w во всяком случае состояла бы либо целиком из a, либо целиком из b; но если бы она при этом была пуста или содержалась в y, то язык L содержал бы любые цепочки вида $a^nb^na^{m+ic}$, где $c > 0$ — постоянная и $i = 1, 2, \ldots$, а в противном случае в L нашлась бы цепочка вида $a^pb^qa^r$, где $p \neq q$.

Дальнейшие примеры см. в упражнениях 4.9 и 4.11.

Замечание. Легко видеть, что полученное в доказательстве теоремы 4.7 представление обладает следующими свойствами: а) существует система составляющих C цепочки xyz, отвечающая некоторому ее дереву вывода в Γ и содержащая отрезок ux_2v, соответственно vz_1w; б) для каждой цепочки $x_1u^nx_2v^ny_2z$, соответственно $xy_1v^nz_1w^nz_2$, $n = 1, 2, \ldots$, существует система составляющих C_n, отвечающая некоторому дереву вывода этой цепочки в Γ и содержащая любой отрезок, полученный из произвольной составляющей системы C, содержащей отрезок $ux_2v(vz_1w)$, заменой этого отрезка на $u^nx_2v^n(v^nz_1w^n)$; в частности, сам отрезок $u^nx_2v^n(v^nz_1w^n)$ принадлежит C_n (как и все отрезки $u^iz_2v^i$, соответственно $v^iz_1w^i$, $i \leqslant n$).

В заключение параграфа коснемся вопроса о замкнутости класса Б-языков относительно основных операций.

Теорема 4.8. а) *Класс* Б*-языков эффективно замкнут относительно операций объединения, умножения, усеченной итерации и подстановки*; б) *класс* Б*-языков не замкнут относительно операций пересечения, вычитания и взятия дополнения.*

Доказательство. а) Конструкции, использованные в доказательстве теоремы 1.1 для объединения

и умножения, в применении к Б-грамматикам дают снова Б-грамматики. Грамматика, порождающая $L(\Gamma)^+$, получается из Б-грамматики Г, удовлетворяющей требованию леммы 4.2, добавлением правила $I \to II$ (I — начальный символ). Доказательство для подстановки предоставляется читателю. б) Незамкнутость класса Б-языков относительно пересечения следует хотя бы из примера 6 § 1.3 и примера 1 настоящего параграфа, поскольку $\{a^nb^na^n \mid n=1, 2, \ldots\} = \{a^nb^na^m \mid m, n=1, 2, \ldots\} \cap \cap \{a^mb^na^n \mid m, n = 1, 2, \ldots\}$. Незамкнутость относительно вычитания и взятия дополнения вытекает из замкнутости относительно объединения и незамкнутости относительно пересечения.

По поводу левого и правого деления см. упражнение 4.12.

§ 4.4. Неоднозначность

Пусть Г — произвольная Б-грамматика. В силу рассуждений, приведенных в конце § 4.1, множество различных полных выводов в Г распадается на попарно непересекающиеся классы, каждый из которых отвечает одному дереву вывода; различие между выводами, принадлежащими одному классу, в известном смысле несущественно. Поэтому введенное в § 3.1 понятие однозначности приобретает в данном случае особенно простой смысл: однозначная Б-грамматика характеризуется тем, что в ней каждая цепочка порождаемого языка выводима, по существу, единственным способом. (Для произвольных НС-грамматик это не так — см. упражнение 4.5.)

Если некоторый язык порождается неоднозначной Б-грамматикой, то для него, вообще говоря, может найтись другая Б-грамматика, являющаяся однозначной. (Например, язык $\{a^nb^m \mid m, n = 1, 2, \ldots; n \leqslant m \leqslant 2n\}$ порождается неоднозначной Б-грамматикой со схемой $\{I \to aIB, I \to aB, B \to b, B \to bb\}$ и однозначной Б-грамматикой со схемой $\{I \to aIbb, I \to abb, I \to A, A \to aAb, A \to ab\}$.) В связи с этим естественно ввести следующее понятие. Б-язык (соответственно ОБ-язык) называется однозначным, если существует хотя бы одна порождающая его однозначная Б-грамматика (соответственно ОБ-грамматика). В противном случае Б-язык называется неоднозначным (или существенно неоднозначным). Цель настоящего параграфа — указать примеры неоднозначных Б-языков.

Пример 1. Пусть $L_1 = \{a^nb^nc^m \mid m, n = 1, 2, \ldots\}$, $L_2 = \{a^mb^nc^n \mid m, n = 1, 2, \ldots\}$, $L = L_1 \cup L_2$. Язык L порождается, например, грамматикой со схемой $\{I \to A_1, I \to A_2, A_1 \to A_1c, A_1 \to B_1c, B_1 \to aB_1b, B_1 \to ab, A_2 \to aA_2, A_2 \to aB_2, B_2 \to bB_2c, B_2 \to bc\}$. Неоднозначность этой грамматики очевидна — всякая цепочка вида $a^nb^nc^n$ имеет в ней два дерева вывода. [Так, для цепочки $a^3b^3c^3$ одно дерево вывода дает систему составляющих $(((a(a(ab)b)b)c)c)c$, другое — систему составляющих $a(a(a(b(b(bc)c)c)))$.] Покажем, что и любая Б-грамматика, порождающая L, является неоднозначной.

Пусть Г — такая грамматика. Найдем для нее число s из теоремы 4.7 и положим $q = 3s!$. Рассмотрим цепочку $a^sb^sc^{s+q}$ и применим к ней теорему 4.7 при $x = a^s$, $y = b^s$, $z = c^{s+q}$. Случай б) (из формулировки теоремы 4.7) исключается, поскольку соответствующая цепочка w имела бы вид b^k, $0 \leqslant k < s$, или c^l, $0 \leqslant l \leqslant s + q$, а так как $v = b^i$, $0 < i \leqslant s$, мы имели бы либо $a^sb^{s+i+k}c^{s+q} \in L$, либо $a^sb^{s+i}c^{s+q+l} \in L$, но в обеих этих цепочках как степени a и b, так и степени b и c различны. Поэтому существуют представления $b^s = y_1vy_2$, $a^sy_1 = x_1ux_2$, соответствующие случаю а). При этом цепочка u должна, очевидно, состоять либо только из a, либо только из b; второе, однако, невозможно, так как в таком случае язык L содержал бы последовательность цепочек с постоянными степенями a и c и бесконечно возрастающими степенями b. Таким образом, мы получили следующее представление нашей цепочки: $a^sb^sc^{s+q} = a^fa^da^gb^hb^db^mc^{s+q}$, где при любом $n = 1, 2, \ldots$ цепочка $t_n = a^{f+nd+g}b^{h+nd+m}c^{s+q}$ принадлежит L, и в силу замечания к теореме 4.7 в некоторой системе составляющих цепочки t_n, отвечающей какому-то дереву вывода t_n в Г, одна из составляющих будет иметь вид $a^{nd+g}b^{h+nd}$. Но ввиду того, что $0 < d \leqslant s$ и $q = 3s!$, найдется такое n_0, что $q = n_0d$. Имеем тогда $t_{n_0+1} =$

$= a^{f+d+g+n_0 d} b^{h+d+m+n_0 d} c^{s+q} = a^{s+q} b^{s+q} c^{s+q}$, и для некоторого дерева вывода T цепочки $a^{s+q}b^{s+q}c^{s+q}$ в Г одна из составляющих соответствующей системы C будет иметь вид $a^{q+g}b^{q+h}$. Теперь мы можем, рассматривая цепочку $a^{s+q}b^s c^s$ и рассуждая так же, как раньше (с точностью до зеркальной симметрии), найти такое дерево вывода T' той же цепочки $a^{s+q}b^{s+q}c^{s+q}$ в Г, что одна из составляющих соответствующей системы C' будет иметь вид $b^{q+h'}c^{q+g'}$. Поскольку $h,\ h' \leqslant s < q$, составляющие $a^{q+g}b^{q+h}$ и $b^{q+h'}c^{q+g'}$ пересекаются; поэтому системы C и C', а значит и деревья T и T', различны.

Пример 2. Пусть $L_1 = \{xb\hat{x}by \mid x,\ y \in \{a_1, a_2\}^+\}$, $L_2 = \{xb\hat{y}by \mid x,\ y \in \{a_1, a_2\}^+\}$, $L = L_1 \cup L_2$. Построение Б-грамматики, порождающей L, не представляет труда. Если Г — произвольная Б-грамматика, порождающая L, то, определив s и q, как в примере 1, рассматривая цепочки $a_1^s b a_1^s b a_1^{s+q}$ и $a_1^{s+q} b a_1^s b a_1^s$ и рассуждая в точности так же, как в предыдущем примере, мы найдем два различных дерева вывода цепочки $a_1^{s+q} b a_1^{s+q} b a_1^{s+q}$.

Пример 3. Пусть $L_1 = \{a^n b^k a^n b^m \mid k,\ m,\ n = 1, 2, \ldots\}$, $L_2 = \{a^k b^n a^m b^n \mid k,\ m,\ n = 1, 2, \ldots\}$, $L = L_1 \cup L_2$ и Г — Б-грамматика, порождающая L. Определим s и q, как в примере 1; рассмотрев цепочку $a^s b^s a^s b^{s+q}$ и рассуждая, как в предыдущих примерах, мы без труда найдем для цепочки $t = a^{s+q}b^s a^{s+q} b^{s+q}$ такое дерево вывода, что одна из составляющих соответствующей системы будет иметь вид $a^{g+q}b^s a^{h+q}$. Применим к цепочке t теорему 4.7, положив $x = a^{s+q}$, $y = b^s$, $z = a^{s+q}b^{s+q}$. Аналогично предыдущему легко видеть, что соответствующая цепочка u или w не содержит вхождений a. Если при этом она содержится в «правой b-зоне», то цепочка t имеет две частично пересекающиеся составляющие и, следовательно, два различных дерева вывода. Остается случай, когда u или w, так же как и v, содержится в «левой b-зоне». Тогда цепочка ux_2v или vz_1w будет иметь вид b^l, $0 < l \leqslant s$, и при подходящем n_0 цепочка $u^{n_0+1}x_2v^{n_0+1}$ или $v^{n_0+1}z_1w^{n_0+1}$ совпадет с цепочкой b^{l+q}. В силу замечания к теореме 4.7 отрезок цепочки $t_{n_0} = a^{s+q}b^{s+q}a^{s+q}b^{s+q}$, полученный из отрезка $a^{g+q}b^s a^{h+q}$ цепочки t заменой b^l на b^{l+q}, будет составляющей цепочки t_{n_0} в некоторой системе, отвечающей какому-то дереву вывода t_{n_0} в Г. Таким образом, некоторое дерево вывода T цепочки t_{n_0} дает составляющую A, начинающуюся левее конца «левой a-зоны» и заканчивающуюся «в правой a-зоне».

Теперь мы можем, рассуждая симметрично, показать, что либо цепочка $t' = a^{s+q}b^{s+q}a^s b^{s+q}$ имеет два различных дерева вывода, либо некоторое дерево вывода T' той же цепочки t_{n_0} дает составляющую A', начинающуюся в «левой b-зоне» и заканчивающуюся правее начала «правой b-зоны». Поскольку A и A' частично пересекаются, $T \neq T'$.

Дальнейшие примеры — в упражнениях 4.16 и 4.20. (См. также упражнение 5.17, б).)

§ 4.5. Машины с магазинной памятью

Подобно тому как классу произвольных грамматик соответствует класс произвольных машин Тьюринга и классу НС-грамматик — класс машин без растяжения, для Б-грамматик также имеется отвечающий им в том же смысле тип машин Тьюринга — так называемые машины с магазинной памятью.

Э-машина называется машиной с магазинной памятью (МП-машиной), если ее программа содержит только инструкции типов (i) — (iii) (стр. 42). Таким образом, рабочая головка МП-машины всегда находится в крайней справа ячейке; создав однажды рабочую ячейку, головка может уйти от нее (если не уничтожит ее тут же) только вправо и получает возможность «заглянуть» в нее снова лишь в тот момент, когда уничтожает ее. Чем раньше ячейка создана, тем позже она будет уничтожена и тем самым «использована»; это напоминает принцип работы магазина в автоматическом оружии, чем и объясняется название «машина с магазинной памятью».

Пример 1. Пусть M — машина с программой $\{q_1 \# \to q_1,\ q_1 \to Aq_2,\ q_2 a \to q_1,\ q_1 b \to q_3,\ Aq_3 \to q_4,\ q_4 b \to q_3,\ q_4 \# \to q_0\}$ и единственным заключительным состоянием q_0. Легко видеть, что $L(M) = \{a^n b^n \mid n = 1, 2, \ldots\}$.

Пример 2. Язык $\{x\hat{x} \mid x \in \{a_1 \ldots, a_n\}^*\}$ допускается машиной с программой $\{q_1 \# \to q_1,\ q_1 \# \to q_{2n+2},$

$q_1 \to A_i q_{i+1}$, $q_{i+1} a_i \to q_1$, $q_1 a_i \to q_{i+n+1}$, $A_i q_{i+n+1} \to q_{2n+2}$, $q_{2n+2} a_i \to q_{i+n+1}$, $q_{2n+2} \ddagger \to q_0\}$ $(i = 1, \ldots, n)$ и единственным заключительным состоянием q_0.

Пример 3. Язык примера 8 из § 1.3 допускается машиной с программой $\{(1)\ q_1 \to Eq_2$, $(2)\ q_2 \# \to q_2$, $(3)\ q_2 \to Aq_3$, $(4)\ q_3 a \to q_2$, $(5)\ q_2 b \to q_4$, $(6)\ Aq_4 \to q_2$, $(7)\ q_4 \to Bq_5$, $(8)\ q_5 b \to q_7$, $(9)\ q_7 a \to q_6$, $(10)\ Bq_6 \to q_4$, $(11)\ q_2 a \to q_6$, $(12)\ Eq_2 \to q_0\}$ и единственным заключительным состоянием q_0.

Дальнейшие примеры МП-машин см. в упражнении 4.22. Примеры детерминированных МП-машин см. на стр. 152 и в упражнении 4.31.

Назовем МП-машину нормальной, если она имеет единственное заключительное состояние и каждое ее полное вычисление начинается записью на рабочей ленте некоторого символа E, который уничтожается лишь на последнем шаге вычисления.

Лемма 4.11. *Для любой МП-машины можно построить эквивалентную ей нормальную МП-машину.*

Доказательство. Достаточно добавить к множеству состояний данной машины два новых состояния q_1' и q_0, объявив их соответственно начальным и заключительным вместо прежних, к рабочему алфавиту — новый символ E и к программе — новые инструкции $q_1' \to Eq_1$ и $q_f E \to q_0$ для каждого $q_f \in Q_0$ (q_1 — прежнее начальное состояние, Q_0 — множество прежних заключительных состояний).

Полным вычислениям нормальной МП-машины можно естественным образом сопоставить P_2-деревья, аналогичные деревьям выводов в Б-грамматике. Покажем, как это сделать.

Пусть M — нормальная МП-машина с программой $R = \{r_1, \ldots, r_s\}$ и C — некоторое ее полное вычисление. Обозначим через N_1 множество всех рабочих ячеек, возникающих в вычислении C; при этом ячейка считается одной и той же с момента ее возникновения до момента ее уничтожения, но если потом «на том же месте» снова возникает ячейка, то мы считаем ее другой, даже если в ней записывается тот же символ, что и в прежней. Кроме того, через N_0 обозначим множество всех участвующих в вычислении входных ячеек. (Ячейки, занятые символами $\#$ и $\ddagger$, в N_0 и N_1 не входят.) Множеством узлов нашего P_2-дерева будет объединение $N_0 \cup N_1$.

Из рабочей ячейки α в рабочую ячейку β мы проведем дугу, если в момент возникновения ячейки β, а значит, и в продолжение всего времени ее существования, α расположена непосредственно слева от нее. Из рабочей ячейки α во входную ячейку γ пойдет дуга в одном из двух случаев: 1) если последний из «рабочих» шагов, предшествующих тому «входному» шагу, на котором читается содержимое γ, есть шаг, на котором возникает α; 2) если первый из «рабочих» шагов, следующих за шагом, на котором читается содержимое γ, — тот, на котором уничтожается α. (Заметим, что эти случаи не исключают друг друга; их конъюнкция возможна, если правее α не возникает новых ячеек.) Из входных ячеек дуги исходить не будут.

Отношение $\prec$ определим так: $\alpha \prec \beta$, если α и β не соединены путем и в α раньше появляется головка, чем в β. (Для рабочих ячеек имеется в виду первое появление головки.)

Определим, наконец, метки в узлах и на дугах. Метка в узле, т. е. в ячейке, будет совпадать с записанным в этой ячейке символом. Дуга, идущая в рабочую ячейку β, помечается парой чисел (i, j), где i, j — номера инструкций r_i, r_j, применяемых соответственно при создании и уничтожении β. Дуга, идущая во входную ячейку γ, помечается числом k — номером инструкции r_k, применяемой при чтении содержимого γ.

Легко видеть, что полученный так «дважды нагруженный биграф» действительно является P_2-деревом; корнем его служит крайняя слева рабочая ячейка, сохраняющаяся в силу нормальности M в продолжение всего вычисления. Мы будем называть это P_2-дерево деревом вычисления C.

Дерево вычисления можно наглядно представить так. Заменим машину M машиной M_1, имеющей одну «ветвящуюся ленту» и одну головку. «Лента» M_1 представляет собой бесконечное дерево, из каждого узла которого исходит счетное множество дуг, упорядоченное изоморфно натуральному ряду. M_1-вычисление строится по M-вычислению следующим образом. Если в начале M-вычисления на рабочей ленте записываются какие-то символы, то головка M_1 идет от корня дерева по самой левой ветви (т. е. из каждого узла переходит в первый

из подчиненных ему узлов) и пишет в ее узлах те же символы. Когда M начинает уничтожать рабочие ячейки, головка M_1 идет по той же ветви назад (ничего не стирая!); когда M снова начинает создавать рабочие ячейки, головка M_1 переходит на другую ветвь, непосредственно примыкающую к прежней справа, и т. д. Когда M читает входной символ, головка M_1 переходит в первую незанятую ячейку, подчиненную обозреваемой, пишет там тот же символ и сразу возвращается назад. Использованная часть «ветвящейся ленты» будет P_1-деревом, которое мы будем называть упрощенным деревом исходного M-вычисления; если пометить дуги этого дерева номерами и парами номеров соответствующих инструкций, получится P_2-дерево, которое, очевидно, совпадает с деревом исходного M-вычисления.

«Машину с ветвящейся лентой» M_1 мы будем называть древообходящим автоматом (Д-автоматом), отвечающим машине M. Можно считать, что Д-автомат имеет инструкции только двух типов: $q \to \alpha q'$ и $q\alpha \to q'$, означающие соответственно «спуститься этажом ниже, в первую слева незанятую ячейку, и записать там α» и «если в обозреваемой ячейке записано α, подняться этажом выше» (в обоих случаях нужно также изменить состояние q на q'); при этом α может быть как рабочим, так и входным символом, и если α — входной символ, то вслед за инструкцией $q \to \alpha q'$ должна выполняться инструкция вида $\alpha q' \to q''$. Действительно, каждую инструкцию вида $qb \to q'$ можно заменить парой инструкций $q \to bq''$, $bq'' \to q'$, где q'' — новое состояние.

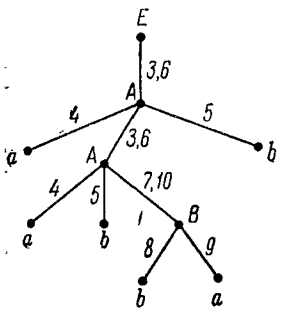


Рис. 6.

Пример. На рис. 6 изображено дерево одного из возможных полных $aabbab$-вычислений в машине примера 3 на стр. 138.

Упрощенное дерево полного x-вычисления позволяет сопоставить цепочке x размеченную систему составляющих в точности тем же способом, как это делается для дерева вывода. Так, дерево рис. 6 дает для цепочки $aabbab$ систему $({}_{E,A}a({}_{A}ab({}_{B}ba))\,b)$. (Метки неточечных составляющих указаны здесь в виде индексов при левых скобках; для точечных составляющих метками служат соответствующие входные символы. В общем случае точечные составляющие могут иметь и другие метки.)

Можно представлять себе работу МП-машины еще и следующим образом: при создании рабочей ячейки перед обозреваемым в данный момент входным символом ставится помеченная левая скобка — меткой служит соответствующий рабочий символ, — а при уничтожении рабочей ячейки после обозреваемого входного символа ставится правая скобка. Таким способом мы получим скобочное изображение соответствующей размеченной системы составляющих — правда, с той особенностью, что одна составляющая может оказаться ограниченной несколькими парами скобок, и внутри некоторых пар скобок может не содержаться входных символов.

Лингвистически МП-машины интерпретируются как «предсказуемостные процедуры синтаксического анализа», или «предсказуемостные анализаторы». В таком анализаторе предложение («содержимое входной ленты») просматривается слева направо, и при просмотре слова (входного символа) может вырабатываться предположение («предсказание») о его синтаксической роли (это соответствует созданию рабочей ячейки) или проверяться его совместимость с последним из уже выработанных, но еще не проверенных предположений (соответствует уничтожению рабочей ячейки). Результатом работы анализатора является размеченная система составляющих (отвечающая дереву вычисления). Подробнее см. [Гладкий — Мельчук 1969, стр. 136—148.]

Отметим две особенности деревьев вычисления, отличающие их от деревьев вывода.

1) Между полными вычислениями нормальной МП-машины и их деревьями имеется очевидное взаимно однозначное соответствие.

2) В то время как для всякой Б-грамматики Г существует постоянная d такая, что ширина дерева вывода в Г никогда не превосходит d, для деревьев вычислений в МП-машине такой постоянной может не быть (упражнение 4.24).

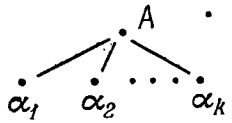
Впрочем, для любой МП-машины можно построить эквивалентную, обладающую такой постоянной; ее даже можно сделать равной двум. Это непосредственно вытекает из доказываемых ниже лемм 4.13 и 4.14.

Докажем теперь следующую основную теорему.

Теорема 4.9. а) *Для всякой* ОБ-*грамматики можно построить эквивалентную ей* МП-*машину.* б) *Для всякой* МП-*машины можно построить эквивалентную ей* ОБ-*грамматику.*

Для доказательства введем некоторые понятия и обозначения.

Назовем простейшей окрестностной Δ-грамматикой с порядком*) (ПО-грамматикой) упорядоченную четверку $G = \langle V, W, I, S\rangle$, где V и W — непересекающиеся конечные множества, I — элемент W и S — конечное множество «окрестностей» — P_1-деревьев с пометками из $V \cup W$, состоящее из «одноэтажных» деревьев вида



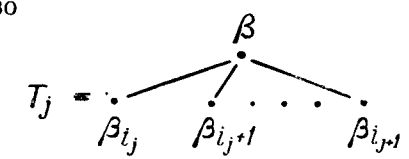
(отношение $\prec$ изображается здесь и далее отношением «левее») и «единичных» деревьев вида $\cdot\alpha$; при этом метки из V могут быть только у висячих узлов. Крайний слева висячий узел неединичного дерева может быть помечен еще дополнительным символом $\llcorner$, а крайний справа — символом $\lrcorner$. (Если висячий узел единственный, то он может нести обе дополнительные метки.)

ПО-грамматика будет называться ограниченной, если в каждой ее неодноэлементной окрестности крайние слева и справа висячие узлы несут соответственно метки $\llcorner$ и $\lrcorner$.

Будем обозначать через $L_\Delta(G)$ множество всевозможных P_1-деревьев T, удовлетворяющих следующим условиям: 1) Если s — некоторый невисячий узел T, $s_1 \prec \ldots \prec s_p$ — все подчиненные ему узлы и $\beta, \beta_1, \ldots, \beta_p$ — метки при $s, s_1, \ldots, s_p$ соответственно, то найдутся такие $i_1, \ldots, i_k$, $1 = i_1 \leqslant i_2 \leqslant \ldots \leqslant i_k = p$, что

*) Δ — от δενδρον — «дерево».

каждое дерево



принадлежит S, и при этом крайний слева висячий узел первого из этих деревьев, рассматриваемого как окрестность из S, несет добавочную метку $\llcorner$, а крайний справа висячий узел последнего — метку $\lrcorner$, и никакие другие узлы деревьев T_j меток $\llcorner$ и $\lrcorner$ не несут. 2) Если s — висячий узел T с меткой α, то дерево $\cdot\alpha$ принадлежит S. 3) Корень T несет метку I.

Будем полагать далее $L(G) = \{\text{Пр}_{V\cup}(T) \mid T \in L_\Delta(G)\}$.

Будем, наконец, обозначать множество всех деревьев полных выводов в ОБ-грамматике Γ, соответственно всех упрощенных деревьев полных вычислений нормальной МП-машины M, через $L_\Delta(\Gamma)$, соответственно $L_\Delta(M)$.

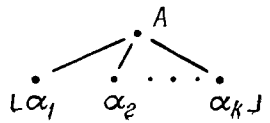
Теорема 4.9 непосредственно вытекает из следующих трех лемм.

Лемма 4.12. а) *Для всякой* ОБ-*грамматики* Γ *можно построить такую ограниченную* ПО-*грамматику* G, *что* $L_\Delta(G) = L_\Delta(\Gamma)$. б) *Для всякой ограниченной* ПО-*грамматики* G *можно построить такую* ОБ-*грамматику* Γ, *что* $L_\Delta(\Gamma) = L_\Delta(G)$.

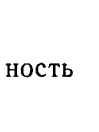
Лемма 4.13. а) *Для всякой нормальной* МП-*машины* M *можно построить такую* ПО-*грамматику* G, *что* $L_\Delta(G) = L_\Delta(M)$. б) *Для всякой* ПО-*грамматики* G *можно построить такую нормальную* МП-*машину* M, *что* $L_\Delta(M) = L_\Delta(G)$.

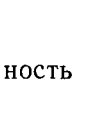
Лемма 4.14. *Для всякой* ПО-*грамматики* G *можно построить такую ограниченную* ПО-*грамматику* G', *что* $L(G') = L(G)$.

Доказательство леммы 4.12. Для произвольной ОБ-грамматики $\Gamma = \langle V, W, I, R\rangle$ положим $G(\Gamma) = \langle V, W, I, S\rangle$, где S состоит из всевозможных окрестностей вида

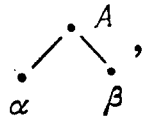


таких, что $A \to \alpha_1\alpha_2 \ldots \alpha_k \in R$, и $\bullet\alpha$ таких, что $\alpha \to \Lambda \in R$ или $\alpha \in V$. Ясно, что таким образом устанавливается взаимно однозначное и эффективное в обе стороны соответствие между ОБ-грамматиками и ограниченными ПО-грамматиками. Равенство $L_\Delta(G(\Gamma)) = L_\Delta(\Gamma)$ очевидно.

Доказательство леммы 4.13. а) Вместо машины M можно рассматривать Д-автомат M_1. Соответствующая ПО-грамматика будет иметь вид $\langle V, W, E, S\rangle$, где V и W — соответственно входной и рабочий алфавиты M, E — символ, записываемый в начале вычисления, и S строится следующим образом: (i) если головка может, записав в некоторой ячейке символ A, тотчас же спуститься этажом ниже и записать α (иначе говоря, для некоторых состояний q, q', q'' имеются инструкции $q' \to Aq$, $q \to \alpha q''$), то в S включается окрестность  ; (ii) если головка может, поднявшись из

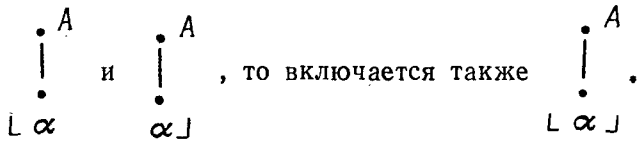
ячейки, где записано α, в ячейку, где записано A, тотчас же подняться еще на этаж (т. е. имеются инструкции $\alpha q' \to q$, $Aq \to q''$), то в S включается окрестность  ; (iii) если головка может, поднявшись на

один этаж из ячейки, где записано α, тотчас же спуститься снова и записать β (т. е. имеются инструкции $\alpha q' \to q$, $q \to \beta q''$), то в S включаются всевозможные окрестности вида



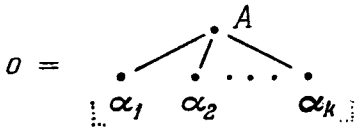
где A — произвольный рабочий символ; (iv) если головка может, записав в некоторой ячейке α, тотчас же подняться (т. е. имеются инструкции $q' \to \alpha q$, $\alpha q \to q''$), то

в S включается окрестность $\bullet\alpha$; (v) если в S включены окрестности



Равенство $L_\Delta(G) = L_\Delta(M)$ очевидно.

б) По ПО-грамматике $G = \langle V, W, I, S\rangle$ мы построим Д-автомат M_1 следующим образом. Пусть



— произвольная не одноэлементная окрестность из S ($k \geqslant 1$; символы и означают наличие или отсутствие пометок $\llcorner$ и $\lrcorner$ соответственно). Сопоставим ей $4k-1$ состояний: $l_1, \ldots, l_k$ («левые состояния»), $r_1, \ldots, r_k$ («правые состояния»), $m_1, \ldots, m_k$ («нижние состояния»), $u_1, \ldots, u_{k-1}$ («средние состояния») и инструкции: (1) $\alpha_1 r_1 \to u_1$, (2) $u_1 \to \alpha_2 l_2$, ..., $(2i-2)$ $u_{i-1} \to \alpha_i l_i$, $(2i-1)$ $\alpha_i r_i \to u_i$, ..., $(2k-3)$ $\alpha_{k-1} r_{k-1} \to u_{k-1}$, $(2k-2)$ $u_{k-1} \to \alpha_k l_k$. (Эти инструкции позволяют обходить дерево o «с перерывами»: из крайнего слева висячего узла, где ранее уже записано α_1, в корень; затем во второй слева висячий узел, причем в нем пишется α_2 и автомат оказывается в состоянии l_2; если потом головка попадает когда-нибудь в тот же узел снова и при этом автомат будет в состоянии r_2, то можно еще раз перейти в корень и оттуда в следующий висячий узел, и т. д.) Если при этом крайний слева висячий узел o несет метку $\llcorner$, то мы включаем в число сопоставляемых также всевозможные инструкции вида $l' \to \alpha_1 l_1$, где l' — левое состояние, сопоставленное произвольному висячему узлу с меткой A произвольной окрестности из S; если крайний справа висячий узел o несет метку $\lrcorner$, то добавляем

всевозможные инструкции вида $\alpha_k r_k \to r'$, где r' определяется аналогично l'. (С помощью этих инструкций начинается, соответственно завершается, обход всего куста узла дерева из $L_\Delta(G)$.) Далее, если N — множество всех тех чисел $i = 1, \ldots, k$, для которых в S имеются окрестности $\cdot\alpha_i$, то для каждого $i \in N$ мы сопоставим окрестности o еще две инструкции: $u_{i-1} \to \alpha_i m_i$, $\alpha_i m_i \to u_i$; сверх того, при $l \in N$ добавим инструкции $l' \to l_1 m_1$ и при $k \in N$ — инструкции $\alpha_k m_k \to r'$, где l' и r' имеют прежний смысл. (Эти инструкции будут выполняться, когда соответствующие узлы висячие.) Полученную систему инструкций обозначим $P(o)$. Объединение всех $P(o)$ будет программой M_1. Непосредственно ясно, что множество деревьев, которые может обходить M_1, совпадает с $L_\Delta(G)$.

Доказательство леммы 4.14. Пусть G — произвольная ПО-грамматика и G' — ограниченная ПО-грамматика. Равенство $L(G') = L(G)$ будет обеспечено, если мы установим между $L_\Delta(G)$ и $L_\Delta(G')$ взаимно однозначное соответствие со следующим свойством:

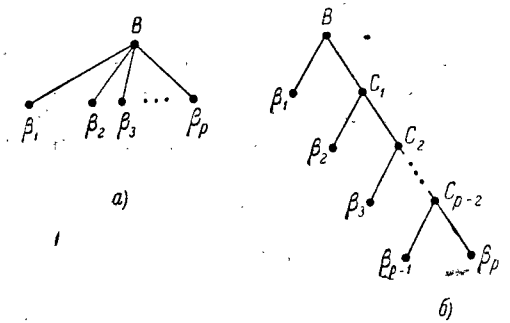
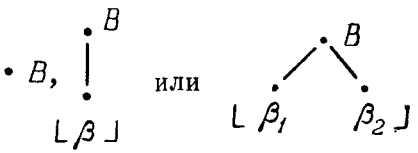


Рис. 7.

каждому кусту дерева из $L_\Delta(G)$, имеющему вид, показанный на рис. 7, а), отвечает в соответствующем дереве из $L_\Delta(G')$ поддерево рис. 7, б), где $C_1, \ldots, C_{p-2}$ — некоторые новые символы.

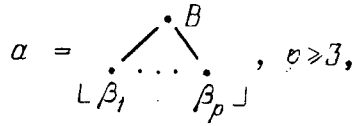
Но такое соответствие будет, как легко проверить непосредственно, иметь место, если сопоставить каждой окрестности a грамматики G систему окрестностей a' грамматики G' следующим образом:

1) если a имеет вид

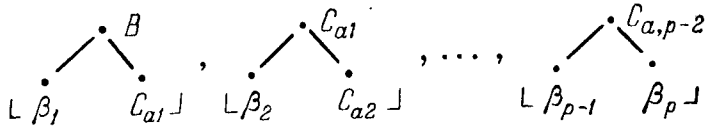


то a' состоит из одной окрестности a.

2) Если

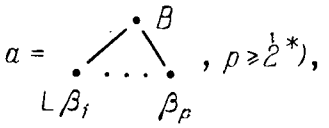


то a' состоит из окрестностей

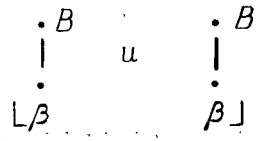


(здесь и далее C_{ai} — новые символы).

3) Если

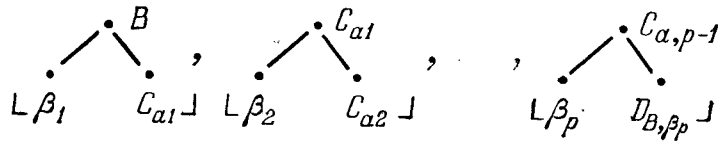


*) Из определения $L_\Delta(G)$ ясно, что окрестности вида



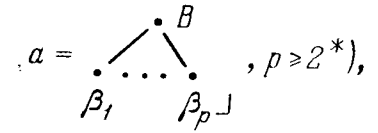
можно изъять, добавив в случае надобности некоторые новые окрестности с более чем двумя висячими вершинами.

то a' состоит из окрестностей

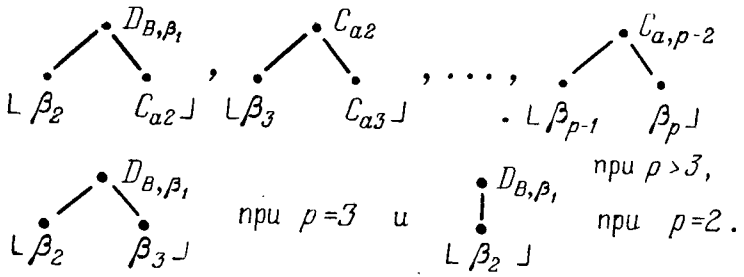


(здесь и далее D_{B,β_j} — новые символы).

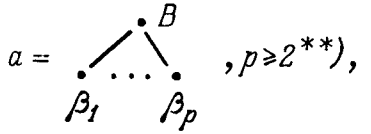
4) Если



то a' состоит из окрестностей



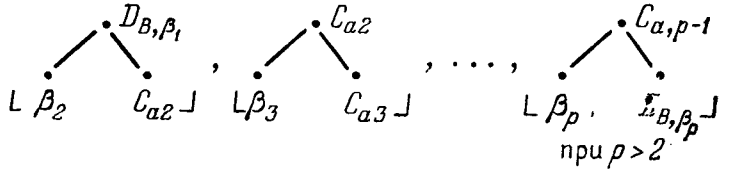
5) Если



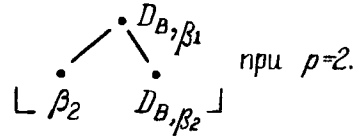
*) См. сноску на стр. 147.

**) Из определения $L_\Delta(G)$ ясно, что окрестности вида  излишни.

то a' состоит из окрестностей



и



Замечания. 1) Процедуры, примененные при доказательстве лемм 4.12, а) и 4.13, б), устанавливают, как легко убедиться, взаимно однозначное соответствие между множеством деревьев полных выводов ОБ-грамматики и множеством упрощенных деревьев полных вычислений эквивалентной ей МП-машины, и при этом соответствии сохраняются отвечающие деревьям цепочки основных символов. Аналогичный факт верен для процедур лемм 4.13, а), 4.12, б) и 4.14.

2) Пусть для произвольной ОБ-грамматики Г выражение $L_C(\Gamma)$ означает множество всевозможных систем составляющих для цепочек языка $L(\Gamma)$, соответствующих деревьям выводов этих цепочек в Г. Аналогичный смысл придадим обозначению $L_C(M)$, где M — МП-машина. Тогда доказательства лемм 4.12, а) и 4.13, б) дают способ для всякой ОБ-грамматики Г построить такую МП-машину M, что $L_C(M) = L_C(\Gamma)$. Обратное, очевидно, неверно: если ширина множества $L_C(M)$ не ограничена (т. е. не существует числа, мажорирующего ширину любой системы из $L_C(M)$), то ни для какой ОБ-грамматики Г равенство $L_C(\Gamma) = L_C(M)$ невозможно. Но если ширина множества $L_C(M)$ ограничена и ограничивающее ее число известно, то ОБ-грамматика Г, удовлетворяющая указанному равенству, может быть построена. Для этого придется несколько усложнить конструкцию пункта а) леммы 4.13, моделируя не пары соседних дуг дерева вычисления, а целые кусты — это

становится возможным ввиду ограниченности числа дуг в них; тогда отпадет надобность в лемме 4.14, т. е. в изменении топологии дерева.

Подробное проведение соответствующих построений предоставляется читателю. Вместе с результатом упражнения 4.28 это позволяет для каждой МП-машины M либо построить ОБ-грамматику Г такую, что $L_C(\Gamma) = L_C(M)$, либо установить, что такой грамматики нет.

3) Пусть C_x — система составляющих цепочки x, отвечающая некоторому дереву вычисления МП-машины M, и C'_x — система составляющих той же цепочки, отвечающая соответствующему дереву вывода в Б-грамматике, построенной по M с помощью процедур лемм 4.13, а), 4.14 и 4.12, б). Если y — произвольная неточечная составляющая системы C_x (точнее, соответствующая подцепочка цепочки x) и если $y = z_1 z_2 \ldots z_s$, где $z_1, z_2, \ldots, z_s$ — составляющие, непосредственно вложенные в y, то все отрезки $y = z_1 z_2 \ldots z_s$, $z_2 \ldots z_s$, $\ldots$, $\ldots, z_{s-1} z_s$ принадлежат C'_x. При этом все неточечные составляющие системы C'_x могут быть получены таким способом. Сверх того, метки новых составляющих вполне определяются метками непосредственно содержащих их старых.

Естественно возникает вопрос о единственности дерева вычисления для данной входной цепочки. По аналогии с понятием однозначной Б-грамматики напрашивается следующее определение: МП-машина M называется **однозначной**, если для каждой цепочки $x \in L(M)$ существует единственное дерево полного x-вычисления (или, что то же самое, единственное полное x-вычисление).

Машины примеров 1 и 2 на стр. 137—138, как легко видеть, однозначны, а машины примера 3 — нет (упражнение 4.29).

Что касается понятия однозначного ОБ-языка, то оно не изменится, если определить его с помощью МП-машины: имеет место

Теорема 4.10. *ОБ-язык тогда и только тогда однозначен, когда существует допускающая его однозначная* МП-*машина.*

Доказательство немедленно получается из замечания 1) к леммам 4.12, 4.13 и 4.14.

В заключение нам остается уделить немного внимания детерминированным МП-машинам — вернее, тем Б-языкам, которые такими машинами допускаются.

Ясно, прежде всего, что всякая детерминированная МП-машина однозначна, так что неоднозначные Б-языки не допускаются детерминированными МП-машинами. Но и не все однозначные Б-языки допускаются ими. Чтобы в этом убедиться, докажем следующую лемму.

Лемма 4.15. *Пусть M — детерминированная* МП-*машина такая, что $L(M) - \{\Lambda\}$ не является* А-*языком. Тогда найдется цепочка $w \in L(M)$, представимая в виде $w = z_1 y_1 x y_2 z_2$ так, что* (а) $y_1 \neq \Lambda$, $y_2 \neq \Lambda$; (б) *какова бы ни была цепочка u такая, что $v = wu \in L(M)$, цепочка $v_n = z_1 y_1^n x y_2^n z_2 u$ также принадлежит $L(M)$.*

Доказательство. Пусть сначала w и u — произвольные непустые цепочки такие, что w, $wu \in L(M)$. Ввиду детерминированности машины wu-вычисление, оборванное после чтения последнего символа из w, будет совпадать с w-вычислением, оборванным после чтения того же символа. При «скобочной» интерпретации вычисления (стр. 141) это означает, что во всяком случае, внутри w при допускании цепочки wu будут поставлены в точности те же скобки — и с теми же метками, — что и при допускании w *).

Построим теперь по машине M Б-грамматику Г с помощью процедур лемм 4.13, а), 4.14 и 4.12, б). Если C_x — система составляющих цепочки x, принадлежащая $L_C(M)$, и C'_x — соответствующая система составляющих из $L_C(\Gamma)$, то по предыдущему всякая составляющая из C_w, расположенная целиком внутри w, принадлежит и системе C_{wu} и имеет в ней в точности те же метки; по замечанию 3) на стр. 150 это верно и для систем C'_w и C'_{wu}. В силу следствия из теоремы 7.6 **) грамматика Г имеет неограниченную степень самовставления, и во всяком случае найдется такая цепочка $w \in L(\Gamma) = L(M)$, что $\bar{X}_\Gamma(w) \geqslant 3$, — иначе говоря, w представима в виде $z_1 s_1 t_1 r t_2 s_2 z_2$, где $s_1, s_2, t_1, t_2 \neq \Lambda$ и отрезки

*) Скобки, поставленные при допускании цепочки w на ее правом конце, могут при допускании wu оказаться внутри u.

**) Ознакомившись с доказательством этой теоремы, читатель убедится, что от настоящей леммы оно не зависит.

r, t_1rt_2, $s_1t_1rt_2s_2$ принадлежат C_w и «происходят» от узлов с одинаковыми метками. По предыдущему отрезки r и t_1rt_2 должны принадлежать и C_{wu} и происходить в дереве вывода цепочки wu от узлов с одинаковыми метками, какова бы ни была цепочка u такая, что $wu \in L(M)$. Поэтому (ср. доказательство теоремы 4.5) для любого n имеем $z_1s_1t_1^nrt_2^ns_2z_2u \in L(M)$.

С помощью леммы 4.15 легко доказать, что язык примера 2 на стр. 137—138 при $n > 1$ не допускается никакой детерминированной МП-машиной. Действительно, в противном случае для некоторой $w \in \{a_1, \ldots, a_n\}^*$ можно было бы цепочку $w\hat{w}$ представить в виде $w\hat{w} = z_1y_1xy_2z_2$ так, чтобы было $y_1 \neq \Lambda$, $y_2 \neq \Lambda$ и для любой цепочки $u \in \{a_1, \ldots, a_n\}^*$ имело бы место $z_1y_1^nxy_2^nz_2u\hat{u}w\hat{w} \in L$. Но если положить $u = a_i^{|y_1|+|y_2|}$, где a_i отличен от последнего символа цепочки w, то в цепочке $z_1y_1^2xy_2^2z_2u\hat{u}w\hat{w}$ на расстоянии $2|w|+|y_1|+|y_2|$ от левого и правого концов будут стоять разные символы.

[Стоит заметить, что «очень похожий» язык $\{xb\hat{x} \mid x \in \{a_1, \ldots, a_n\}^*\}$ допускается детерминированной МП-машиной, — например, с программой $\{q_1\# \to q_2$; $q_2a_i \to q_{i+3}$; $q_2b \to q_3$; $q_{i+3} \to A_iq_2$; $q_3a_i \to q_{i+n+3}$; $q_3\# \to q_0$; $A_iq_{i+n+3} \to q_3\}$ (q_0 — единственное заключительное состояние). Символ b служит здесь сигналом к изменению направления движения рабочей головки, тогда как в предыдущем примере менять направление приходилось наугад, и поэтому в любой момент нужно было иметь возможность это сделать.]

Дальнейшие примеры см. в упражнении 4.34.

Упражнения

4.1. а) Показать, что для всякой Б-грамматики Г можно построить эквивалентную ей Б-грамматику Г′, каждое правило которой либо имеет вид $l \to a$, где l — начальный символ и a — основной символ, либо длина его правой части больше единицы.

Замечание. Очевидно, $T_{\Gamma'}(n) \leqslant n$. Поэтому $\mathscr{L}_n^T(\text{Б}) = \mathscr{L}(\text{Б})$. Тем более $\mathscr{L}(\text{Б}) \subseteq \mathscr{L}_n^T(\text{НС})$.

б) Показать, что для произвольной Б-грамматики Г функция $T_\Gamma(n)$ мажорируется линейной функцией.

4.2. Показать, что если в теореме 2.2 Г есть Б-грамматика, то и Г′ можно сделать Б-грамматикой.

4.3. Оценить изменение временной сложности в конструкции леммы 4.3.

4.4. Показать, что:

а) если два вывода в Б-грамматике имеют одно и то же дерево, то они получаются друг из друга перестройкой;

б) обратное неверно (указать пример).

4.5. Указать пример НС-грамматики, в которой вывод может иметь несколько разных деревьев.

4.6. Показать, что для любой Б-грамматики можно построить эквивалентную ей Б-грамматику без правил вида $A \to B$ (A, B — вспомогательные символы), в которой каждый вспомогательный символ, кроме, быть может, начального, циклический.

4.7. Указать алгоритмы, позволяющие по любой Б-грамматике Г с данным основным словарем V и любой цепочке $x \in V^*$ распознать, выводима ли в Г цепочка:

а) содержащая вхождение x;

б) начинающаяся с x;

в) оканчивающаяся x.

[Bar-Hillel — Perles — Shamir 1961.]

4.8. Распространить на ОБ-грамматики леммы 4.1—4.10 и теоремы 4.1, 4.2.

4.9. Показать, что языки из упражнений 3.6, 3.13, 3.20, а) не являются бесконтекстными (язык из упражнения 3.6, б) — только при $k > 2$, язык из упражнения 3.6, г) — только при $k > 1$).

4.10. Показать, что коммутативное замыкание (см. упражнение 3.7) Б-языка может не быть Б-языком.

4.11. Пользуясь теоремой 4.7, показать, что языки

$$\{a^nb^na^m \mid m,\ n = 1, 2, \ldots;\ m \geqslant n\}$$

и

$$\{a^nb^na^m \mid m,\ n = 1, 2, \ldots;\ m \neq n\}$$

не являются бесконтекстными.

4.12. Показать, что класс ОБ-языков эффективно замкнут относительно операций левого и правого деления на цепочку.

4.13. Доказать незамкнутость класса Б-языков относительно вычитания и взятия дополнения непосредственно (построив соответствующий пример).

4.14. Проанализировав доказательство леммы 4.3, убедиться, что Б-язык тогда и только тогда является однозначным, когда он может быть порожден однозначной стандартной бинарной Б-грамматикой.

4.15. Замкнут ли класс однозначных Б-языков относительно объединения, пересечения, усеченной итерации?

4.16. Доказать неоднозначность следующих Б-языков:

а) $\{a^mb^mc^nd^n \mid m,\ n = 1, 2, \ldots\} \cup \{a^mb^nc^nd^m \mid m,\ n = 1, 2, \ldots\}$;

б) $\{a^kb^mc^n \mid k,\ m,\ n = 1, 2, \ldots;\ n \leqslant k \vee n \leqslant m\}$;

в) $\{a^kb^kc^md^me^n \mid k,\ m,\ n = 1, 2, \ldots\} \cup \{a^kb^mc^md^ne^n \mid k, m, n = 1, 2, \ldots\}$.

4.17. Замкнут ли класс неоднозначных Б-языков относительно объединения, умножения, усеченной итерации?

4.18. а) Показать, что произведение неоднозначного Б-языка и двухэлементного языка может быть однозначным Б-языком.

б) Верен ли аналогичный факт для одноэлементного языка?

4.19. Построить Б-грамматику, порождающую язык $L = a^*b^*c^*d^*e^* - L'$, где L' — язык из упражнения 4.16, в) [Ullian 1966]. [Указание. Представить L в виде $L_1 \cup L_2 \cup L_3$, где $L_i = \{a^g b^h c^i d^j e^l \mid \mathfrak{A}_i\}$, причем $\mathfrak{A}_1$ означает $h \neq i \,\&\, g + i \neq h + j$, $\mathfrak{A}_2$ означает $i \neq j \,\&\, h + j \neq i + l$, $\mathfrak{A}_3$ означает $g \neq h \,\&\, j \neq l$.]

4.20. Положим $L_{s,k} = \{a^n b a^{i_1} b \ldots b a^{i_{k-1}} b a^n b a^{i_{k+1}} b \ldots b a^{i_s} \mid n, i_j = 1, 2, \ldots\}$, $L_s = L_{s1} \cup L_{s2} \cup \ldots \cup L_{ss}$, $L = L_1 \cup L_2, \ldots$ ($s, k = 2, 3, \ldots$; $k \leqslant s$). Показать, что:

а) для любого натурального числа s в L_s найдется цепочка, имеющая в любой Б-грамматике, порождающей L_s, не менее s различных деревьев вывода;

б) для любого натурального числа s в L найдется цепочка, имеющая в любой Б-грамматике, порождающей L, не менее s различных деревьев вывода.

Замечание. Языки L_s и L не только бесконтекстны, но и линейны (см. ниже, § 5.2).

4.21. а) Показать, что при любом $n = 2, 3, \ldots$ язык $M_n = \bigcup_{i=1}^{n} (a^i b)^*$ порождается Б-грамматикой с $n + 1$ вспомогательными символами и не порождается никакой Б-грамматикой с меньшим числом вспомогательных символов.

Замечание. Легко видеть, что все языки M_n автоматные.

б) Указать пример А-языка, порождаемого Б-грамматикой с двумя вспомогательными символами и не порождаемого никакой Б-грамматикой с одним вспомогательным символом.

[Gruska 1967].

4.22. Построить МП-машины, допускающие языки примеров 6, 7, 9, б) из § 1.3 и языки из упражнения 1.15.

4.23. Построить деревья вычислений цепочки a^4b^4 в машине примера 1 из § 4.5 и цепочки $a_1a_2^2a_3a_3a_2^2a_1$ в машине примера 2 из § 4.5.

4.24. Показать, что для машины примера 3 из § 4.5 ширина деревьев вычисления не ограничена.

4.25. Дополним МП-машину M третьей лентой — назовем ее выходной — со своей головкой и своим алфавитом и добавим к ее программе инструкции вида $q_\alpha \to q_\beta a$, где a — выходной символ; выполнение такой инструкции будем интерпретировать как запись символа a. Полученное устройство M' назовем МП-машиной с выходом.

Если M' сможет, начав работу, когда M находится в начальной x-ситуации и выходная лента пуста, закончить ее, когда M находится в заключительной x-ситуации и на выходной ленте записана цепочка y, то мы скажем, что M' переводит x в y, и будем писать $y = M'(x)$. Положим: $M'(L) = \{y \mid \exists x [x \in L \,\&\, y = M'(x)]\}$ $(L \subseteq V^*)$; $L_0(M') = M'(V^*)$.

Показать, что:

а) для любой МП-машины M можно построить такую МП-машину с выходом M', что $L_0(M') = L(M)$;

б) для любой МП-машины с выходом M' можно построить такую МП-машину M, что $L(M) = L_0(M')$;

в) для любой Э-машины M можно построить такую Б-грамматику Γ и такую МП-машину с выходом M_1, что $M_1(L(\Gamma)) = L(M)$.

4.26. Показать, что для любой МП-машины можно построить эквивалентную ей нормальную МП-машину, у которой в любом дереве вычисления непустой цепочки все висячие узлы помечены входными символами (так что в скобочном изображении соответствующей системы составляющих — см. стр. 141 — не будет «пустых» пар скобок). Доказать это двумя способами: с помощью теоремы 4.3 (и анализа доказательств лемм 4.12—4.14) и непосредственно (без обращения к грамматикам).

4.27. Доказать теорему 4.8, а) и результат упражнения 4.12 с помощью МП-машин (т. е. указать способ построения по МП-машинам M_1 и M_2 МП-машины, допускающей язык $L(M_1) \cup L(M_2)$, и т. д.).

4.28. Указать алгоритм, позволяющий для произвольной МП-машины M распознать, ограничена ли ширина множества $L_C(M)$, и в случае положительного ответа найти ее верхнюю границу.

4.29. а) Найти для цепочки *aabbab* еще одно дерево вычисления в машине примера 3 из § 4.5, кроме изображенного на рис. 6.

б) Убедиться, что для любого натурального n найдется допускаемая этой машиной цепочка, имеющая больше n вычислений.

4.30. Показать, что язык $\{a^m b^n a^n b^m \mid m, n = 0, 1, \ldots\}$ не допускается никакой МП-машиной, рабочий алфавит которой состоит из одного символа. (Такие машины иногда называют счетчиковыми.)

4.31. Построить детерминированные МП-машины, допускающие:

а) язык примера 7 из § 1.3;

б) «язык бинарных скобочных последовательностей» (порождаемый грамматикой со схемой $\{I \to (II), I \to (\ )\}$);

в) язык примера 3 из § 4.5;

г) язык $\{x_1 b y b \hat{y} x_2 b \hat{x}_2 \hat{x}_1 \mid x_1, x_2, y \in \{a_1, \ldots, a_n\}^*, b \notin \{a_1, \ldots, a_n\}\}$.

4.32. Показать, что для всякой детерминированной МП-машины можно построить эквивалентную ей детерминированную МП-машину «без зацикливания», т. е. такую, что для любой входной цепочки x этой машины ее (единственное) x-вычисление заканчивается [Schützenberger 1963, Ginsburg 1966].

4.33. Показать, что класс языков, допускаемых детерминированными МП-машинами, эффективно замкнут относительно: (а) умножения, (б) итерации, (в) подстановки, (г) левого и правого деления на цепочку, (д) взятия дополнения [Ginsburg — Greibach 1966 (b), Ginsburg 1966]. [*Указание.* Для (д) использовать упражнение 4.32.]

Замечание. Из (д) немедленно следует, что для всякой детерминированной МП-машины M можно построить детерминированную МП-машину, распознающую язык $L(M)$.

4.34. Показать, что следующие языки не допускаются детерминированными МП-машинами:

а) $L^2, L^3, \ldots$, где $L = \{x\hat{x} \mid x \in \{a_1, \ldots, a_n\}^*\}$ $(n > 1)$;

б) L^*, где L — то же, что в а);

в) $\{x_1 x_2 b \hat{x}_2 y b \hat{y} b \hat{x}_1 \mid x_1, x_2, y \in \{a_1, \ldots, a_n\}^*\}$ $(n > 1)$.

Замечание. Сопоставляя упражнения 4.31, г) и 4.34, в), видим, что класс языков, допускаемых детерминированными МП-машинами, не замкнут относительно обращения *).

4.35. Показать, что если U и V — непересекающиеся алфавиты, $L \subseteq U^*$ и $y \in V^*$, то:

а) по детерминированной МП-машине, допускающей язык Ly, можно построить детерминированную МП-машину, допускающую L;

б) по однозначной Б-грамматике, порождающей Ly, можно построить однозначную Б-грамматику, порождающую L.

4.36. Показать, что для всякой грамматики, левые части правил которой не содержат основных символов, а правая часть каждого правила содержит хотя бы одно вхождение основного символа, можно построить эквивалентную ей Б-грамматику [Ginsburg — Greibach 1966(a)].

*) Обращением называется операция, сопоставляющая языку L язык $\hat{L} = \{\hat{x} \mid x \in L\}$ (а также ее результат).

ГЛАВА 5

НЕКОТОРЫЕ СПЕЦИАЛЬНЫЕ КЛАССЫ БЕСКОНТЕКСТНЫХ ЯЗЫКОВ

§ 5.1. Автоматные и обобщенные автоматные языки. Конечные автоматы

В предыдущей главе мы видели, что присоединение к Б-грамматикам правил вида $A \to \Lambda$ делает класс порождаемых этими грамматиками языков в ряде отношений более естественным (достаточно вспомнить об удобстве описания с помощью МП-машин). В еще большей степени это справедливо, как мы скоро убедимся, для А-грамматик. Поэтому их изучение мы начнем со следующего определения.

(ОБ-)грамматика называется обобщенной автоматной грамматикой (ОА-грамматикой), если каждое ее правило имеет вид $A \to aB$, $A \to a$ или $A \to \Lambda$, где A, B — вспомогательные символы и a — основной символ. Язык, порождаемый ОА-грамматикой, называется ОА-языком.

Полезно заметить, что в ОА-грамматике всякая промежуточная цепочка полного вывода имеет вид xB, где x — цепочка в основном словаре и B — вспомогательный символ.

Теорема 5.1. *Для любой* ОА-*грамматики* Γ *можно построить А-грамматику* Γ' *такую, что* $L(\Gamma') = L(\Gamma) - \{\Lambda\}$.

Доказательство. С помощью процедуры, примененной в доказательстве леммы 4.2, можно перестроить Γ так, чтобы начальный символ не встречался в правых частях правил. После этого достаточно для каждой пары правил вида $A \to aB$, $B \to \Lambda$ добавить к схеме

грамматики правило $A \to a$ (если оно не содержалось там ранее) и затем изъять все правила вида $A \to \Lambda$.

Будем называть ОА-грамматику стандартной, если она не имеет правил вида $A \to a$. Для каждой ОА-грамматики легко построить эквивалентную ей стандартную ОА-грамматику: достаточно добавить к вспомогательному словарю новый символ K, к схеме — правило $K \to \Lambda$ и заменить каждое правило $A \to a$ правилом $A \to aK$.

Стандартные ОА-грамматики допускают следующее очень простое и наглядное изображение. Рассмотрим мультиграф, узлами которого служат вспомогательные символы данной грамматики и для каждой пары узлов A, B из A в B идет столько дуг, сколько грамматика имеет правил вида $A \to aB$; каждую из этих дуг пометим соответствующим основным символом a. Начальный символ будем называть начальным узлом, а каждый узел A, для которого имеется правило $A \to \Lambda$, — заключительным. Построенный таким образом мультиграф с помеченными дугами и выделенными в множестве узлов начальным и заключительными узлами мы будем называть диаграммой данной грамматики.

Нестандартной ОА-грамматике мы также будем сопоставлять диаграмму, совпадающую, по определению,

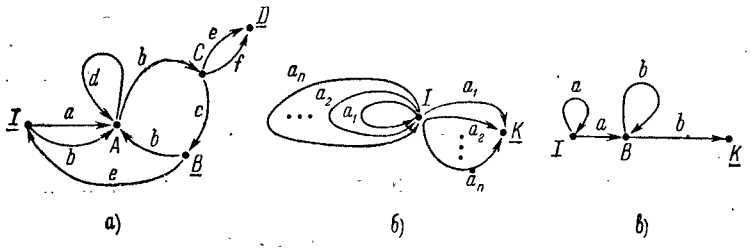


Рис. 8.

с диаграммой стандартной грамматики, полученной из данной указанным выше способом.

На рис. 8, *а*) показана диаграмма стандартной ОА-грамматики со схемой $\{I \to aA,\ I \to bA,\ A \to dA,\ A \to bC,\ C \to cB,\ B \to bA,\ C \to eD,\ C \to fD,\ B \to eI,\ I \to \Lambda,\ B \to \Lambda,\ D \to \Lambda\}$. На рис. 8, *б*), *в*) показаны диаграммы А-грамматик примеров 2 и 4 из § 1.3. Здесь и далее начальный узел обозначается символом I, заключительные узлы выделяются подчеркиванием.

Если $A_0, A_1, \ldots, A_n$ — путь в диаграмме грамматики Г и для каждого $i = 1, \ldots, n$ имеется дуга, идущая из A_{i-1} в A_i и помеченная символом a, мы будем говорить, что данный путь производит цепочку $a_1 \ldots a_n$. Всякий путь длины 0 производит, по определению, цепочку Λ. Цепочка x будет, очевидно, производиться путем, начинающимся в A и оканчивающимся в B, тогда и только тогда, когда $A \underset{\Gamma}{\models} xB$ ($\models$ означает здесь то же, что на стр. 124).

Будем называть путь $A_0, \ldots, A_n$ циклом, если $A_0 = A_n$, и полным путем, если A_0 — начальный узел и A_n — заключительный узел. Между полными выводами в Г и полными путями диаграммы имеется очевидное взаимно однозначное соответствие, и цепочка принадлежит $L(\Gamma)$ тогда и только тогда, когда она производится некоторым полным путем.

Диаграммы позволяют, в частности, особенно просто интерпретировать для ОА-грамматик понятия и утверждения из § 4.2. Так, зависимость B от A означает наличие пути из A и B, бесплодность A — отсутствие пути из A в заключительный узел, неустранимость A — наличие проходящего через A полного пути, цикличность A — наличие проходящего через A цикла. Выбрасывание всех устранимых узлов сразу дает теперь приведенную грамматику; приведенная грамматика порождает конечный язык тогда и только тогда, когда ее диаграмма не содержит циклов.

Заметим еще, что $\Lambda \in L(\Gamma)$ тогда и только тогда, когда начальная вершина диаграммы является заключительной.

Перейдем теперь к характеристике ОА-языков в терминах машин Тьюринга. Рассмотрим наиболее простой из мыслимых нетривиальных частных случаев Э-машины, а именно машину, ничего не делающую на рабочей ленте, иначе говоря, имеющую инструкции только вида (i) (стр. 42). Такую Э-машину мы будем называть конечным автоматом. Можно представлять себе

конечный автомат как Э-машину (если угодно, МП-машину) без рабочей ленты; поэтому мы будем говорить о ленте и головке конечного автомата, подразумевая входную ленту и входную головку.

Конечному автомату можно сопоставить диаграмму аналогично тому, как это было сделано для ОА-грамматик. Именно: узлами диаграммы будут состояния; из узла q_α в узел q_β будет идти дуга, помеченная символом a, если $a \neq \#, \dashv$ и автомат имеет инструкцию $q_\alpha a \to q_\beta$; узлы q_α, для которых имеются инструкции вида $q_1 \# \to q_\alpha$, будут называться начальными, узлы, принадлежащие Q_0, — заключительными.

Пример диаграммы конечного автомата представлен на рис. 9; начальные узлы обведены кружками, заключительный (единственный) выделен подчеркиванием.

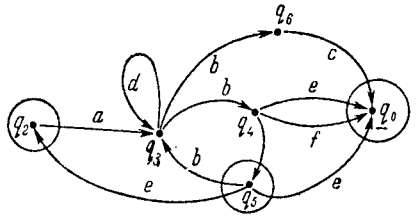


Рис. 9.

Цепочка, производимая путем, а также цикл и полный путь определяются для диаграммы автомата точно так же, как для диаграммы грамматики. Непосредственно ясно, что цепочка x производится путем, начинающимся в q_α и оканчивающимся в q_β, тогда и только тогда, когда для любой цепочки $z \in \# V^*$ и подходящего символа $b \in V \cup \{\dashv\}$, где V — (входной) алфавит автомата, из ситуации (q_α, z, xb)* достижима ситуация (q_β, zx, b) *). Поэтому цепочка x производится полным путем диаграммы конечного автомата M тогда и только тогда, когда $x \in L(M)$. Автомат допускает пустую цепочку тогда и только тогда, когда пересечение множеств начальных и заключительных узлов не пусто.

*) В обозначениях ситуаций опущены компоненты, относящиеся к рабочей ленте.

Различие между диаграммами грамматики и автомата сводится к тому, что первая имеет единственный начальный узел, чего не требуется от второй. Поэтому для всякой ОА-грамматики можно построить эквивалентный ей конечный автомат. Однако верно и обратное, поскольку для всякого конечного автомата можно построить эквивалентный ему конечный автомат с единственным начальным узлом в диаграмме: достаточно ввести новое состояние q' и новую инструкцию $q_1 \# \to q'$, изъять все старые инструкции вида $q_1 \# \to q_\alpha$ и добавить всевозможные инструкции вида $q' a \to q_\beta$, где a и q_β таковы, что старая программа содержала для некоторого q_α инструкции $q_1 \# \to q_\alpha$, $q_\alpha a \to q_\beta$ (попросту говоря, нужно добавить к диаграмме новый узел и провести из него дуги во все те узлы, в которые идут дуги из прежних начальных узлов, пометив эти дуги теми же символами); кроме того, если хотя бы один из прежних начальных узлов заключительный, то нужно объявить заключительным и новый начальный узел. Итак, доказана

Теорема 5.2. а) *Для всякой ОА-грамматики можно построить эквивалентный ей конечный автомат.* б) *Для всякого конечного автомата можно построить эквивалентную ему ОА-грамматику.*

Теперь естественно заняться вопросом о взаимоотношении между детерминированными и недетерминированными конечными автоматами. Ситуация здесь существенно отлична от той, которую мы наблюдали для МП-машин: имеет место

Теорема 5.3. *Для всякого конечного автомата можно построить эквивалентный ему детерминированный конечный автомат.*

Доказательство. Пусть M — конечный автомат с множеством состояний Q, начальным состоянием q_1, множеством заключительных состояний Q_0 и программой P.

Построим новый конечный автомат M' следующим образом: а) состояниями M' будут всевозможные подмножества Q; б) начальным состоянием будет $\{q_1\}$; в) заключительными состояниями будут те подмножества Q, которые имеют непустые пересечения с Q_0; г) программа будет состоять из всевозможных инструкций вида

$Q_1 a \to Q_2$, где $Q_1 \subseteq Q$ и $Q_2 = \{q_\beta | (\exists q_\alpha \in Q_1)[q_\alpha a \to q_\beta \in P]\}$. Поскольку Q_1 и a однозначно определяют Q_2, автомат M' детерминирован. Эквивалентность M' и M усматривается без труда.

§ 5.2. Операции над ОА-языками. Регулярные языки

Теорема 5.4. *Класс* ОА-*языков эффективно замкнут относительно операций объединения, пересечения, вычитания, взятия дополнения, умножения, итерации, усеченной итерации, подстановки, левого и правого деления на цепочку.*

Доказательство. Пусть Γ, Γ_1, Γ_2 — ОА-грамматики и D, D_1, D_2 — их диаграммы. Мы можем считать, что в каждой диаграмме в начальный узел не входит ни одна дуга. (Это достигается применением конструкции

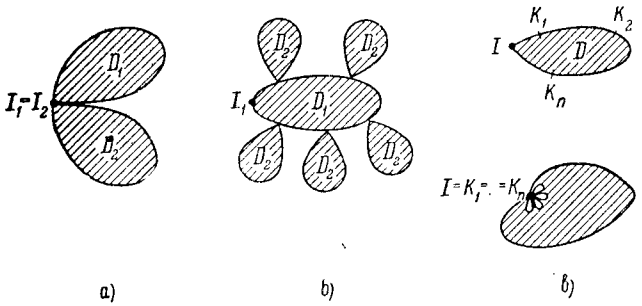


Рис. 10.

теоремы 5.1 с последующим построением эквивалентной стандартной ОА-грамматики, как указано на стр. 158.) Допустим, кроме того, что множества узлов D_1 и D_2 не пересекаются. Тогда диаграмма ОА-грамматики, порождающей $L(\Gamma_1) \cup L(\Gamma_2)$, получается из объединения диаграмм D_1 и D_2 отождествлением («склеиванием») их начальных узлов, причем заключительными узлами новой диаграммы будут все заключительные узлы D_1 и D_2 (рис. 10, *а*)). Диаграмма ОА-грамматики, порождающей $L(\Gamma_1)L(\Gamma_2)$, получается следующим образом: берется столько разных экземпляров D (попарно непересекающимися множествами узлов), сколько заключительных узлов имеет D_1, и каждый заключительный узел D_1

«склеивается» с начальным узлом одного из экземпляров D_2; начальным узлом новой диаграммы будет начальный узел D_1, заключительными — все заключительные узлы всех экземпляров D_2 (рис. 10, *б*)). Чтобы получить диаграмму ОА-грамматики, порождающей $L(\Gamma)^*$, нужно в D «склеить» вместе начальный и все заключительные узлы (рис. 10, *в*)).

Чтобы построить ОА-грамматику, порождающую $CL(\Gamma)$, построим сначала детерминированный конечный автомат M, допускающий $L(\Gamma)$ (теоремы 5.2 и 5.3). Пусть D' — его диаграмма. D' имеет единственный начальный узел (будучи детерминированным, M может иметь только одну инструкцию вида $q_1 \# \to q_\alpha$). Построим новую диаграмму D'' следующим образом. Все узлы D' будут узлами D'', все дуги D' — дугами D'' (с теми же метками). Кроме того, D'' будет содержать еще один узел q_0', и из каждого узла $q \neq q_0'$ будет идти в q_0' дуга с меткой a тогда и только тогда, когда в D' нет дуги с этой меткой, выходящей из q. Кроме того, для каждого основного символа a в D'' будет дуга из q_0' в q_0' («петля») с меткой a. Других узлов и дуг в D'' не будет. Единственным начальным узлом D'' будет начальный узел D', заключительными узлами D'' будут q_0' и все незаключительные узлы D'. Нетрудно видеть, что цепочка x в основном словаре производится каким-либо полным путем в D'' тогда и только тогда, когда она не производится никаким полным путем в D'. Действительно, это очевидно, если x производится некоторым путем в D', исходящим из начального узла q_1, — ведь такой путь единственен, а диаграмма D'' определена так, что и в ней нет другого пути с началом q_1, производящего x. Если же x не производится никаким путем в D' с началом q_1 и z — наибольшее начало x, производимое некоторым путем $q_1 = r_0, \ldots, r_s$ с началом q_1 в D', то полный путь $r_0, \ldots, r_s, \underbrace{q_0', \ldots, q_0'}_{|x|-s \text{ раз}}$ в D'' будет производить x.

Поскольку D'' — диаграмма некоторой ОА-грамматики Γ'', имеем $L(\Gamma'') = CL(\Gamma)$.

Эффективная замкнутость класса ОА-грамматик относительно пересечения и вычитания следует из уже

полученных результатов для объединения и взятия дополнения.

Диаграмма ОА-грамматики, порождающей $x\backslash L(\Gamma)$, получается из диаграммы D', построившейся в процессе доказательства для $CL(\Gamma)$, если в ней объявить начальным узлом конец пути, начинающегося в прежнем начальном узле и производящего x. (Если такой путь существует, то он единственен, а если его нет, то язык $x\backslash L(\Gamma)$ пуст. Какой из этих двух случаев имеет место, легко узнать по D'.) Аналогично для $L(\Gamma)/x$.

Доказательство для усеченной итерации и подстановки предоставляется читателю.

Замечания. 1) Конструкции, использованные при доказательстве теоремы 5.4 для объединения, умножения и итерации, применимы, очевидно, к произвольным Э-машинам (нужно только вместо узлов диаграммы говорить о состояниях). Мы будем называть отвечающие этим конструкциям операции — и их результаты — параллельной композицией, последовательной композицией и итерацией соответственно. Ясно, что параллельная композиция M и M', последовательная композиция M и M' и итерация M допускают соответственно языки $L(M) \cup L(M')$, $L(M)L(M')$, $L(M)^*$. Очевидным образом определяются также параллельная и последовательная композиции любого конечного числа Э-машин.

2) Из доказательства теоремы 5.4 для левого деления ясно, что ОА-язык может иметь лишь конечное число различных левых частных. То же верно, разумеется, и для правых.

Отметим еще следующий полезный факт.

Теорема 5.5. *Для любой* МП-*машины* M_1 *и любого конечного автомата* M_2 *можно построить* МП-*машину* M *такую, что* $L(M) = L(M_1) \cap L(M_2)$. *Если при этом* M_1 *и* M_2 *детерминированные, то и* M *можно сделать детерминированной.*

Доказательство. Построим МП-машину M следующим образом. Состояниями M будут всевозможные упорядоченные пары (q, q'), где q — состояние M_1 и q' — состояние M_2. Инструкция $(q_\alpha, q'_\beta)a \to (q_\gamma, q'_\delta)$ будет принадлежать программе M тогда и только тогда, когда программа M_1 содержит инструкцию $q_\alpha a \to q_\gamma$,

а программа M_2 — инструкцию $q'_\beta a \to q'_\gamma$. Инструкциями типов (ii) и (iii) будут всевозможные инструкции вида $A(q_\alpha, q') \to (q_\beta, q')$ и $(q_\alpha, q') \to A(q_\beta, q')$, где $Aq_\alpha \to q_\beta$, соответственно $q_\alpha \to Aq_\beta$, — инструкции M_1 и q' — произвольное состояние M_2. Начальным состоянием будет (q_0, q'_0), где q_0, q'_0 — начальные состояния M_1 и M_2 соответственно, заключительными состояниями — всевозможные пары (q_f, q'_f), где q_f, q'_f — заключительные состояния M_1 и M_2 соответственно. Ясно, что так построенная машина M будет искомой.

Следствие. *Для любой* ОБ (Б)-*грамматики* Γ_1 *и любой* ОА (А)-*грамматики* Γ_2 *можно построить* ОБ (Б)-*грамматику* Γ *такую, что* $L(\Gamma) = L(\Gamma_1) \cap L(\Gamma_2)$.

Теперь мы в состоянии получить еще одну весьма важную характеристику класса ОА-языков; эта характеристика дает способ их задания, часто оказывающийся гораздо более удобным, чем ОА-грамматики и конечные автоматы.

Пусть $V = \{a_1, \ldots, a_n\}$ — произвольный словарь и $\mathfrak{A}(\xi_1, \ldots, \xi_n, \xi_{n+1})$ — регулярное выражение без коэффициентов от $n+1$ переменных $\xi_1, \ldots, \xi_n, \xi_{n+1}$. Выражение $\mathfrak{A}(a_1, \ldots, a_n, \Lambda)$ мы будем называть регулярной формулой над V, а язык, задаваемый таким выражением (в смысле стр. 24), — регулярным языком.

Теорема 5.6 (теорема Клини). *Класс регулярных языков совпадает с классом непустых* ОА-*языков. Более того, по всякой* ОА-*грамматике* Γ, *порождающей непустой язык, можно построить регулярную формулу* $\mathfrak{A}$, *задающую* $L(\Gamma)$, *а по всякой регулярной формуле* $\mathfrak{A}$ — ОА-*грамматику* Γ, *порождающую задаваемый этой формулой язык.*

Доказательство. а) Поскольку ОА-грамматики для языков $\{a_1\}, \ldots, \{a_n\}, \{\Lambda\}$ строятся тривиальным образом, построение нужной ОА-грамматики для заданной регулярной формулы обеспечивается теоремой 5.4.

б) Вторую половину теоремы докажем индукцией по числу s дуг диаграммы данной стандартной ОА-грамматики Γ. Если $s = 0$, язык $L(\Gamma)$ может быть непустым лишь тогда, когда он состоит из одной пустой цепочки, так что в этом случае имеем $\mathfrak{A} = \Lambda$. Пусть утверждение доказано для $s - 1$ и Γ — стандартная грамматика

с диаграммой D из s дуг. Выбросим из D какую-либо дугу, идущую из начального узла I в узел B и помеченную символом a. Полученную так диаграмму обозначим через D_1, диаграмму, возникающую из D_1 при перенесении начального узла в B, — через D_2, диаграмму, возникающую из D_1 при перенесении заключительного узла в I, — через D_3 и диаграмму, возникающую из D_2 при перенесении заключительного узла в I, — через D_4 (рис. 11). (На рис. 11 буквой K обозначен один из заключительных узлов диаграммы D, а цифрами 1, 2, 3, 4 в кружках — примеры путей в D_1, D_2, D_3, D_4 соответственно.) Грамматику с диаграммой D_i, $i = 1, 2, 3, 4$, будем обозначать Γ_i.

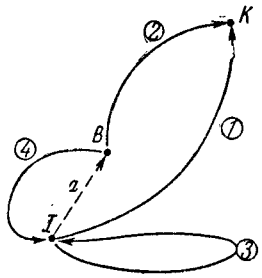


Рис. 11.

В силу индуктивного предположения можно построить регулярные формулы $\mathfrak{A}_i$, $i = 1, 2, 3, 4$, задающие языки $L(\Gamma_i)$.

Всякий полный путь в D, очевидно, либо является полным путем в D_1, либо представляется в виде $\xi\alpha\eta_1\alpha\eta_2 \ldots \alpha\eta_k\alpha\zeta$, где α — дуга (I, B), ξ — полный путь в D_3, $\eta_1, \ldots, \eta_k$ — полные пути в D_4 $(k = 0, 1, \ldots)$ и ζ — полный путь в D_2. Ясно также, что каждый путь, представимый в таком виде, является полным путем в D. Поэтому $L(\Gamma) = L(\Gamma_1) \cup L(\Gamma_3)(aL(\Gamma_4))^* aL(\Gamma_2)$, так что $L(\Gamma)$ задается регулярной формулой $\mathfrak{A}_1 \cup \mathfrak{A}_3(a\mathfrak{A}_4)^* a\mathfrak{A}_2$.

Следствие. *Класс* ОА-*языков есть замыкание класса конечных языков относительно объединения, умножения и итерации.*

Замечание. Теорема 5.6, как ясно из ее доказательства, останется справедливой, если вместо произвольных ОА-грамматик рассматривать только такие, в диаграммах которых нет циклов (т. е. ОА-грамматики, порождающие конечные языки), а вместо произвольных регулярных выражений — многочлены.

Пример. Языки, порождаемые грамматиками, диаграммы которых показаны на рис. 8, задаются соответственно формулами $((a \cup b)(d \cup bcb)^* bce)^*(\Lambda \cup (a \cup b)(d \cup bcb)^* b(e \cup \Lambda \cup c))$, $(a_1 \cup \ldots \cup a_n)^*(a_1 \cup \ldots \cup a_n)$, $a^* ab^* b$.

Дальнейшие примеры см. в упражнении 1.6.

В заключение докажем еще одну теорему, дающую очень удобный критерий автоматности языка.

Теорема 5.7. *Для того чтобы язык L в словаре V был* ОА-*языком, необходимо и достаточно, чтобы существовала согласованная с ним конгруэнция конечного индекса на V^*.* (Определения конгруэнции и согласованности см. ниже, стр. 316 и 318.)

Доказательство. а) Пусть $\Gamma = \langle V, W, I, R\rangle$ — стандартная ОА-грамматика и $L(\Gamma) = L$. Для каждой цепочки $x \in V^*$ обозначим через $O(x)$ множество всех пар (A, B), A, $B \in W$, обладающих следующим свойством: в диаграмме Γ существует путь из A в B, производящий x. Положим xRy, если $O(x) = O(y)$. Легко видеть, что R — конгруэнция. Индекс R не превосходит числа всевозможных подмножеств множества W^2 и, следовательно, конечен. При этом $x \in L$ тогда и только тогда, когда $O(x)$ содержит пару вида (I, C), где C — заключительный узел диаграммы Γ. Таким образом, принадлежность x к L вполне определяется множеством $O(x)$, а это означает согласованность R с L.

б) Пусть R — согласованная с L конгруэнция конечного индекса на V^*, A — произвольный R-класс и $x \in V^*$. Все цепочки вида zx, где $z \in A$, будут, очевидно, принадлежать одному и тому же R-классу B; мы будем говорить, что x переводит A в B. Определим теперь ОА-грамматику $\Gamma = \langle V, W, I, R\rangle$ следующим образом: W есть множество всех R-классов; I — класс, содержащий Λ; R состоит из всевозможных правил вида $A \to aB$, где a — элемент V, переводящий A в B, и вида $A \to \Lambda$, где $A \subseteq L$. Ясно, что цепочка $x \in V^*$ тогда и только тогда производится путем диаграммы Γ, идущим из A в B, когда она переводит A в B. В частности, x производится полным путем, т. е. принадлежит $L(\Gamma)$, тогда и только тогда, когда x переводит I в некоторый «заключительный» класс, а это, поскольку $\Lambda \in I$, равносильно тому, что x сама принадлежит «заключительному» классу; но объединение таких классов совпадает с L.

Из доказанной теоремы и леммы ПП. 1 (см. ниже, стр. 318) вытекает

Следствие. *Для того чтобы язык L в словаре V был* ОА-*языком, необходимо и достаточно, чтобы отношение $\underset{V, L}{\Leftrightarrow}$ имело конечный индекс.*

Теорема 5.7 и сформулированное только что следствие часто оказываются полезными, когда надо доказать, что тот или иной язык не автоматный. Рассмотрим, например, язык $L = \{a^n b^n \mid n = 1, 2, \ldots\}$. Если бы отношение $\underset{\{a, b\}, L}{\Leftrightarrow}$ имело конечный индекс, то среди цепочек a^n, $n = 1, 2, \ldots$, нашлись бы две взаимозамещаемые; но взаимозамещаемость цепочек a^k и a^l при $k \neq l$ невозможна, так как $a^k b^k \in L$ и $a^k b^l \notin L$. Дальнейшие примеры см. в упражнении 5.16.

§ 5.3. Линейные, металинейные и итерационно-линейные языки

В этом и следующем параграфах будут рассмотрены некоторые классы грамматик, промежуточные между классом ОА-грамматик и классом всех ОБ-грамматик, а также соответствующие классы МП-машин.

Начнем с рассмотрения так называемых линейных грамматик, наиболее близких к автоматным.

ОБ-грамматика называется линейной, если правая часть каждого ее правила содержит не более одного вхождения вспомогательного символа. Язык, порождаемый такой грамматикой, называется линейным языком.

Всякая ОА-грамматика, очевидно, линейна. Грамматики примеров 5, 6, 9 и 12 из § 1.3 не автоматны, но линейны.

Пусть фиксированы основной и вспомогательный словари V и W. P_1- или P_2-дерево, узлы которого помечены символами из $V \cup W$, мы назовем линейным, если никакой его узел не подчиняет более одного узла, помеченного вспомогательным символом. Очевидно, ОБ-грамматика тогда и только тогда линейна, когда линейно каждое ее дерево вывода.

Линейные языки легко охарактеризовать в терминах машин Тьюринга, подобно тому, как это делалось выше для других классов языков. Назовем МП-машину однодорожечной, если множество ее состояний можно

разбить на три попарно непересекающиеся подмножества: Q_1 («состояния прямого хода»), Q_2 («состояния обратного хода») и Q_3 («поворотные состояния») так, что: а) ни в одной инструкции не встречаются одновременно состояния из Q_1 и из Q_2; б) если $q \in Q_3$, то q может встречаться только в инструкциях вида $q' \to Aq$ и $Bq \to q''$, где $q' \in Q_1$, $q'' \in Q_2$; в) начальное состояние принадлежит Q_1, а все заключительные состояния принадлежат Q_2. Очевидно, в любом полном вычислении однодорожечной МП-машины направление движения меняется не более одного раза, иначе говоря, дерево вычисления линейно. [В соответствующем Д-автомате головка может двигаться только по одной «дорожке»; отсюда название «однодорожечная МП-машина». Термин «дорожка» (англ. path) обычен в работах по «предсказуемостному анализу» (см. стр. 141).]

Из лемм 4.12 и 4.13 немедленно вытекает

Теорема 5.8. а) *Для любой линейной* ОБ-*грамматики можно построить эквивалентную ей однодорожечную* МП-*машину.* б) *Для любой однодорожечной* МП-*машины можно построить эквивалентную ей линейную* ОБ-*грамматику.*

Отметим некоторые частные случаи линейных грамматик.

Линейная ОБ-грамматика называется левосторонней, соответственно правосторонней, если правая часть каждого ее правила либо не содержит вхождения вспомогательного символа, либо имеет вид Bx, соответственно xB, где x — цепочка в основном словаре. Читатель без труда докажет, что для каждой односторонней, т. е. левосторонней или правосторонней, линейной Б(ОБ)-грамматики можно построить эквивалентную ей А(ОА)-грамматику.

Линейная ОБ-грамматика называется симметричной, если правая часть каждого ее правила либо не содержит вхождения вспомогательного символа, либо имеет вид xAy, где A — вспомогательный символ и $|x| = |y|$.

Назовем МП-машину равномерной, если в любом ее вычислении между каждыми двумя следующими друг за другом перемещениями рабочей головки читается точно один входной символ. Читателю предо-

ставляется доказать, пользуясь леммами 4.12 и 4.13, что для всякой симметричной линейной ОБ-грамматики можно построить эквивалентную ей равномерную однодорожечную МП-машину. (Обратное также справедливо.) Отсюда, в частности, следует, поскольку конечный автомат можно считать (с несущественной модификацией) частным случаем равномерной однодорожечной МП-машины, что всякий ОА-язык порождается симметричной линейной ОБ-грамматикой, и такая грамматика может быть построена по данной ОА-грамматике.

Отметим еще, что класс линейных языков эффективно замкнут относительно объединения, а также относительно пересечения с ОА-языком и умножения слева и справа на ОА-язык.

Непосредственным обобщением линейных ОБ-грамматик являются металинейные ОБ-грамматики. Так называются грамматики, правила которых делятся на две группы: а) правила вида $I \to \omega$, где I — начальный символ и ω не содержит вхождений I; б) правила вида $A \to \varphi$, где $A \neq I$ и φ содержит не более одного вхождения вспомогательного символа, причем этот символ отличен от I. Язык, порождаемый металинейной ОБ-грамматикой, называется металинейным языком.

Из определения ясно, что каждый металинейный язык является объединением конечного числа языков, каждый из которых есть произведение нескольких линейных языков (причем линейные грамматики, порождающие эти языки, эффективно строятся по металинейной грамматике, порождающей данный язык). С другой стороны, читатель без труда докажет, что класс металинейных языков эффективно замкнут относительно объединения и умножения. Таким образом, класс металинейных языков представляет собой замыкание класса линейных языков относительно этих двух операций.

Чтобы получить «автоматную характеристику» металинейных языков, нам придется ввести два новых класса МП-машин.

Будем называть МП-машину чисто стирающей, если множество ее состояний можно разбить на четыре попарно непересекающиеся подмножества: Q_1 («состояния прямого хода»), Q_2 («состояния обратного хода»),

Q_3 («состояния перехода к прямому ходу») и Q_4 («состояния перехода к обратному ходу») так, что: а) ни в одной инструкции не встречаются одновременно состояния из Q_1 и из Q_2; б) если $q \in Q_3$, то q может встречаться только в инструкциях вида $A_0 q'' \to q$ и $q \to A_0 q'$, где $q'' \in Q_2 \cup Q_4$, $q' \in Q_1 \cup Q_4$ и A_0 — выделенный рабочий символ, один и тот же во всех инструкциях этого вида и не входящий ни в какие другие инструкции; в) если $q \in Q_4$, то q может встречаться только в инструкциях вида $q' \to Aq$ и $Bq \to q''$, где $q' \in Q_1 \cup Q_3$, $q'' \in Q_2 \cup Q_3$; г) начальное состояние и все заключительные состояния принадлежат Q_3.

Ясно, что чисто стирающая МП-машина может переходить от уничтожения рабочих ячеек к их созданию, лишь уничтожив всю рабочую ленту (поскольку в крайней слева рабочей ячейке всегда пишется символ A_0, который нигде больше не может быть записан).

Будем говорить далее, что МП-машина M имеет ограниченное число поворотов, если существует такое число k, что в любом полном вычислении этой машины направление движения рабочей головки изменяется не более k раз. Наименьшее из таких чисел k будем обозначать $\Pi(M)$; если $\Pi(M) = l$, будем называть M машиной с l поворотами.

Таким образом, однодорожечная МП-машина — это чисто стирающая машина с одним поворотом. Впрочем, для любой МП-машины с одним поворотом очевидным образом строится эквивалентная ей однодорожечная.

В отличие от предыдущих классов МП-машин, машины с ограниченным числом поворотов определяются не в терминах программ, а в терминах вычислений. Однако ниже (замечание к теореме 5.11) будет показано, что по программе произвольной МП-машины M можно распознать, имеет ли она ограниченное число поворотов, и если да, найти $\Pi(M)$.

Теперь может быть сформулирована

Теорема 5.9. а) *Для любой металинейной* ОБ-*грамматики можно построить эквивалентную ей чисто стирающую* МП-*машину с ограниченным числом поворотов.* б) *Для любой чисто стирающей* МП-*машины с ограниченным числом поворотов можно построить эквивалентную ей металинейную* ОБ-*грамматику.*

Для доказательства этой теоремы нам понадобится следующая очевидная лемма.

Лемма 5.1. *Пусть M и M' — МП-машины, и пусть M_1, M_2, M^* — их параллельная и последовательная композиции и итерация M соответственно. Тогда*: а) *если M и M' — чисто стирающие машины, то таковы же M_1, M_2 и M^**; б) *если M и M' имеют ограниченное число поворотов, то это верно и для M_1 и M_2, причем*

$$\Pi(M_1) \leqslant \max(\Pi(M), \Pi(M')), \quad \Pi(M_2) \leqslant \Pi(M) + \Pi(M').$$

Заметим теперь, что для машин с одним поворотом утверждение пункта а) теоремы 5.9 справедливо в силу теоремы 5.8; поэтому для общего случая оно следует из леммы 5.1.

Доказательство пункта б) нам будет удобно ненадолго отложить.

С помощью теоремы 5.9 легко указать примеры нелинейных металинейных языков (упражнение 5.26).

Металинейные языки допускают естественное обобщение в двух направлениях. С одной стороны, можно рассматривать языки, допускаемые машинами с ограниченным числом поворотов, отказавшись от требования чистоты стирания, с другой — допускаемые чисто стирающими машинами без ограничения на число поворотов. Первой возможности будет посвящен следующий параграф, а сейчас мы займемся второй.

Теорема 5.10. а) *Для любого бескоэффициентного регулярного выражения $\mathfrak{A}(\xi_1, \ldots, \xi_n, \xi_{n+1})$ от переменных $\xi_1, \ldots, \xi_n, \xi_{n+1}$ и любых n линейных* ОБ-*грамматик $\Gamma_1, \ldots, \Gamma_n$ можно построить чисто стирающую* МП-*машину, допускающую язык $\mathfrak{A}(L(\Gamma_1), \ldots, L(\Gamma_n), \Lambda)$.* б) *Для любой чисто стирающей* МП-*машины M можно построить бескоэффициентное регулярное выражение $\mathfrak{A}(\xi_1, \ldots, \xi_n, \xi_{n+1})$ и линейные* ОБ-*грамматики $\Gamma_1, \ldots, \Gamma_n$ такие, что $\mathfrak{A}(L(\Gamma_1), \ldots, L(\Gamma_n), \Lambda) = L(M)$.*

Доказательство. а) Следует из теоремы 5.8 и леммы 5.1. б) Пусть M — чисто стирающая МП-машина, и пусть q_α, q_β — два произвольных состояния перехода к прямому ходу. Выбросим из программы M все инструкции, содержащие состояния перехода к прямому ходу, кроме тех, которые содержат q_α в левой части или q_β в правой, и при этом заменим q_β его «двойником» — новым состоянием q'_β («двойники» различных состояний различны). Начальным состоянием новой машины, которую будем обозначать $M_{\alpha\beta}$, объявим q_α, единственным заключительным состоянием — q'_β. $M_{\alpha\beta}$ является, очевидно, машиной с одним поворотом *), и для нее по теореме 5.8 можно построить эквивалентную линейную ОБ-грамматику $\Gamma_{\alpha\beta}$. Сопоставим каждой машине $M_{\alpha\beta}$ новый символ $a_{\alpha\beta}$ и рассмотрим диаграмму D, строящуюся так: узлами ее служат состояния перехода к прямому ходу машины M — пусть это $q_1, \ldots, q_p$; начальными и заключительными узлами будут начальное и заключительные состояния M соответственно; из q_α в q_β идет точно одна дуга, и эта дуга помечена символом $a_{\alpha\beta}$. Если L_0 — множество всевозможных цепочек, производимых полными путями диаграммы, то, очевидно,

$$L(M) = S(L_0; a_{11}, a_{12}, \ldots, a_{pp} | L(M_{11}), L(M_{12}), \ldots, L(M_{pp})).$$

Но по диаграмме можно построить регулярное выражение, представляющее L_0 (теорема 5.6). Остается заменить в этом выражении каждый символ $a_{\alpha\beta}$ на $L(\Gamma_{\alpha\beta})$.

В частности, если M — машина с ограниченным числом поворотов, то в диаграмме D длины путей ограничены, так что D не имеет циклов, и поэтому соответствующее регулярное выражение является, — точнее, может быть сделано — многочленом (замечание к теореме 5.6). Следовательно, в этом случае $L(M)$ получается из $L(\Gamma_{\alpha\beta})$ с помощью объединения и умножения; но класс металинейных языков, как уже отмечалось, эффективно замкнут относительно этих операций. Тем самым доказана теорема 5.9, б).

Назовем языки, допускаемые чисто стирающими МП-машинами, итерационно-линейными языками. Теорема 5.10 позволяет указать следующий «канонический вид» ОБ-грамматики, порождающей итерационно-линейный язык: $\Gamma = \langle V, W, I, R\rangle$, где $W = W_1 \cup W_2$, $R = R_1 \cup R_2$, причем $W_1 \cap W_2 = \varnothing$; R_1 состоит из правил вида $A \to BC$, где A, $C \in W_1$, $B \in W_2$; R_2 состоит из правил вида $A \to \omega$, где $A \in W_2$ и ω содержит

*) Ради этого нам и пришлось заменить q_β «двойником». Впрочем, на самом деле «двойник» необходим только при $\alpha = \beta$.

не более одного вхождения символа из W_2 и не содержит вхождений символов из W_1. Такие грамматики можно назвать итерационно-линейными.

Из теоремы 5.10 следует также, что класс итерационно-линейных языков представляет собой замыкание класса линейных языков относительно объединения, пересечения и итерации.

Существуют итерационно-линейные языки, не являющиеся металинейными (упражнение 5.27).

О «допускающих возможностях» детерминированных машин рассмотренных классов см. упражнение 5.33.

§ 5.4. Грамматики с ограниченной активной емкостью выводов. ОАЕВ-языки

Займемся теперь изучением языков, допускаемых МП-машинами с ограниченным числом поворотов. Мы получим для этих языков, подобно металинейным и итерационно-линейным, две характеристики: в терминах грамматик и в алгебраических терминах.

Пусть Γ — грамматика с основным словарем V и вспомогательным словарем W, и $\omega \in (V \cup W)^*$. Назовем активной емкостью цепочки число вхождений в нее вспомогательных символов. Активной емкостью вывода в Γ будем называть наибольшую активную емкость входящей в этот вывод цепочки*). Если существует число k, мажорирующее активные емкости всех полных выводов в Γ, мы будем называть Γ грамматикой с ограниченной активной емкостью выводов (ОАЕВ-грамматикой); наименьшее из таких чисел k будет обозначаться $I_0(\Gamma)$. Языки, порождаемые ОАЕВ-грамматиками, мы будем называть ОАЕВ-языками. Очевидно, для приведенной ОБ-грамматики Γ тогда и только тогда $I_0(\Gamma) = 1$, когда Γ линейна.

Введем еще одно вспомогательное понятие. Пусть Γ — произвольная ОБ-грамматика, T — дерево вывода в

*) Обе эти величины фактически уже рассматривались нами. Именно, фигурирующая в определении активной емкости грамматики (стр. 57) функция $f(\Gamma, D, i)$ — это активная емкость цепочки ω_i, а $f(\Gamma, D, x)$ — активная емкость вывода D. (Активная емкость Б-грамматик изучается ниже, § 7.2.)

ней и T' — P_1-дерево, полученное из T выбрасыванием всех узлов, помеченных основными символами. Число висячих узлов дерева T' будем называть приведенным индексом ветвления дерева T (не T'!). Нетрудно видеть, что это число равно наибольшему значению активной емкости вывода, отвечающего дереву T. (В самом деле, это наибольшее значение достигается на выводе, в котором правила, не содержащие в правых частях вспомогательных символов, применяются в последнюю очередь; а цепочка этого вывода, имеющая наибольшую активную емкость, — точнее, последняя из таких цепочек — должна «проходить» через все висячие узлы T'.)

Таким образом, ОБ-грамматика Γ тогда и только тогда является ОАЕВ-грамматикой, когда множество приведенных индексов ветвления полных выводов в Γ имеет конечную наименьшую верхнюю грань; последняя, если она существует, равна $I_0(\Gamma)$.

Теперь мы можем доказать следующее утверждение.

Лемма 5.2. *Существует алгоритм, позволяющий по любой* ОБ-*грамматике* Γ *распознать, является ли она* ОАЕВ-*грамматикой, и в случае положительного ответа найти* $I_0(\Gamma)$.

Доказательство. Назовем рангом вспомогательного символа A ОБ-грамматики $\Gamma = \langle V, W, I, R\rangle$ наименьшую верхнюю грань множества приведенных индексов ветвления деревьев выводов в Γ, начинающихся символом A и заканчивающихся цепочками в V. Кроме того, каждому основному символу припишем ранг 0. Все символы конечного ранга можно найти — и определить их ранги — следующим способом. Определим последовательность $W_0, W_1, \ldots$ подмножеств словаря $V \cup W$ так: $W_0 = V$; если определены $W_0, \ldots, W_n$, то W_{n+1} состоит из тех $A \in W - (W_0 \cup \ldots \cup W_n)$, которые обладают следующим свойством: какова бы ни была последовательность $A_0 \to \xi_1 A_1 \eta_1, A_1 \to \xi_2 A_2 \eta_2, \ldots, A_{s-1} \to \xi_s A_s \eta_s$ правил Γ такая, что $A_0 = A$ и $A_0, \ldots, A_s \in W - (W_0 \cup \ldots \cup W_n)$, все цепочки ξ_i, η_i принадлежат V^*. (Это свойство, очевидно, эффективно проверяемо.) Непосредственная индукция по n показывает, что для каждого W_n все принадлежащие ему символы имеют конечные ранги; способ их нахождения ясен из конструкции. В то же

время символы, не вошедшие ни в одно W_n, если такие есть, имеют бесконечный ранг. Действительно, если n_0 — наибольшее из чисел n, для которых W_n не пусты, то из всякого $B \in W - (W_0 \cup \ldots \cup W_{n_0})$ будет выводима некоторая цепочка вида $\xi C \eta$, где $C \in W - (W_0 \cup \ldots \cup W_{n_0})$ и $\xi\eta$ содержит вхождение вспомогательного символа; а поскольку то же самое имеет место для C и т. д., для любого p найдется выводимая из B цепочка, содержащая не менее p вхождений вспомогательных символов, что и обеспечивает бесконечность ранга B. Итак, Г тогда и только тогда будет ОАЕВ-грамматикой, когда ее начальный символ содержится в одном из W_n, и $I_0(\Gamma)$ будет равно в этом случае рангу начального символа.

Легко заметить, что понятие приведенного индекса ветвления позволяет охарактеризовать и МП-машины с ограниченным числом поворотов. Именно, если определить этот индекс для дерева вычисления так же, как для дерева вывода, то он будет равен числу переходов от создания ячеек к их уничтожению, иначе — половине увеличенного на единицу общего числа поворотов. Теперь без труда получается

Теорема 5.11. а) *Для любой* МП-*машины* M *с ограниченным числом поворотов можно построить эквивалентную ей* ОБ-ОАЕВ-*грамматику* Г *такую, что* $I_0(\Gamma) = 1/2(\Pi(M) + 1)$. б) *Для любой* ОБ-ОАЕВ-*грамматики* Г *можно построить эквивалентную ей* МП-*машину* M *с ограниченным числом поворотов такую, что* $\Pi(M) = 2I_0(\Gamma) - 1$.

Доказательство немедленно следует из лемм 4.12 и 4.13 и того факта, что конструкция, использованная при установлении леммы 4.14, сохраняет приведенные индексы ветвления деревьев.

Замечание. Это же обстоятельство вместе с леммой 5.2 дает алгоритм, позволяющий по любой МП-машине M распознать, имеет ли она ограниченное число поворотов, и в случае положительного ответа найти $\Pi(M)$.

Перейдем теперь к алгебраической характеристике ОБ-ОАЕВ-языков. Пусть $V = \{a_1, \ldots, a_k, b\}$ $(k \geqslant 0)$ и V' — произвольные словари, $L \subseteq V^*$, $L' \subseteq V'^*$, и пусть каждая цепочка языка L содержит не более одного вхождения b. Положим $S_b(L, L') = S(L; a_1, \ldots, a_k, b | \{a_1\}, \ldots, \{a_k\}, L')$. Операцию S_b будем называть центральной подстановкой L' в L (с центром b). Выражение, составленное из абстрактных символов $\xi_1, \ldots, \xi_n$ с помощью знаков центральной подстановки (с произвольными центрами), будем называть центрально-подстановочным выражением от (переменных) $\xi_1, \ldots, \xi_n$. Например, $S_c(\xi_1, S_a(\xi_2, S_c(\xi_3, \xi_1)))$ есть центрально-подстановочное выражение. Можно определить представление языка с помощью центрально-подстановочного выражения так же, как представление с помощью многочлена или регулярного выражения (стр. 24).

Мы покажем, что класс языков, представимых с помощью центрально-подстановочных выражений от линейных языков (иначе говоря, замыкание класса линейных языков относительно центральной подстановки), «эффективно совпадает» с классом ОБ-ОАЕВ-языков. Точнее, имеет место

Теорема 5.12. а) *Для любого центрально-подстановочного выражения* $\mathfrak{A}(\xi_1, \ldots, \xi_n)$ *и любых* n *линейных* ОБ-*грамматик* $\Gamma_1, \ldots, \Gamma_n$ *таких, что выражение* $\mathfrak{A}(L(\Gamma_1), \ldots, L(\Gamma_n))$ *определено**), *можно построить* ОБ-ОАЕВ-*грамматику* Г, *порождающую язык* $\mathfrak{A}(L(\Gamma_1), \ldots, L(\Gamma_n))$. б) *Для любой* ОБ-ОАЕВ-*грамматики* Г *можно построить центрально-подстановочное выражение* $\mathfrak{A}(\xi_1, \ldots, \xi_n)$ *и линейные грамматики* $\Gamma_1, \ldots, \Gamma_n$ *такие, что* $L(\Gamma) = \mathfrak{A}(L(\Gamma_1), \ldots, L(\Gamma_n))$.

Доказательство. а) Достаточно показать, что если Γ_1 и Γ_2 — ОБ-ОАЕВ-грамматики и каждая цепочка языка $L(\Gamma_1)$ содержит не более одного вхождения символа b, то можно построить ОБ-ОАЕВ-грамматику Γ_b, порождающую $S_b(L(\Gamma_1), L(\Gamma_2))$. Но при $\Gamma_i = \langle V, W_i, I_i, R_i \rangle$ $(i = 1, 2)$ и $W_1 \cap W_2 = \varnothing$ мы можем, очевидно, положить $\Gamma_b = \langle V, W_1 \cup W_2, I_1, R_1[b, I_2] \cup R_2 \rangle$, где $R_1[b, I_2]$ получается из R_1 заменой всех вхождений b вхождениями I_2. б) При $I_0(\Gamma) = 1$ утверждение тривиально. Пусть оно доказано для случая $I_0(\Gamma) < k$ $(k > 1)$, и для данной грамматики $\Gamma = \langle V, W, I, R \rangle$ имеет место $I_0(\Gamma) = k$. Рассмотрим произвольный полный вывод $D = (\omega_0, \ldots, \omega_s)$

---

*) Это значит, что если в данном выражении для каких-либо i, j в $L(\Gamma_i)$ подставляется $L(\Gamma_j)$ вместо символа b, то каждая цепочка языка $L(\Gamma_i)$ содержит не более одного вхождения b.

в Γ и обозначим через i_D наибольшее число i, для которого каждая из цепочек $\omega_0, \omega_1, \ldots, \omega_i$ содержит точно одно вхождение вспомогательного символа. Пусть L' — множество цепочек ω_{i_D} для всех полных выводов в Γ, $A_1, \ldots, A_t$ — все вспомогательные символы, встречающиеся в цепочках языка L', и $A_j \to \omega_{j1}, \ldots, A_j \to \omega_{js_j}$ $(j=1, \ldots, t)$ — всевозможные правила Γ с левой частью A_j, содержащие в правых частях либо более одного вхождения вспомогательного символа, либо ни одного. Для произвольных j, l, $1 \leqslant j \leqslant t$, $1 \leqslant l \leqslant s_j$, обозначим через $\alpha_{jl1}, \ldots, \alpha_{jlr_{jl}}$ все вхождения вспомогательных символов в ω_{jl} (очевидно, $r_{jl} \leqslant k$) и через $B_{jl1}, \ldots, B_{jlr_{jl}}$ — те вспомогательные символы, вхождениями которых являются $\alpha_{jl1}, \ldots, \alpha_{jlr_{jl}}$ соответственно.

Положим

$$\Gamma_{jlm} = \langle V, W, B_{jlm}, R\rangle \quad (m=1, \ldots, r_{jl}).$$

Ясно, что все Γ_{jlm} являются ОАЕВ-грамматиками и при этом $I_0(\Gamma_{jlm}) < k$. Введем новые символы $b_{jl1}, \ldots, b_{jlr_{jl}}$ и заменим каждое вхождение α_{jlm} вхождением символа b_{jlm}; полученную цепочку обозначим φ_{jl}. (Если ω_{jl} не содержит вспомогательных символов, то φ_{jl} совпадает с ω_{jl}.) Положим

$$L'' = S\left(L'; A_1, \ldots, A_t \middle| \{\varphi_{11}, \ldots, \varphi_{1s_1}\}, \ldots, \{\varphi_{t1}, \ldots, \varphi_{ts_t}\}\right).$$

Язык L'' порождается линейной грамматикой, получаемой из Γ заменой каждого правила $A_j \to \omega_{jl}$, $j=1, \ldots, t$; $l=1, \ldots, s_j$, правилом $A_j \to \varphi_{jl}$ и выбрасыванием всех остальных «нелинейных» правил. Но язык L может быть получен из L'' последовательным применением операций $S_{b_{111}}, S_{b_{112}}, \ldots, S_{b_{ts_tr_{ts_t}}}$ (причем в качестве подставляемых языков выступают $L(\Gamma_{111}), L(\Gamma_{112}), \ldots$ соответственно). Ввиду индуктивного предположения и того очевидного факта, что все φ_{jl} и Γ_{jlm} могут быть найдены эффективно, это завершает доказательство.

О «допускающих возможностях» детерминированных машин с ограниченным числом поворотов см. упражнение 5.33.

Замечания. 1) Из теорем 5.9 и 5.11 вытекает, что класс металинейных языков совпадает с пересечением классов итерационно-линейных и Б-ОАЕВ-языков.

2) Просмотрев доказательство теоремы 5.5, легко убедиться, что если фигурирующая там машина M_1 однодорожечная, чисто стирающая или с ограниченным числом поворотов, то и машина M будет такой же. Поэтому классы линейных, металинейных, итерационно-линейных и Б-ОАЕВ-языков эффективно замкнуты относительно пересечения с ОА-языками.

3) Все эти классы эффективно замкнуты также относительно гомоморфных отображений (в частности, проекций). Это непосредственно следует из определений соответствующих типов грамматик. (Действительно, в этих определениях существенно только расположение вхождений вспомогательных символов в правые части правил, а оно не изменяется при переходе от данной грамматики к грамматике для гомоморфного образа.) Справедлив, впрочем, и более общий факт, доказательство которого предоставляется читателю, — все указанные классы эффективно замкнуты относительно подстановки вместо элементарных символов любых ОА-языков.

Упражнения

5.1. Построить диаграммы грамматик примеров 1 и 3, а) из § 1.3.

5.2. Показать, что для всякой А(ОА)-грамматики можно построить эквивалентную ей однозначную А(ОА)-грамматику.

5.3. Показать, что коммутативное замыкание (упражнение 3.7) А-языка может не быть А-языком (и даже Б-языком).

5.4. По конечному автомату с программой $\{q_1 \# \to q_2, q_1a \to q_2, q_2b \to q_2, q_2b \to q_3, q_3a \to q_4, q_3a \to q_2, q_4b \to q_1\}$ и заключительными состояниями q_2, q_3 построить детерминированный конечный автомат, пользуясь конструкцией, примененной в доказательстве теоремы 5.3. Построить диаграммы обоих автоматов.

5.5. Показать, что для всякой грамматики, все правила которой имеют вид $A \to aB$, $aA \to B$, $A \to B$ и $A \to \Lambda$, где A, B — вспомогательные символы и a — основной символ, можно построить эквивалентную ей ОА-грамматику [Büchi 1964].

5.6. Показать, что всякий А-язык является проекцией подходящего стандартного А-языка. (Определение стандартного А-языка см. в § 6.5.)

5.7. Пусть V — словарь и $\#$ — символ, не принадлежащий V. Простой окрестностной грамматикой в алфавите V

[Шрейдер 1969] называется конечное множество S вхождений, именуемых окрестностями, символов из V в цепочки, принадлежащие множеству $\{\Lambda, \#\} V^+ \{\Lambda, \#\}$.

Окрестность $\alpha x*a*y\beta$, где α и β могут означать $\#$ или Λ, выполняется для вхождения $u*a*v$ символа a в цепочку $z = uav \in V^*$, если u и v представимы соответственно в виде $u = sx$, $v = yt$, причем, если $\alpha = \#$, то $s = \Lambda$, и если $\beta = \#$, то $t = \Lambda$.

Языком, определяемым простой окрестностной грамматикой S, называется множество всех цепочек из V^+, в которых для каждого вхождения символа выполняется хотя бы одна окрестность из S.

А-грамматика называется k-определенной (k — натуральное число), если она обладает следующим свойством: каковы бы ни были две цепочки x и y, производимые какими-либо путями ее диаграммы, если $[x]_k$ ($[y]_k$) означает наибольший конец x (соответственно y), длина которого не превосходит k, то из совпадения $[x]_k$ и $[y]_k$ следует совпадение последних узлов соответствующих путей.

а) Показать, что класс языков, определяемых простыми окрестностными грамматиками, совпадает (и притом «эффективно») с классом языков, порождаемых k-определенными А-грамматиками (при всевозможных k).

б) Показать, что для всякой простой окрестностной грамматики можно построить эквивалентную ей простую окрестностную грамматику, все окрестности которой имеют вид $\alpha x*a*\beta$ ($\alpha, \beta = \Lambda, \#$).

в) Показать, что всякий стандартный А-язык (см. ниже, § 6.5) порождается 1-определенной А-грамматикой.

5.8. Определим конечный автомат с выходом аналогично тому, как определялась МП-машина с выходом (упражнение 4.25). Для конечного автомата с выходом M' определим язык $L_0(M')$ точно так же, как это делалось для МП-машины с выходом. Показать, что:

а) для любого конечного автомата M можно построить такой конечный автомат с выходом M', что $L_0(M') = L(M)$;

б) для любого конечного автомата с выходом M' можно построить такой конечный автомат M, что $L(M) = L_0(M')$.

5.9. Определим преобразование, осуществляемое конечным автоматом с выходом (упражнение 5.8), аналогично тому, как это делалось для МП-машин с выходом (упражнение 4.25).

а) Показать, что классы ОА-языков и ОБ-языков эффективно замкнуты относительно преобразований, осуществляемых конечными автоматами с выходом.

б) Верно ли это для классов линейных, металинейных, итерационно-линейных и ОБ-ОАЕВ-языков?

5.10. Указать способ построения по диаграмме ОА-грамматики Γ с основным словарем V диаграммы ОА-грамматики, порождающей язык $\Pi p_Z L(\Gamma)$, $Z \subseteq V$.

5.11. Показать, что класс ОА-языков эффективно замкнут относительно обращения. (Определение см. в сноске на стр. 156).

5.12. Показать, что классы линейных, металинейных, итерационно-линейных и ОБ-ОАЕВ-языков эффективно замкнуты относительно пересечения с ОА-языками и относительно подстановки ОА-языков.

5.13. Множество натуральных чисел E (или, что то же самое, язык в словаре $\{|\}$, см. стр. 34) называется периодическим, если существуют такие k, l, что при $n > k$ из $n \in E$ следует $n + l \in E$. Показать, что множество натуральных чисел является периодическим тогда и только тогда, когда оно есть ОА-язык, причем по наименьшим соответствующим k и l можно построить нужную ОА-грамматику, и обратно, по такой грамматике можно найти наименьшие k и l.

5.14. а) Указать алгоритм, позволяющий для любой ОА-грамматики $\langle V, W, I, R\rangle$ и любого множества $O \subseteq W^2$ распознать, существует ли такая цепочка $x \in V^*$, что $O = O(x)$ (см. доказательство теоремы 5.7).

б) Указать алгоритм, позволяющий для любой ОА-грамматики $\langle V, W, I, R\rangle$ и любых цепочек $x, y, z \in V^*$ распознать, переводит ли z класс $O(x)$ в класс $O(y)$.

5.15. Будем говорить, что отношение эквивалентности $\sim$ на V^*, где V — словарь, совместимо с конкатенацией справа, если для любых $x, y, z \in V^*$ из $x \sim y$ следует $xz \sim yz$. Показать, что:

а) Если для произвольной ОА-грамматики $\Gamma = \langle V, W, I, R\rangle$ определить отношение $\sim$ на V^* следующим образом: $x \sim y$ тогда и только тогда, когда $\Pi(x) = \Pi(y)$, где $\Pi(u)$ означает множество тех $A \in W$, которые являются концами путей, начинающихся в I и производящих u, то $\sim$ есть эквивалентность, согласованная с $L(\Gamma)$ и совместимая с конкатенацией справа.

б) Для того чтобы язык L в словаре V был ОА-языком, необходимо и достаточно, чтобы существовала согласованная с L и совместимая с конкатенацией справа эквивалентность конечного индекса на V^*.

в) Для того чтобы язык L в словаре V был ОА-языком, необходимо и достаточно, чтобы отношение R, определяемое следующим образом:

$$xRy \equiv (\forall z \in V^*)\,[xz \in L \equiv yz \in L],$$

имело конечный индекс.

5.16. Показать, что следующие линейные языки не являются автоматными: $\{x\hat{x} \mid x \in V^+\}$; $\{xc\hat{x} \mid x \in V^+\}$ (в обоих случаях мощность $V \geqslant 2$); язык примера 12 из § 1.3.

5.17. а) Доказать следствие из теоремы 5.5 непосредственно, т. е. указать способ построения по Б (ОБ)-грамматике Γ_1 и А (ОА)-грамматике Γ_2 такой Б (ОБ)-грамматики Γ, что $L(\Gamma) = L(\Gamma_1) \cap L(\Gamma_2)$. Показать, что если при этом Γ_1 однозначна, то и Γ можно сделать однозначной.

б) Показать, что Б-язык

$$\{x \mid x \in \{a, b, c, d\}^+;\ |x|_a = |x|_b \vee |x|_c = |x|_d\}$$

неоднозначен.

5.18. Для того чтобы каждому дереву вывода в ОБ-грамматике Γ отвечало единственное растянутое дерево вывода, необходимо и достаточно, чтобы Γ была линейной. Доказать.

5.19. Показать, что для всякой линейной ОБ-грамматики можно построить эквивалентную ей линейную ОБ-грамматику, все правила

которой имеют вид $A \to aB$, $A \to Ba$ или $A \to \Lambda$, где A, B — вспомогательные символы, a — основной символ.

5.20. Определим диаграмму линейной ОБ-грамматики $\Gamma = \langle V, W, I, R\rangle$ с правилами вида, указанного в упражнении 5.19, следующим образом [Диковский 1970]: узлами диаграммы будут элементы W; если $A \to aB \in R$, то из A в B идет дуга, помеченная парой (a, Λ); если $A \to Ba \in R$, то из A в B идет дуга, помеченная парой (Λ, a); I — начальный узел; если $A \to \Lambda \in R$, то A — заключительный узел. Полный путь определяется так же, как для диаграммы ОА-грамматики.

а) Определить цепочку, производимую путем в диаграмме, так, чтобы множество цепочек, производимых полными путями, совпало с $L(\Gamma)$.

б) Обобщить понятие диаграммы на случай произвольной линейной ОБ-грамматики.

5.21. а) Назовем двуленточным конечным автоматом (ДК-автоматом) машину Тьюринга с двумя лентами $Л_1$ и $Л_2$, каждая из которых имеет свою головку, общим для обеих лент алфавитом, инструкциями только типа (i) (стр. 42) и множеством состояний, разбитым на два непересекающихся подмножества Q_1 и Q_2 так, что инструкция, имеющая в левой части состояние из Q_i ($i = 1, 2$), выполняется на ленте $Л_i$. ДК-автомат допускает пару цепочек (x, y), если при записанных на $Л_1$ и $Л_2$ цепочках x и y соответственно автомат может, начав вычисление в начальном состоянии с головками, обозревающими левые граничные маркеры, закончить его в заключительном состоянии с головками, обозревающими правые граничные маркеры. Язык, допускаемый ДК-автоматом M (обозначение: $L(M)$), есть множество допускаемых им пар цепочек. ДК-автомат M и грамматика Γ эквивалентны, если $L(\Gamma) = \{x\hat{y} \mid (x, y) \in L(M)\}$. Показать, что для всякого ДК-автомата можно построить эквивалентную ему линейную ОБ-грамматику и обратно [Rosenberg 1967].

б) Пользуясь результатом предыдущего пункта, показать, что подстановка автоматных языков в линейный дает линейный язык.

5.22. а) Показать, что для всякого детерминированного ДК-автомата (упражнение 5.21) можно построить эквивалентную ему однозначную линейную ОБ-грамматику.

б) Показать, что язык $\{xc\hat{x}y \mid x \in \{a, b\}^*, y = \Lambda \vee y \in d\{a, b\}^*\}$, порождаемый однозначной линейной ОБ-грамматикой, не допускается никаким детерминированным ДК-автоматом [Диковский 1970].

5.23. Показать, что если $L \subseteq a^*b^*$, где a, b — элементарные символы, и $K(L)$ полулинейно (см. § 4.3), то L — линейный язык.

Замечание. Отсюда следует, что всякий ОБ-язык, содержащийся в a^*b^*, является линейным.

5.24. Если правая часть каждого правила НС-грамматики Γ содержит не более одного вхождения вспомогательного символа, то для Γ можно построить эквивалентную ей линейную Б-грамматику. Доказать.

5.25. Указать способ непосредственного построения по данной ОА-грамматике эквивалентной ей симметричной линейной ОБ-грамматики.

5.26. Показать, что языки

$$\{a^{n_1}b^{n_1} \dots a^{n_k}b^{n_k} \mid n_i = 0, 1, \dots\} \qquad (k = 2, 3, \dots)$$

и

$$\{x_1\hat{x}_1 \dots x_k\hat{x}_k \mid x_1, \dots, x_k \in V^*, \mu(V) \geqslant 2\}$$

допускаются чисто стирающими МП-машинами с $2k - 1$ поворотами и не допускаются никакими МП-машинами с меньшим числом поворотов.

5.27. Показать, что итерационно-линейный язык $\{a^nb^n \mid n = 0, 1, \dots\}^*$ не является ОБ-ОАЕВ-языком. Указать другие аналогичные примеры.

5.28. Показать, что языки

$$\{a^{k+l}b^kc^ma^mb^l \mid k, l, m = 0, 1, \dots\},$$
$$\{a^{k+l}b^ka^mb^{l+m} \mid k, l, m = 0, 1, \dots\},$$
$$\{xy\hat{y}z\hat{z}\hat{x} \mid x, y, z \in V^*, \mu(V) \geqslant 2\}$$

допускаются МП-машинами с тремя поворотами, но не допускаются никакими чисто стирающими МП-машинами.

5.29. Показать, что для каждой чисто стирающей МП-машины M можно построить эквивалентную ей МП-машину M', в которой для каждого полного вычисления найдется равносильное ему полное вычисление такое, что рабочая головка на каждом заданном расстоянии от начала ленты бывает не более четырех раз.

5.30. Пусть $\mathscr{L}_0$ — класс всевозможных языков, каждый из которых состоит из одной однобуквенной цепочки, и при каждом $i = 0, 1, \dots$ $\mathscr{L}_{i+1}$ есть класс всевозможных языков вида $S(L; a_1, \dots, a_n \mid L_1, \dots, L_n)$, где L — линейный язык и $L_1, \dots, L_n \in \mathscr{L}_0 \cup \dots \cup \mathscr{L}_i$. Пусть, кроме того, $\mathfrak{M}_i$ и $\mathfrak{N}_i$ ($i = 0, 1, \dots$) означают замыкание $\mathscr{L}_i$ относительно объединения и умножения, соответственно объединения, умножения и итерации. Показать, что для любого $i = 0, 1, \dots$ имеет место $\mathscr{L}_i \subseteq \mathfrak{M}_i \subset \mathfrak{N}_i \subseteq \mathscr{L}_{i+1}$.

5.31. Непосредственно (не пользуясь грамматиками) доказать «эффективную эквивалентность» центрально-подстановочных выражений и МП-машин с ограниченным числом поворотов.

5.32. Показать, что всякий ОБ-язык, содержащийся в $\omega_1^* \dots \omega_k^*$, где $\omega_1, \dots, \omega_k$ — фиксированные цепочки, порождается при $k = 1$ ОА-грамматикой и при $k > 1$ — ОБ-ОАЕВ-грамматикой Γ такой, что $l_0(\Gamma) \leqslant k - 1$ (ср. замечания к теореме 4.6 и к упражнению 5.23).

(Языки указанного здесь вида называются ограниченными; об ограниченных ОБ-языках см. [Ginsburg 1966, гл. V].)

5.33. Показать, что:

а) Линейный язык

$$P_k = \{a^{m_1}b^{n_1} \dots a^{m_k}b^{n_k} \mid m_i, n_i = 0, 1, \dots;$$
$$m_1 = n_1 \vee \dots \vee m_k = n_k\} \qquad (k = 2, 3, \dots)$$

допускается детерминированной чисто стирающей МП-машиной с

$2k-1$ поворотами и не допускается никакой детерминированной МП-машиной с меньшим числом поворотов.

б) Линейный язык $P_2 \cup P_3 \cup \ldots$ допускается детерминированной чисто стирающей МП-машиной и не допускается никакой детерминированной МП-машиной с ограниченным числом поворотов.

в) Линейный язык

$$R_k = \{a^{p_1}b^{q_1}ca^{p_2}b^{q_2}c \ldots ca^{p_{k-1}}b^{q_{k-1}}ca^{p_k}b^{q_k+r_k}cb^{r_{k-1}}c \ldots$$
$$\ldots cb^{r_2}cb^{r_1} \mid p_i, q_i, r_i = 0, 1 \ldots; p_1 = q_1 + r_1 \vee \ldots \vee p_k = q_k + r_k\}$$
$$(k = 2, 3, \ldots)$$

допускается детерминированной МП-машиной с $2k-1$ поворотами и не допускается никакой детерминированной МП-машиной с меньшим числом поворотов и никакой детерминированной чисто стирающей МП-машиной.

г) Б-ОАЕВ-язык $R_2 \cup R_3 \cup \ldots$ (порождаемый грамматикой Г, для которой $I_0(\Gamma) = 2$) не допускается никакой детерминированной МП-машиной с ограниченным числом поворотов и никакой детерминированной чисто стирающей МП-машиной.

5.34. Показать, что в упражнении 4.25, в) можно взять в качестве Г итерационно-линейную ОБ-грамматику и в качестве M_1 чисто стирающую МП-машину с выходом.

ГЛАВА 6

ДОПОЛНИТЕЛЬНЫЕ СВЕДЕНИЯ О БЕСКОНТЕКСТНЫХ ГРАММАТИКАХ. ДРУГИЕ СПОСОБЫ ЗАДАНИЯ БЕСКОНТЕКСТНЫХ ЯЗЫКОВ

Наряду с Б-грамматиками и МП-машинами существует ряд других способов задания Б-языков. Многие из них вызваны к жизни потребностями приложений: различным способам задания языка отвечают различные способы описания синтаксической структуры его предложений (цепочек), отражающие различные лингвистические аспекты этой структуры. Некоторые из таких способов будут рассмотрены в настоящей главе. Здесь же нам будет удобно изложить еще несколько фактов, относящихся к «основным» способам задания Б-языков — Б-грамматикам и МП-машинам.

§ 6.1. Категориальные грамматики

Рассмотрим конечное множество W, которое будем называть *словарем категорий*, а его элементы — *элементарными категориями*. Введем в рассмотрение символы \, /, [,], не принадлежащие W. Из этих символов и элементов W будем строить выражения специального вида — *категории* (*над* W). Именно:

I. Всякая элементарная категория есть категория.

II. Если Φ, Ψ — категории, то выражение $[\Phi/\Psi]$ (читается «Φ над Ψ») и $[\Phi\backslash\Psi]$ (читается «Φ под Ψ») — также категории.

III. Всякая категория является таковой либо в силу I, либо в силу II.

Множество всех категорий над W будем называть *категориальной системой над* W и обозначать

$K(W)$. Множество всех цепочек, составленных из категорий над W, будет обозначаться $K(W)^*$.

Пусть ξ, η — цепочки категорий. Мы будем говорить, что ξ непосредственно сокращается до η, если либо $\xi = \zeta\Phi[\Phi\backslash\Psi]\theta$, $\eta = \zeta\Psi\theta$, либо $\xi = \zeta[\Phi/\Psi]\Psi\theta$, $\eta = \zeta\Phi\theta$, где $\Phi, \Psi \in K(W)$ и ζ, θ — цепочки категорий.

Замечания. 1. Можно рассматривать категорию $[\Phi\backslash\Psi]$ как оператор, действующий справа на категорию Φ и превращающий ее в Ψ. Аналогично для $[\Phi/\Psi]$.

2. Для запоминания удобно представлять себе непосредственное сокращение категорий как сокращение дробей, например при умножении «дроби» $[\Phi/\Psi]$ на Ψ (справа) Ψ сокращается со «знаменателем».

Далее будем говорить, что ξ сокращается до η, если существует такая последовательность цепочек категорий $\xi_0, \xi_1, \ldots, \xi_n$ над W, что $\xi_0 = \xi$, $\xi_n = \eta$ и ξ_{i-1} непосредственно сокращается до ξ_i при каждом $i = 1, \ldots, n$.

Теперь мы можем сформулировать определение категориальной грамматики.

Категориальной грамматикой (К-грамматикой) называется упорядоченная четверка $G = \langle V, W, \Phi_0, f\rangle$, где: 1), 2) V и W — конечные множества, называемые соответственно основным словарем и словарем категорий (их элементы называются соответственно основными символами и элементарными категориями); 3) Φ_0 — элемент W, называемый главной категорией; 4) f — функция, ставящая в соответствие каждому основному символу конечное подмножество категориальной системы $K(W)$; f называется приписывающей функцией.

Пусть $x = a_1 \ldots a_k$ — цепочка в V, $a_1, \ldots, a_k \in V$ и $\xi = \Phi_1 \ldots \Phi_k$ — цепочка категорий над W, $\Phi_1, \ldots, \Phi_k \in K(W)$. Мы будем говорить, что грамматика G сопоставляет цепочке x цепочку ξ, если для каждого $i = 1, \ldots, k$ имеет место $\Phi_i \in f(a_i)$; грамматика G приписывает цепочке x категорию Φ, если либо $x = a \in V$ и $\Phi \in f(a)$, либо некоторая цепочка категорий, сопоставленная x, сокращается до Φ.

Языком, определяемым К-грамматикой G (обозначение: $L(G)$), называется множество тех цепочек в основном словаре, которым грамматика G приписывает главную категорию.

Две К-грамматики эквивалентны, если они определяют один и тот же язык.

Приведем несколько примеров К-грамматик. При этом будем явно выписывать только приписывающую функцию, имея в виду, что Φ_0, Φ_1, A, K, T, T_1 обозначают элементарные категории, в том числе Φ_0 — главную категорию. Если значение $f(a)$ приписывающей функции f состоит из одной категории Ψ, то вместо $f(a) = \{\Psi\}$ пишем $f(a) = \Psi$.

Пример 1. Определим приписывающую функцию f_1 на словаре $\{a, b\}$ так: $f_1(a) = \Phi_0$, $f_1(b) = [\Phi_0\backslash\Phi_0]$. Легко видеть, что цепочка, состоящая из таких категорий, сокращается до Φ_0 или равна Φ_0 тогда и только тогда, когда она имеет вид $\Phi_0([\Phi_0\backslash\Phi_0])^n$ $(n = 0, 1, \ldots)$. Поэтому соответствующая грамматика определяет язык $\{ab^n \mid n = 0, 1, \ldots\}$.

Пример 2. Определим функцию f_2 на $\{a, b\}$ так: $f_2(a) = \{A, [A/\Phi_0]\}$, $f_2(b) = [A\backslash\Phi_0]$. Ясно, что всякая цепочка категорий вида $([A/\Phi_0])^{n-1} A([A\backslash\Phi_0])^n$ $(n = 1, 2, \ldots)$ сокращается до Φ_0. С другой стороны, цепочка категорий, являющихся значениями f_2 (точнее, входящих в значения f_2), длина которой равна $2n$, $n = 1, 2, \ldots$, сокращается до Φ_0 только тогда, когда она имеет вид $([A/\Phi_0])^{n-1} A([A\backslash\Phi_0])^n$. (Для доказательства нужно к этому утверждению присоединить еще одно: цепочка категорий, являющихся значениями или частями значений f_2, длина которой равна $2n-1$, $n = 1, 2, \ldots$, сокращается до Φ_0 или равна Φ_0 только тогда, когда она имеет вид $([A/\Phi_0])^{n-1}\Phi_0([A\backslash\Phi_0])^{n-1}$; оба утверждения доказываются одновременной индукцией.) Таким образом, соответствующая грамматика определяет язык $\{a^n b^n \mid n = 1, 2, \ldots\}$.

Пример 3. Определим функцию f_3 на словаре $V = \{p_1, \ldots, p_k, \neg, \&, \vee, \supset, (,)\}$ так: $f_3(p_i) = \Phi_0$ $(i = 1, \ldots, k)$, $f_3(() = K$, $f_3()) = [\Phi_0\backslash[K\backslash\Phi_1]]$, $f_3(\neg) = [\Phi_0/\Phi_1]$, $f_3(\&) = f_3(\vee) = f_3(\supset) = [\Phi_1\backslash[\Phi_0/\Phi_1]]$.

Пусть $F(\Theta)$ — множество тех цепочек, которым такая грамматика приписывает категорию Θ. Тогда: (а) $F(\Phi_0)$ совпадает с множеством всех правильно построенных

формул (ппф) логики высказываний *); (б) $F(\Phi_1) = \{(\mathfrak{A}) \mid \mathfrak{A}$ есть ппф$\}$; (в) $F([\Phi_0/\Phi_1]) = \{\neg\} \cup \{(\mathfrak{A})\alpha \mid \mathfrak{A}$ есть ппф; $\alpha = \&, \vee, \supset\}$; (г) $F(K) = \{(\}$; (д) $F([\Phi_0\backslash[K\backslash\Phi_1]]) = \{)\}$; (е) $F([\Phi_1\backslash[\Phi_0/\Phi_1]]) = \{\&, \vee, \supset\}$.

Утверждения (г), (д), (е) очевидны; (а), (б), (в) доказываются одновременной индукцией по длине цепочки (проведение которой предоставляется читателю).

Пример 4. Определим «терм» в словаре $\{a_1, \ldots, a_k, +, \cdot, (,), =\}$ следующим образом: (i) $a_1, \ldots, a_k$ — термы; (ii) если t_1, t_2 — термы, то $(t_1) + (t_2)$ и $(t_1) \cdot (t_2)$ — термы; (iii) других термов нет. Выражения вида $t_1 = t_2$, где t_1, t_2 — термы, назовем «формулами».

Читателю предоставляется доказать, что множество формул определяется К-грамматикой со следующей приписывающей функцией f_4: $f_4(a_l) = T$, $f_4(+) = f_4(\cdot) = [T_1\backslash[T\backslash T_1]]$, $f_4()) = K$, $f_4()) = [T\backslash[K\backslash T_1]]$, $f_4(=) = [T\backslash[\Phi_0/T]]$. Полезно также найти, аналогично предыдущему примеру, множества $F(\Theta)$.

Дальнейшие «абстрактные» примеры см. в упражнении 6.1. Лингвистический пример см. [Гладкий — Мельчук 1969, стр. 127—131].

Примеры 3 и 4 в какой-то степени поясняют содержательный аспект работы К-грамматик: категория, приписываемая грамматикой той или иной цепочке, отражает «синтаксическую роль» последней, причем эта роль понимается либо как принадлежность к одному из немногочисленных «основных синтаксических классов» (если приписанная категория является элементарной; так, в примере 4 основные классы — «терм», «формула», «левая скобка»), либо «операционно»: «данная цепочка есть оператор, превращающий цепочку такого-то класса в цепочку такого-то класса» (так, в примере 3 категория $[\Phi_0/\Phi_1]$ приписывается унарному оператору $\neg$ и цепочкам вида $(\mathfrak{A})\alpha$, синтаксически также ведущим себя, как унарные операторы, а категория $[\Phi_1\backslash[\Phi_0/\Phi_1]]$ — бинарным операторам $\&, \wedge, \supset$). Аналогичным образом можно поступать для естественных языков; например, непереходный глагол может

*) Имеется в виду следующий вариант определения ппф: (i) $p_1, \ldots, p_k$ суть ппф; (ii) если $\mathfrak{A}$, $\mathfrak{B}$ — ппф и $\alpha = \&, \vee, \supset$, то $\neg(\mathfrak{A})$ и $(\mathfrak{A})\alpha(\mathfrak{B})$ — ппф; (iii) других ппф нет.

рассматриваться как оператор, переводящий существительное в предложение. Подробнее о лингвистическом аспекте К-грамматик см. [Гладкий — Мельчук 1969, стр. 122—136, 150—153].

Перейдем к вопросу о взаимоотношении между К-грамматиками и Б-грамматиками. В одну сторону этот вопрос решается чрезвычайно просто. Пусть $G = \langle V, W, \Phi_0, f\rangle$ — К-грамматика. Рассмотрим Б-грамматику $\Gamma = \langle V, W', \Phi_0, R\rangle$, где W' состоит из всех категорий, являющихся частями элементов значений f *), а R — из всевозможных правил следующих трех видов:

(1) $\Psi \to \Phi[\Phi\backslash\Psi]$, где $[\Phi\backslash\Psi] \in W'$ (тогда и $\Phi, \Psi \in W'$);

(2) $\Psi \to [\Psi/\Phi]\Phi$, где $[\Psi/\Phi] \in W'$;

(3) $\Phi \to a$, где $\Phi \in f(a)$.

Будем говорить, что Г канонически соответствует G. Эквивалентность G и Г, т. е. равенство $L(G) = L(\Gamma)$, очевидна.

Произвольное дерево вывода T в грамматике Г мы будем называть деревом сокращения цепочки $\mu(T)$ в грамматике G. (Разумеется, дерево сокращения можно определить и независимо от Г.) Любой цепочке языка $L(G)$ каждое ее дерево сокращения позволяет хорошо известным нам способом (см. § 3.1) сопоставить размеченную систему составляющих (очевидно, бинарную). В содержательном аспекте эта система обладает любопытной особенностью: среди всевозможных ее иерархизаций естественно выделяются две «главные», наиболее важные для приложений, так что К-грамматика позволяет достаточно «хорошим» способом сопоставлять цепочкам не только системы составляющих, но и деревья подчинения (см. § ПI.3). Эти иерархизации получаются так. Пусть A — неточечная составляющая, «происходящая» от узла дерева сокращения с меткой Φ; тогда из двух непосредственно вложенных в A составляющих B, C одна — пусть B — происходит от узла с меткой $[\Phi/\Psi]$ или $[\Psi\backslash\Phi]$, где Ψ — метка узла, от которого происходит C. При одной из интересующих нас

*) Часть категории Φ определяется так: (α) Φ есть часть Φ; (β) если $\Phi = [\Psi\backslash\Theta]$ или $\Phi = [\Psi/\Theta]$, то Ψ и Θ — части Φ; (γ) если Ψ — часть Φ и Θ — часть Ψ, то Θ — часть Φ; (δ) Ψ может быть частью Φ только в силу (α) (β), (γ).

иерархизаций главной составляющей будет всегда считаться C («аргумент»), при другой B («оператор»). Первую иерархизацию мы назовем прямой, вторую — обратной. Для формальных языков математической логики наиболее адекватной обычно оказывается обратная иерархизация (полезно убедиться в этом для языков примеров 3 и 4, построив для некоторых их цепочек соответствующие деревья подчинения), в то время как для естественных языков предпочтительнее прямая *). Впрочем, для последних еще удобнее может оказаться иерархизация, выбранная более сложным способом.

Возникает вопрос, для всякой ли Б-грамматики можно построить эквивалентную ей К-грамматику. Мы покажем сейчас, что этот вопрос решается положительно; более того, нужная К-грамматика может быть построена так, чтобы значения ее приписывающей функции содержали только категории вида A, $[A\backslash B]$ и $[A\backslash[B\backslash C]]$, где A, B, C — элементарные категории.

Пусть $\Gamma = \langle V, W, I, R\rangle$ — стандартная бинарная Б-грамматика. Построим для нее К-грамматику следующим образом:

1) Каждой упорядоченной паре (A, B), $A, B \in W$, сопоставим новый символ A^B. Положим $Z' = \{A^B \mid A, B \in W\}$ и $Z = Z' \cup \{\Phi_0\}$, где $\Phi_0 \notin Z'$.

2) Чтобы определить приписывающую функцию f, введем сначала вспомогательную функцию f', отображающую W в множество конечных подмножеств $K(Z)$, следующим образом:

а) $\Phi_0 \in f'(I)$.

б) Если $A \to BC \in R$ и $D \in W$, то категории C^A и $[A^D\backslash C^D]$ принадлежат $f'(B)$.

в) Если B, $D \in W$ и Δ — категория, сопоставленная символу D по одному из пунктов а), б), то $[B^D\backslash\Delta] \in f'(B)$.

г) Категория может принадлежать $f'(B)$, $B \in W$, только в силу а), б), в).

*) Это связано прежде всего со следующим обстоятельством. Основное структурное различие между естественными и формально-логическими языками состоит в том, что в первых имеются как предикативные, так и определительные (атрибутивные) связи, тогда как во вторых — только предикативные. Но определение естественно, с одной стороны, считать подчиненным определяемому, с другой — рассматривать как оператор, действующий на определяемое (а не наоборот).

Категории, сопоставленные символу B по пунктам а) и б), мы будем называть левыми B-категориями, сопоставленные по пункту в) — правыми B-категориями. (Одна и та же категория может, вообще говоря, оказаться одновременно левой и правой даже для одного и того же B.)

Функцию f мы определим так: $f(a) = \bigcup\limits_{A\to a\in R} f'(A)$.

Положим теперь $G = \langle V, Z, \Phi_0, f\rangle$. Покажем, что G эквивалентна Г.

Чтобы доказать включение $L(\Gamma) \subseteq L(G)$, достаточно установить, что для всякого дерева полного вывода T в Г найдется такое дерево сокращения T' в G, что $\mu(T') = \mu(T)$. Для этого, в свою очередь, достаточно убедиться в справедливости следующего утверждения:

(А) Пусть D — вывод в Г, начинающийся символом I и заканчивающийся цепочкой X в словаре W, и T — произвольное дерево этого вывода. Тогда найдется такое дерево сокращения T_1 в G с пометкой Φ_0 в корне, что $\mu(T_1)$ есть (некоторая) цепочка, сопоставленная цепочке X с помощью функции f'.

Утверждение (А) мы докажем индукцией по числу узлов в дереве T. Для удобства проведения индукции будем доказывать несколько более сильное предложение. Чтобы его сформулировать, введем понятие левого и правого узлов бинарного дерева. Именно, если узлы α, β, γ расположены следующим образом:

то мы называем узел β левым и γ — правым. Кроме того, корень считается левым узлом.

Добавим теперь к (А) следующее требование: если α — левый (правый) висячий узел T, помеченный символом B, то соответствующий висячий узел T_1 должен быть помечен левой (соответственно правой) B-категорией (иначе: если вхождение символа B в X отвечает левому (правому) узлу T, то соответствующее вхождение символа в $\mu(T_1)$ должно быть вхождением левой (правой) B-категории). Полученное таким образом предложение обозначим (А′). Докажем его.

I. Если T — единичное дерево, (A′) очевидно.

II. Пусть (A′) доказано для деревьев, имеющих менее n узлов, и T — дерево с n узлами ($n > 1$). Рассмотрим некоторый невисячий левый узел α дерева T, обладающий тем свойством, что в полном α-поддереве τ дерева T все левые узлы, кроме самого α, являются висячими (так что это поддерево имеет вид, показанный на рис. 12, *а*)); такой узел обязательно существует *).

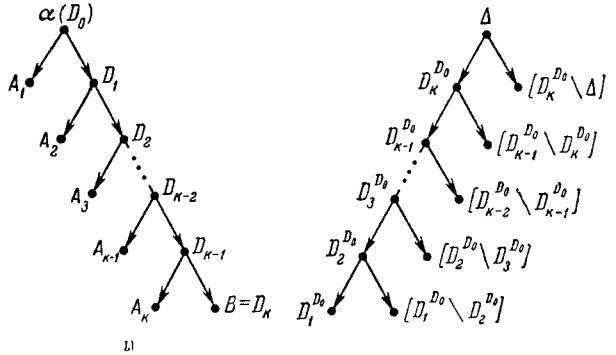


Рис. 12.

Положим $T' = \mathrm{Sub}(T, \tau, \alpha)$. Дерево T' имеет меньше n узлов, и α — висячий левый узел T'. Если $\mu(T') = X'$, $\mu(\tau) = Z$ и метка при α есть D_0, то, очевидно, X и X' могут быть представлены в виде $X = Y_1 Z Y_2$, $X' = Y_1 D_0 Y_2$. По индуктивному предположению найдется такое дерево сокращения T_1' в грамматике G, что $\mu(T_1')$ есть цепочка, сопоставляемая цепочке X' функцией f', и при этом для каждого вхождения символа B в X соответствующее вхождение в $\mu(T_1')$ является вхождением левой или правой B-категории в зависимости от того, какому узлу T' отвечает данное вхождение B в X'. В частности, вхождению D_0, отвечающему узлу α дерева T', соответствует вхождение в $\mu(T_1')$ левой D_0-категории Δ, отвечающее некоторому висячему узлу α' дерева T', причем $\mu(T_1') = U_1 \Delta U_2$, где U_1 и U_2 — цепочки категорий, сопоставляемые функцией f' цепочкам Y_1 и Y_2 соответственно. Пусть теперь $Z = A_1 \ldots A_k B$ ($A_1, \ldots, A_k$, $B \in W$). Вхождения $A_1, \ldots, A_k$ отвечают здесь левым узлам дерева τ (а значит, и дерева T'), вхождение B — правому узлу. Пусть, кроме того, $D_1, D_2, \ldots, D_{k-1}$, $D_k = B$ — метки при правых узлах τ, упорядоченных сверху вниз (см. рис. 12, *а*)). Тогда схема R содержит правила $D_0 \to A_1 D_1$, $D_1 \to A_2 D_2$, …, $D_{k-1} \to A_k D_k$. Обозначим через τ' P_1-дерево, изображенное на рис. 12, *б*). Имеем $\mu(\tau') = D_1^{D_0}[D_1^{D_0} \backslash D_2^{D_0}] \ldots [D_{k-1}^{D_0} \backslash D_k^{D_0}][D_k^{D_0} \backslash \Delta]$, так что $\mu(\tau')$ сокращается до Δ, и при этом дерево $T_1 = \mathrm{Sub}(T_1', \alpha', \tau')$ является, очевидно, деревом сокращения цепочки категорий $U = U_1 \mu(\tau') U_2$; сверх того, $U = \mu(T_1)$ есть цепочка, сопоставляемая функцией f' цепочке X (поскольку $\mu(\tau')$ сопоставляется, как сразу видно из определения f', цепочке Z), и каждому вхождению символа B в X соответствует вхождение в U левой (правой) B-категории, если данное вхождение B в X отвечает левому (правому) узлу T (действительно, для вхождений символов в U_1 и U_2 это верно по индуктивному предположению, а для вхождений в Z ясно из построения). Итак, (A′) доказано.

*) В самом деле, если корень β дерева T не обладает нужным свойством, то, идя из β по правым узлам, мы непременно придем к такому узлу γ (возможно, $\gamma = \beta$), что подчиненный ему левый узел δ не является в T висячим. Если δ не обладает нужным свойством, то, идя из δ по правым узлам, мы опять-таки придем к такому узлу, что подчиненный ему левый узел не является в T висячим, и т. д. Но этот процесс не может быть бесконечным.

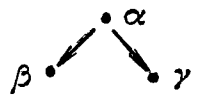
Докажем теперь включение $L(G) \subseteq L(\Gamma)$. Аналогично предыдущему для этого достаточно установить справедливость следующего предложения:

(Б) Если для некоторой цепочки $X \in V_1^+$ найдется дерево сокращения T' в G с меткой Φ_0 в корне такое, что $\mu(T')$ сопоставляется цепочке X функцией f', то либо $X = I$, либо X выводима из I в Γ.

Докажем (Б) индукцией по $|X|$.

I. Если $|X| = 1$, то $X = I$.

II. Пусть (Б) доказано для всех цепочек длины, меньшей n, $|X| = n (n > 1)$, и T' — соответствующее дерево сокращения. Если узлы α, β, γ дерева T' расположены так:



и Φ, Ψ, Θ — метки при α, β, γ соответственно, то $\Psi\Theta$ непосредственно сокращается до Φ; в этом сокращении Ψ является «аргументом», а Θ — «оператором» (поскольку метки при узлах T' не содержат символа /), так что $\Theta = [\Psi\backslash\Phi]$, и Ψ — элементарная категория («знаменатели» всех категорий, входящих в значения f', элементарны). Таким образом, все левые узлы T' помечены элементарными категориями, а все правые — неэлементарными.

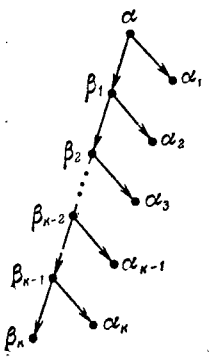


Рис. 13.

Рассмотрим невисячий узел α дерева T' такой, что а) α — правый узел или совпадает с корнем T'; б) в полном α-поддереве τ' дерева T' все правые узлы, отличные от α, висячие; такой узел обязательно существует (аналогично предыдущему). Пусть β_1, β_2, ..., β_k — все отличные от α левые узлы τ', упорядоченные сверху вниз, и $\alpha = \alpha_0, \alpha_1, \ldots, \alpha_k$ — все остальные узлы τ', упорядоченные так же (рис. 13). Обозначим через Ψ_i, Θ_j метки при α_i, β_j соответственно и через B, A_k, ..., A_1 — те символы из V_1, вхождениям которых в X соответствуют вхождения Θ_k, Ψ_k, ..., Ψ_1 в $u_t(T')$, отвечающие узлам β_k, α_k, ..., α_1 дерева T' (в этом порядке). Тогда X представляется в виде $X = Y_1BA_k \ldots A_1Y_2$. Категории Ψ_1, ..., Ψ_k, Θ_k принадлежат $f'(A_1)$, ..., $f'(A_k)$, $f'(B)$ соответственно. По предыдущему, категории Θ_1, ..., Θ_k элементарны, а Ψ_1, ..., Ψ_k неэлементарны; категория Ψ_0 либо неэлементарна, либо равна Φ_0; кроме того, $\Psi_1 = [\Theta_1\backslash\Psi_0]$ и $\Psi_i = [\Theta_i\backslash\Theta_{i-1}]$ при $i = 2, \ldots, k$. Заметим, что Ψ_1 — правая A_1-категория, так как «числитель» неэлементарной левой категории всегда является элементарной категорией, отличной от Φ_0.

Обозначим теперь через l наибольшее из чисел $i, \ldots, k$, для которого Ψ_l есть правая A_l-категория, и через τ'' — полное β_{l-1}-поддерево T' (если $l = 1$, считаем $\beta_{l-1} = \alpha$). Положим $T_1' = \mathrm{Sub}(T', \tau'', \beta_{l-1})$. «Числитель» правой категории — левая категория; поэтому найдется такое $D \in W$, что Θ_{l-1} является левой D-категорией. Если положить $X_1 = Y_1DA_{l-1} \ldots A_1Y_2$, (при $l = 1$ полагаем $X_1 = Y_1DY_2$), то $u_t(T_1')$ будет цепочкой категорий, сопоставляемой цепочке X_1 функцией f'. Отсюда, по индуктивному предположению, следует, что X_1 выводима из I в Г. Поэтому для завершения доказательства достаточно установить, что цепочка $BA_kA_{k-1} \ldots A_l$ выводима в Г из D.

Поскольку Θ_k — элементарная B-категория, а Ψ_k, Ψ_{k-1}, ..., Ψ_{l+1} — неэлементарные левые A_k-, A_{k-1}-, ..., A_{l+1}-категории, они имеют соответственно вид C^F, $\left[B_k^{F_k}\backslash C_k^{F_k}\right]$, $\left[B_{k-1}^{F_{k-1}}\backslash C_{k-1}^{F_{k-1}}\right]$, ..., $\left[B_{l+1}^{F_{l+1}}\backslash C_{l+1}^{F_{l+1}}\right]$, где C, F, B_i, C_i, $F_i \in W$. Но из равенства $\Psi_k = [\Theta_k\backslash\Theta_{k-1}]$ получаем $C^F = B_k^{F_k}$. Далее из того же равенства вместе с равенством $\Psi_{k-1} = [\Theta_{k-1}\backslash\Theta_{k-2}]$ следует $C_k^{F_k} = B_{k-1}^{F_{k-1}}$ и т. д. Имеем, таким образом: $\Theta_k = B_k^F$, $\Psi_k = \left[B_k^F\backslash B_{k-1}^F\right]$, $\Psi_{k-1} = \left[B_{k-1}^F\backslash B_{k-2}^F\right]$, ..., $\Psi_{l+1} = \left[B_{l+1}^F\backslash C_{l+1}^F\right]$. Отсюда, поскольку Θ_k — B-категория и Ψ_l — A_l-категория, вытекает наличие в R правил $F \to BB_k$, $B_k \to A_kB_{k-1}$, $B_{k-1} \to A_{k-1}B_{k-2}$, ..., $B_{l+1} \to A_{l+1}C_{l+1}$, что дает $F \underset{\Gamma}{\vdash} BA_kA_{k-1} \ldots A_{l+1}C_{l+1}$.

В то же время $\Psi_l = [\Theta_l\backslash\Theta_{l-1}] = \left[C_{l+1}^F\backslash\Theta_{l-1}\right]$, и Ψ_l — правая A_l-категория, а Θ_{l-1} — левая D-категория; но это в силу определения правой A_l-категории дает $C_{l+1} = A_l$ и $F = D$.

Итак, $D \underset{\Gamma}{\vdash} BA_kA_{k-1} \ldots A_l$. Доказательство закончено.

Подведем итог. Будем называть сложностью категории общее число вхождений в нее символов \ и / и сложностью К-грамматики — максимальную сложность категорий, входящих в значения ее приписывающей функции. Кроме того, назовем К-грамматику левосторонней (правосторонней), если категории, входящие в значения ее приписывающей функции, не содержат вхождений символа / (соответственно \). В последнем рассуждении мы имели дело с левосторонней грамматикой сложности, не превосходящей 2; ничто не помешало бы нам, разумеется, повторить это рассуждение для правосторонней грамматики. Суммируя, получаем следующую теорему.

Теорема 6.1. а) *Для всякой* К-*грамматики можно построить эквивалентную ей* Б-*грамматику.* б) *Для вся-*

кой Б-грамматики можно построить эквивалентную ей левостороннюю (правостороннюю) К-грамматику сложности, не превосходящей 2.

Следствие. *Для всякой К-грамматики можно построить эквивалетную ей левостороннюю (правостороннюю) К-грамматику сложности, не превосходящей 2.*

Замечания. 1. В теореме 6.1, б) нельзя заменить двойку единицей (упражнение 6.4); это невозможно и при отказе от односторонности (упражнение 6.5).

2. Назовем путь $\alpha_0, \alpha_1, \ldots, \alpha_s$ в бинарном дереве левым (правым), если узлы $\alpha_1, \ldots, \alpha_s$ левые (правые). Если $\alpha_0, \alpha_1, \ldots, \alpha_s$ — правый путь в дереве сокращения левосторонней К-грамматики и узел α_0 помечен категорией Ψ_0, то метка при α_1 будет иметь вид $[\Psi_1 \backslash \Psi_0]$, метка при α_2 — вид $[\Psi_2 \backslash [\Psi_1 \backslash \Psi_0]]$ и т. д. Поэтому длина правого пути в дереве сокращения левосторонней К-грамматики не может превосходить сложности этой грамматики; аналогично для левых путей в деревьях сокращения правосторонних грамматик. Отсюда по теореме 6.1 получаем, переходя к канонически соответствующей Б-грамматике, что для всякой Б-грамматики Γ можно построить эквивалентную ей стандартную бинарную Б-грамматику Γ' такую, что длины правых (левых) путей в деревьях полных выводов в Γ' не превосходят 2.

§ 6.2. Нормальная форма Б-грамматики

Замечание 2 к теореме 6.1 мы используем для доказательства следующей теоремы о Б-грамматиках.

Теорема 6.2 (теорема о нормальной форме). *Для всякой Б-грамматики можно построить эквивалентную ей Б-грамматику, все правила которой имеют один из следующих видов*: $A \to a$, $A \to aB$, $A \to aBC$, *где* A, B, C — *вспомогательные символы и* a — *основной символ.*

Доказательство. Пусть $\Gamma^0 = \langle V^0, W^0, I^0, R^0 \rangle$ — стандартная бинарная Б-грамматика. Будем сопоставлять символам из W^0 (не обязательно всем) числа, которые назовем их левыми степенями, следующим образом: (i) левая степень считается равной нулю для тех $A \in W^0$, для которых не существует правил вида $A \to BC$ (так что всякое правило с левой частью A имеет вид $A \to a$, где $a \in V^0$); (ii) если для каждого $k = 0, \ldots$

$\ldots, n-1$ определено понятие символа левой степени k, то символу $A \in W^0$ приписывается левая степень n, если ему не приписано в качестве левой степени ни одно из чисел $0, \ldots, n-1$ и в каждом правиле вида $A \to BC$, $B, C \in W^0$, символ B имеет левую степень, меньшую n.

Если всем символам из W^0 можно приписать левые степени и наибольшая из них равна N, будем называть Γ^0 грамматикой левой степени N. Ясно, что если Γ^0 — приведенная грамматика, то она имеет левую степень N тогда и только тогда, когда N есть точная верхняя граница длин левых путей в деревьях полных выводов в Γ^0.

Пусть теперь Γ — произвольная Б-грамматика с основным словарем V. По замечанию 2 к теореме 6.1 можно построить эквивалентную Γ стандартную бинарную Б-грамматику $\Gamma^1 = \langle V, W^1, I^1, R^1 \rangle$ такую, что длины левых путей в деревьях полных выводов в Γ^1 не превосходят 2. Перейдем от Γ^1 к эквивалентной приведенной грамматике $\Gamma^0 = \langle V, W^0, I^0, R^0 \rangle$, где $W^0 \subseteq W^1$ и $R^0 \subseteq R^1$ (лемма 4.8). При этом множество деревьев полных выводов не изменится, и по предыдущему Γ^0 будет грамматикой левой степени, не превосходящей 2.

Пусть $r = A \to BC \in R^0$. Тогда левая степень символа B равна 0 или 1. Сопоставим правилу r всевозможные правила вида $A \to bC$, где $b \in V$ и $B \to b \in R^0$, а если левая степень B равна 1, — еще и всевозможные правила вида $A \to b_1 B_2 C$, где b_1 и B_2 удовлетворяют условию

$$b_1 \in V \,\&\, B_2 \in W^0 \,\&\, \&\, (\exists B_1 \in W^0)[B \to B_1 B_2 \in R^0 \,\&\, B_1 \to b_1 \in R^0]$$

Если заменить в Γ^0 каждое правило вида $A \to BC$ совокупностью сопоставленных ему новых правил, то полученная грамматика будет, очевидно, эквивалентна Γ^0, а следовательно, и Γ. Доказательство закончено.

Грамматика, удовлетворяющая требованию теоремы 6.2, называется Б-грамматикой в нормальной форме Грейбах (или просто в нормальной форме).

Теорему 6.2 можно усилить следующим образом.

Теорема 6.3. *Для всякой* Б-*грамматики* Γ *и всякого целого положительного числа* s *можно построить* Б-*грамматику* $\Gamma' = \langle V, W', I', R' \rangle$, *эквивалентную* Γ, *все правила которой имеют один из следующих видов*:

$A \to x$, $A \to yB$, $A \to yBC$, *где* $A, B, C \in W'$,

$$x, y \in V^+, |x| \leqslant s, |y| = s.$$

Доказательство. Пусть $\Gamma = \langle V, W, I, R \rangle$ — Б-грамматика в нормальной форме. Если $A \in W$, $x \in V^+$, $|x| = s$, $X \in W^*$ и цепочка xX выводима в Γ из A, то $|X| \leqslant s+1$. Сопоставим каждой цепочке $X \in W^+$, $|X| \leqslant s+1$, новый символ $\beta(X)$; совокупность всех таких символов обозначим W'. Пусть $\Gamma' = \langle V, W', \beta(I), R' \rangle$, где R' состоит из всевозможных правил следующих видов:

1) $\beta(A_1 \dots A_k) \to x$, где $1 \leqslant k \leqslant s+1$; $|x| \leqslant s$; $A_1 \dots A_k \underset{\Gamma}{\vdash} x$;

2) $\beta(A_1 \dots A_k) \to y\beta(A_{i+1} \dots A_k)$, где $1 \leqslant i < k \leqslant s+1$; $|y| = s$; $A_1 \dots A_i \underset{\Gamma}{\vdash} y$;

3) $\beta(A_1 \dots A_k) \to uv\beta(Z)\beta(A_{i+1} \dots A_k)$, где $1 \leqslant i < k \leqslant s+1$; $|uv| = s$; $Z \in W^+$; $A_1 \dots A_{i-1} \underset{\Gamma}{\vdash} u$; $A_i \underset{\Gamma}{\vdash} vZ$;

4) $\beta(A_1 \dots A_k) \to uv\beta(Z)$, где $1 \leqslant k \leqslant s+1$; $|uv| = s$; $Z \in W^+$; $A_1 \dots A_{k-1} \underset{\Gamma}{\vdash} u$; $A_k \underset{\Gamma}{\vdash} vZ$.

Эти правила строятся по R эффективно. Легко видеть, что Γ' эквивалентна Γ.

Теорему 6.3 мы используем для доказательства следующего утверждения о МП-машинах.

Теорема 6.4. *Для всякой* МП-*машины* M *можно построить эквивалентную ей* МП-*машину* M', *обладающую тем свойством, что ни в одном ее вычислении ни разу не выполняются подряд два элементарных шага на рабочей ленте* (*иначе говоря, между каждыми двумя «рабочими» шагами читается хоть один входной символ*).

Доказательство. При построении машины M' мы можем отправляться не от МП-машины M, а от Б-грамматики $\Gamma = \langle V, W, I, R \rangle$ и при этом имеем право считать, что Γ удовлетворяет требованию теоремы 6.3 при $s = 3$. Входным и рабочим алфавитами M' будут V и W соответственно. Программа M' будет представлять собой объединение систем инструкций, сопоставляемых

правилам Γ указываемым далее способом, с добавлением инструкций $q_1\# \to q$, $q\# \to q_0$. Сопоставление правилам систем инструкций производится следующим образом:

1) правилу вида $r = A \to a\,(a \in V)$ сопоставляется система $\{qa \to q_1(r),\ Aq_1(r) \to q\}$;

2) правилу вида $A \to a_1a_2\,(a_1, a_2 \in V)$ сопоставляется система $\{qa_1 \to q_1(r),\ q_1(r)\,a_2 \to q_2(r),\ Aq_2(r) \to q\}$;

3) правилу вида $A \to a_1a_2a_3\,(a_1, a_2, a_3 \in V)$ сопоставляется система $\{qa_1 \to q_1(r),\ q_1(r)\,a_2 \to q_2(r),\ q_2(r)\,a_3 \to q_3(r),\ Aq_3(r) \to q\}$;

4) правилу вида $A \to a_1a_2a_3B\,(a_1, a_2, a_3, \in V;\ B \in W)$ сопоставляется система $\{qa_1 \to q_1(r),\ Aq_1(r) \to q_2(r),\ q_2(r)\,a_2 \to q_3(r),\ q_3(r) \to Bq_4(r),\ q_4(r)\,a_3 \to q\}$;

5) правилу вида $A \to a_1a_2a_3BC\,(a_1, a_2, a_3 \in V;\ B, C \in W)$ сопоставляется система $\{qa_1 \to q_1(r),\ Aq_1(r) \to q_2(r),\ q_2(r)\,a_2 \to q_3(r),\ q_3(r) \to Cq_4(r),\ q_4(r)\,a_3 \to q_5(r),\ q_5(r) \to Bq\}$.

Здесь $q_i(r)$ — состояния, различные при разных i и при разных r. Единственным заключительным состоянием является q_0.

Из самой конструкции ясно, что каждое вычисление машины M' удовлетворяет нужному условию. Эквивалентность Γ и M' доказывается без труда; мы предоставляем это читателю. [Указание. Между вычислениями M' и упорядоченными размеченными выводами в Γ устанавливается соответствие таким образом, что каждой промежуточной цепочке вывода, имеющей вид xX, где $x \in V^*$, $X \in W^*$, отвечает участвующая в вычислении ситуация, при которой прочитанная часть входной цепочки есть x, а на рабочей ленте записана цепочка X.]

Замечания. 1. Теорему 6.4 нетрудно было бы усилить, потребовав, чтобы между каждыми двумя «рабочими» шагами читалось не менее s входных символов, где s — произвольное фиксированное натуральное число.

2. Машина, удовлетворяющая требованию теоремы 6.4, является Э-машиной без растяжения (§ 3.2). Таким образом, иерархии основных классов грамматик: $\Gamma \supset$ НС $\supset$ Б $\supset$ А отвечает иерархия классов Э-машин: произвольные Э-машины $\supset$ Э-машины без растяжения $\supset$ МП-машины без растяжения $\supset$ конечные автоматы. Можно было бы включить в эту иерархию также и соответствующие машины для классов языков, рассмотренных в §§ 5.3 и 5.4 (упражнение 6.9).

§ 6.3. Доминационные грамматики

Для лингвистических приложений полезно иметь возможность задавать язык таким образом, чтобы каждой его цепочке сопоставлялось описание ее синтаксической структуры в виде дерева подчинения, подобно тому, как с помощью НС-грамматики цепочкам сопоставляются описания в виде систем составляющих. Ввиду наличия простой связи между системами составляющих и проективными деревьями подчинения (§ ПI.3) для этой цели можно использовать те же НС-грамматики, некоторым образом «иерархизуя» их, так, чтобы для каждой системы составляющих, которую «выдает» данная грамматика, автоматически указывалась некоторая иерархизация и тем самым определялось дерево подчинения. Примером могут служить упоминавшиеся в § 6.1 прямая и обратная иерархизации категориальных грамматик. Сейчас мы проведем соответствующие построения в общем виде, ограничиваясь, впрочем, случаем Б-грамматик (о распространении на произвольные НС-грамматики см. упражнение 6.18).

Будем называть доминационной грамматикой (Д-грамматикой) упорядоченную пару $G = (\Gamma, f)$, где $\Gamma = \langle V, W, I, R\rangle$ — Б-грамматика без правил вида $A \to B$, $A, B \in W$, и $A \to a$, $A \in W - \{I\}$, $a \in V$ (такие Б-грамматики будут в дальнейшем называться удлиняющими), и f — отображение, сопоставляющее каждому правилу $r = A \to \omega \in R$, кроме правил вида $I \to a$, $a \in V$, некоторое непустое подмножество $f(r)$ множества $\{1, \ldots, |\omega|\}$.

Если $r = A \to \varphi\alpha\psi \in R$, $\alpha \in V \cup W$ и $|\varphi\alpha| \in f(r)$, мы будем называть вхождение $\varphi * \alpha * \psi$ символа α в правую часть r отмеченным.

Пусть T — дерево полного вывода в Г и C — соответствующая система составляющих; поскольку грамматика Г удлиняющая, дерево T, рассматриваемое как дерево без меток, изоморфно системе C. Пусть далее α — произвольный невисячий узел T и $A \to \omega$ — правило, применяемое на шаге вывода, отвечающем узлу α. Назовем отмеченными те из подчиненных α узлов, которые соответствуют отмеченным вхождениям символов в ω.

Будем называть иерархизацию C' системы C допустимой (для G), если все главные (т. е. принадлежащие C') составляющие отвечают отмеченным узлам T. По теореме ПI.2 существует единственное связанное с (C, C') (и определяемое по (C, C') эффективно) дерево подчинения D для цепочки $x = y(T)$; мы скажем, что D сопоставляется цепочке x грамматикой G.

Замечание. Возможен другой способ определения Д-грамматик, при котором они трактуются как частный случай удлиняющих Б-грамматик. Для этого нужно: а) ввести для каждого символа $\alpha \in V \cup W$ «двойника» — новый вспомогательный символ α'; б) заменить каждое правило $r \neq I \to a (a \in V)$ системой правил, отличающихся от r тем, что в правой части каждого нового правила одно вхождение символа отмечается штрихом, а также добавить правила $\alpha' \to \alpha$ для всех $\alpha \in V \cup W$. Каждому дереву выводу в такой грамматике отвечает единственная иерархизованная система составляющих: главными составляющими в ней будут те, которые соответствуют узлам, помеченным штрихованными символами.

Будем говорить, что Д-грамматики $G_1 = (\Gamma_1, f_1)$ и $G_2 = (\Gamma_2, f_2)$ сильно эквивалентны, если Γ_1 эквивалентна Γ_2 и при этом дерево подчинения тогда и только тогда сопоставляется цепочке грамматикой G_1, когда оно сопоставляется той же цепочке грамматикой G_2. Скажем, кроме того, что Б-грамматика Γ_0 и Д-грамматика $G = (\Gamma, f)$ слабо эквивалентны, если Γ_0 эквивалентна Г, и сильно эквивалентны, если они слабо эквивалентны и при этом: а) для всякой системы составляющих C, отвечающей полному выводу какой-либо цепочки в Г, найдется дерево подчинения, сопоставляемое той же цепочке грамматикой G и согласованное с C; б) для всякого дерева подчинения T, сопоставляемого какой-либо цепочке грамматикой G, найдется система составляющих, отвечающая некоторому полному выводу той же цепочки в Γ_0 и согласованная с T. Будем говорить, наконец, что Б-грамматика Г вкладывается в Д-грамматику (Γ, f).

Мы рассмотрим теперь один специальный класс Д-грамматик, представляющий особый интерес (в частности, для приложений).

Будем называть Д-грамматику простой*), если во всех ее правилах отмечены только вхождения основных символов.

Если Д-грамматика (Γ, f) простая, то во всех строящихся с ее помощью иерархизованных системах составляющих главными составляющими могут быть только точечные. Для случая, когда Γ — приведенная грамматика, справедливо и обратное.

Понятие простой Д-грамматики естественно обобщается следующим образом. Пусть $\Gamma = (\langle V, W, I, R\rangle, f)$ — Д-грамматика. Введем на множестве $V \cup W$ бинарное отношение $\mathscr{R}$ следующим образом: $\alpha \mathscr{R} \beta$ означает, что в некотором правиле Γ с левой частью α правая часть содержит отмеченное вхождение β. Если граф $\langle V \cup W; \mathscr{R}\rangle$ не содержит циклов (замкнутых путей ненулевой длины), будем называть G Д-грамматикой без циклов.

Лемма 6.1. *Для всякой Д-грамматики без циклов можно построить сильно эквивалентную ей простую Д-грамматику.*

Доказательство. Пусть h — наибольшая длина пути в графе $\langle V \cup W; \mathscr{R}\rangle$ для данной Д-грамматики $G = (\Gamma, f)$, где $\Gamma = \langle V, W, I, R\rangle$. Б-грамматику Γ мы можем считать приведенной. (Действительно, если $\Gamma' = \langle V, W', I, R'\rangle$ — грамматика, соответствующая Γ по лемме 4.8, и f' — функция, которую f индуцирует на R', то Д-грамматика (Γ', f) сильно эквивалентна G, а ее граф $\langle V \cup W'; \mathscr{R}'\rangle$ является подграфом исходного графа $\langle V \cup W; \mathscr{R}\rangle$.) Поэтому при $h = 1$ грамматика G простая. Пусть $h > 1$ и утверждение доказано для всех $h' < h$. Назовем рангом вспомогательного символа A наибольшую длину пути в графе $\langle V \cup W; \mathscr{R}\rangle$ с началом A. Ясно, что правые части правил Γ, имеющих левыми частями символы ранга 1, являются цепочками в V. Заменим теперь каждое правило, содержащее в правой части вхождения символов ранга 1, правилом, получающимся из данного путем замены всех вхождений символов ранга 1 в правую часть вхождениями цепочек в V, непосредственно выводимых из этих символов.

*) В литературе для простых Д-грамматик часто используется термин «грамматики зависимостей» (англ. dependence grammars).

Функцию f переопределим на новых правилах так, чтобы для каждого такого правила отмеченными вхождениями были, во-первых, те, которые были таковыми в старом правиле и не были заменены при переходе к новому, и, во-вторых, те, которые возникают из отмеченных вхождений символов в правые части правил вида $A \to x$, $x \in V^+$, при подстановке x вместо отмеченного вхождения A в правую часть старого правила. Построенная таким образом новая Д-грамматика будет, очевидно, сильно эквивалентна G, и в то же время каждый максимальный путь в графе $\langle V \cup W; \mathscr{R}\rangle$ при переходе к новой грамматике укорачивается на единицу; остается воспользоваться индуктивным предположением.

Будем далее называть Д-грамматику $G = (\Gamma, f)$ Д-грамматикой ограниченной ширины, если существует такое число k, что ширина дерева подчинения, сопоставляемого грамматикой G цепочке из $L(\Gamma)$, не может превосходить этого k. Наименьшее из таких чисел будет называться шириной грамматики G.

Лемма 6.2. а) *Если $G = (\Gamma, f)$ — Д-грамматика без циклов, $\Gamma = \langle V, W, I, R\rangle$ и наибольшая длина пути в графе $\langle V \cup W; \mathscr{R}\rangle$ равна h, то G есть грамматика ограниченной ширины, и ширина ее не превосходит $h \cdot (g - 1)$, где g — максимум длин правых частей правил Γ. (В частности, ширина простой Д-грамматики не превосходит $g - 1$.)*

б) *Если Γ — приведенная Б-грамматика и $G = (\Gamma, f)$ — Д-грамматика ограниченной ширины, то G не имеет циклов.*

в) *Если в условиях пункта* а) *грамматика Γ приведенная, то ширина G не меньше h.*

Доказательство. Назовем путь в дереве полного вывода в Γ отмеченным, если все его узлы, кроме, быть может, начального, отмечены. Для дальнейшего заметим, что в произвольном дереве полного вывода из любого узла идет хотя бы один отмеченный путь в висячий узел. Пусть $\alpha_0, \ldots, \alpha_n$ — отмеченный путь в дереве полного вывода, причем узел α_n висячий, и $A_0, \ldots, A_n$ — соответствующая последовательность составляющих. Вспоминая указанный в доказательстве теоремы П1.2 способ построения дерева подчинения,

связанного с данной иерархизованной системой составляющих, мы видим, что если при какой-то иерархизации все составляющие $A_1, \ldots, A_n$ окажутся главными, а A_0 — не главной, то узлу соответствующего дерева подчинения, являющемуся главной точкой A_0, будут подчинены все узлы, являющиеся главными точками не главных составляющих, непосредственно вложенных в $A_0, A_1, \ldots, A_{n-1}$, и только они. Но число таких узлов не меньше n и не больше $n\cdot(g-1)$. Если G — грамматика без циклов и наибольшая длина пути в графе $\langle V \cup W; \mathscr{R}\rangle$ есть h, то длина отмеченного пути в дереве вывода не может быть больше h, и, следовательно, никакой узел дерева подчинения не может подчинять более $h\cdot(g-1)$ узлов. (Вспомним, что всякая точка цепочки главная для некоторой не главной составляющей.) В то же время из самого определения отношения $\mathscr{R}$ следует, что хотя бы в одном дереве вывода должен быть хотя бы один отмеченный путь длины h, идущий в висячий узел; а если грамматика Г приведенная, то любое дерево вывода можно вложить в дерево полного вывода, так что, по предыдущему, в некотором дереве подчинения, сопоставляемом какой-либо цепочке грамматикой G, найдется узел, подчиняющий не менее h узлов. Теперь нам достаточно показать, что если Г — приведенная Б-грамматика и в графе $\langle V \cup W; \mathscr{R}\rangle$ есть циклы, то в деревьях полных выводов найдутся сколь угодно длинные отмеченные пути. Пусть символ A принадлежит какому-либо циклу указанного графа. Тогда для некоторой цепочки $\omega \in (V \cup W)^+$ найдется (A, ω)-дерево T_1 в Г, в котором из корня идет отмеченный путь в висячий узел, помеченный тем же символом A. Для $n = 2, 3, \ldots$ положим $T_n = \mathrm{Sub}(T_{n-1}, T_0, T_1)$, где T_0 — единичное дерево с меткой A в единственном узле. При достаточно большом n дерево T_n будет содержать сколь угодно длинный отмеченный путь из корня в висячий узел с меткой A. Поскольку грамматика Г приведенная, найдется дерево полного вывода T, в котором некоторый узел α имеет метку A. По предыдущему в полном α-поддереве T' дерева T из корня исходит хотя бы один отмеченный путь. Полагая $T'_n = \mathrm{Sub}(T_n, T_0, T')$, видим, что в T'_n из корня идет отмеченный путь, более длинный, чем в T_n, уже в узел, помеченный основным символом; но ввиду приведенности Г дерево T'_n можно «достроить» до дерева полного вывода.

Из лемм 6.1 и 6.2 и из того очевидного обстоятельства, что свойство ограниченности ширины сохраняется при сильной эквивалентности, немедленно вытекает

Теорема 6.5. а) *Для всякой Д-грамматики ограниченной ширины можно построить сильно эквивалентную ей простую Д-грамматику.* б) *Если для Д-грамматики G существует сильно эквивалентная ей простая Д-грамматика, то G имеет ограниченную ширину.*

Посмотрим теперь, для каких Б-грамматик существуют эквивалентные в том или ином смысле простые Д-грамматики, или, что то же самое, Д-грамматики ограниченной ширины.

Ясно, прежде всего, что удлиняющая Б-грамматика тогда и только тогда вкладывается в простую Д-грамматику (и притом эффективно*)), когда правая часть каждого ее правила содержит вхождения основных символов. Поэтому из теоремы 6.2 следует, что для всякой Б-грамматики можно построить слабо эквивалентную ей простую Д-грамматику.

Чтобы исследовать вопрос о сильной эквивалентности, введем следующее понятие.

Пусть $\Gamma = \langle V, W, I, R\rangle$ — Б-грамматика. Будем сопоставлять символам из $V \cup W$ числа, которые назовем их степенями, следующим образом: (i) степени основных символов, и только их, равны нулю; (ii) пусть для каждого $k = 0, \ldots, n-1$ определено понятие символа степени k; тогда символу A приписывается степень n, если ему не приписано в качестве степени ни одно из чисел $0, \ldots, n-1$ и правая часть каждого правила с левой частью A содержит хотя бы одно вхождение символа степени, не превосходящей $n-1$. Символы, которым с помощью такой процедуры никакая степень не приписывается, считаем имеющими бесконечную степень. Наибольшая из степеней символов полного словаря Г будет называться степенью грамматики Г.

*) То есть для Б-грамматики Г можно построить простую Д-грамматику вида (Γ, f).

Ясно, что степень каждого символа, а значит, и степень грамматики можно найти эффективно. Так, грамматики со схемами

$$\{I \to CI,\ I \to aCb,\ I \to ab,\ C \to aCb, C \to ab\},$$

$$\{I \to CI,\ I \to aCb,\ I \to ab,\ C \to ACB,\ C \to ab,$$

$$C \to aabb,\ A \to aa,\ B \to bb\} \text{ и}$$

$$\{I \to II,\ I \to aIb,\ I \to ab\}$$

имеют соответственно степени 2, 3 и ∞ (хотя все они эквивалентны).

Легко заметить (ср. доказательство теоремы 6.2), что если Г — приведенная удлиняющая Б-грамматика, то ее степень равна точной верхней границе степеней систем составляющих, отвечающих деревьям полных выводов в Г. Поэтому, если назвать Б-грамматики Γ_1 и Γ_2 сильно эквивалентными в случае, когда они эквивалентны и множества систем составляющих, отвечающих деревьям полных выводов в Γ_1 и Γ_2, совпадают, то будет справедлива

Лемма 6.3. *Если две приведенные удлиняющие* Б-*грамматики сильно эквивалентны, то их степени равны.*

Нам понадобится также

Лемма 6.4. *Для всякой* Б-*грамматики конечной степени можно построить сильно эквивалентную ей удлиняющую* Б-*грамматику конечной степени.*

Доказательство, весьма простое, предоставляется читателю (ср. доказательство леммы 4.1 и упражнение 4.1).

Связь понятия степени с Д-грамматиками видна из следующей леммы.

Лемма 6.5. *Приведенная удлиняющая* Б-*грамматика тогда и только тогда вкладывается в* Д-*грамматику без циклов* (*и притом эффективно*), *когда ее степень конечна.*

Доказательство. а) Пусть степень Б-грамматики Г конечна, $A \to \omega$ — произвольное правило Г и степень A равна n. Объявим отмеченными все вхождения в ω символов степеней, не превосходящих $n-1$. Ясно, что полученная так Д-грамматика не будет иметь циклов. б) Если Д-грамматика $(\langle V, W, I, R\rangle, f)$ не имеет циклов и наибольшая длина пути в графе $\langle V \cup W; \mathscr{R}\rangle$ равна h, то степень каждого символа $\alpha \in V \cup W$ не превосходит наибольшей длины h_α пути в графе $\langle V \cup W; \mathscr{R}\rangle$, исходящего из узла с меткой α (доказывается очевидной индукцией по h_α). Поэтому степень грамматики $\langle V, W, I, R\rangle$ не превосходит h.

Из лемм 6.1, 6.3, 6.4, 6.5 непосредственно вытекает

Теорема 6.6. а) *Для всякой* Б-*грамматики конечной степени можно построить сильно эквивалентную ей простую* Д-*грамматику.* б) *Если для приведенной удлиняющей* Б-*грамматики* Г *существует сильно эквивалентная ей простая* Д-*грамматика, то степень* Г *конечна.*

Замечания. 1. Сопоставляя конец доказательства леммы 6.5 с леммой 6.2 (пункты б) и в)), видим, что если $G = (\Gamma, f)$ — Д-грамматика ограниченной ширины, то степень Г не превосходит ширины G. Отсюда по лемме 6.3 вытекает, что если Г — приведенная удлиняющая Б-грамматика, то ширина Д-грамматики, сильно эквивалентной Г, не может быть меньше степени Г.

2. Характеристикой Д-грамматики, в известном смысле двойственной ширине, является ее высота — максимум высот соответствующих деревьев подчинения. Если такой максимум существует, естественно сказать, что Д-грамматика имеет ограниченную высоту. О возможности построения сильно или слабо эквивалентной Д-грамматики ограниченной высоты для данной Б-грамматики см. упражнение 7.7.

§ 6.4. Системы уравнений в языках. Формальные степенные ряды

Сейчас мы покажем, что ОБ-грамматики можно интерпретировать как своеобразные системы уравнений, в которых коэффициентами и неизвестными являются языки. Такой интерпретацией, вернее, подсказываемым ею способом записи, особенно охотно пользуются специалисты по языкам программирования.

Фиксируем словарь $V = \{a_1, \ldots, a_s\}$ и конечный набор переменных $\Xi = \{\xi_1, \ldots, \xi_n\}$. Рассмотрим n многочленов от этих переменных, или, как мы будем здесь говорить, неизвестных, с коэффициентами, представляющими собой непустые конечные языки в словаре V: $f_1(\xi_1, \ldots, \xi_n), \ldots, f_n(\xi_1, \ldots, \xi_n)$. (Не требуется,

чтобы каждый многочлен действительно содержал все неизвестные или даже чтобы каждая из них присутствовала хотя бы в одном многочлене.) Для каждого из многочленов $f_1, \ldots, f_n$ существует тождественно равный ему многочлен, коэффициенты которого являются цепочками (точнее, одноэлементными языками) в V; мы будем считать, что коэффициенты всех f_i с самого начала таковы.

Рассмотрим теперь систему уравнений

$$(*)\qquad \begin{cases} f_1(\xi_1, \ldots, \xi_n) = \xi_1, \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ f_n(\xi_1, \ldots, \xi_n) = \xi_n. \end{cases}$$

Такую систему мы будем называть нормальной. Ее решением естественно назвать всякую упорядоченную n-ку $(L_1, \ldots, L_n)$ языков в словаре V, подстановка которой вместо $(\xi_1, \ldots, \xi_n)$ обращает каждое уравнение системы в тождество, т. е. такую, что $f_j(L_1, \ldots, L_n) = L_j$ для каждого $j = 1, \ldots, n$.

Покажем прежде всего, что система (*) имеет по меньшей мере одно решение, которое можно найти «методом последовательных приближений». Именно: определим последовательность n-ок языков $\mathscr{L}^{(0)}, \mathscr{L}^{(1)}, \ldots$, где $\mathscr{L}^{(i)} = (L_1^{(i)}, \ldots, L_n^{(i)})$, следующим образом: (I) $\mathscr{L}^{(0)} = (\varnothing, \ldots, \varnothing)$; (II) $\mathscr{L}^{(i+1)} = (f_1(L_1^{(i)}, \ldots, L_n^{(i)}), \ldots, f_n(L_1^{(i)}, \ldots, L_n^{(i)}))$.

Поскольку всякий многочлен является неубывающей функцией *), из очевидных соотношений $L_1^{(0)} \subseteq L_1^{(1)}, \ldots, L_n^{(0)} \subseteq L_n^{(1)}$ вытекает $L_1^{(1)} \subseteq L_1^{(2)}, \ldots, L_n^{(1)} \subseteq L_n^{(2)}$ и т. д.; иначе говоря, для каждого $j = 1, \ldots, n$ последовательность $L_j^{(0)}, \ldots, L_j^{(i)}, \ldots$ является неубывающей. Положим теперь $\tilde{L}_j = L_j^{(0)} \cup L_j^{(1)} \cup \ldots$ $(j = 1, \ldots, n)$. Тогда, с одной стороны, имеем $L_j^{(i)} \subseteq \tilde{L}_j$ для всех $j = 1, \ldots, n$; $i = 0, 1, \ldots$,

*) Функция $g(X_1, \ldots, X_k)$, аргументы и значения которой суть множества, называется неубывающей, если из $X_1 \subseteq X_1', \ldots, X_k \subseteq X_k'$ следует $g(X_1, \ldots, X_k) \subseteq g(X_1', \ldots, X_k')$. Это свойство очевидным образом имеет место для объединения и произведения а значит, и для любого многочлена.

откуда $\tilde{L}_j = \bigcup_{i=0}^{\infty} f_j(L_1^{(i)}, \ldots, L_n^{(i)}) \subseteq f_j(\tilde{L}_1, \ldots, \tilde{L}_n)$; с другой стороны, если $z \in f_j(\tilde{L}_1, \ldots, \tilde{L}_n)$, то найдутся такие цепочки $x_1 \in \tilde{L}_1, \ldots, x_n \in \tilde{L}_n$, что $z \in f_j(x_1, \ldots, x_n)$; но отсюда, обозначая через i_0 наименьшее i, для которого $x_1 \in L_1^{(i)}, \ldots, x_n \in L_n^{(i)}$, получаем $z \in f_j(L_1^{(i_0)}, \ldots, L_n^{(i_0)}) = L_j^{(i_0+1)} \subseteq \tilde{L}_j$. Итак, $\tilde{L}_j = f_j(\tilde{L}_1, \ldots, \tilde{L}_n)$.

Нетрудно видеть, далее, что если $(L_1, \ldots, L_n)$ — произвольное решение системы (*), то $\tilde{L}_1 \subseteq L_1, \ldots, \tilde{L}_n \subseteq L_n$; действительно, тривиальным образом $L_1^{(0)} \subseteq L_1, \ldots, L_n^{(0)} \subseteq L_1$; отсюда $L_1^{(1)} = f_1(L_1^{(0)}, \ldots, L_n^{(0)}) \subseteq f_1(L_1, \ldots, L_n) = L_1, \ldots, L_n^{(1)} = f_n(L_1^0, \ldots, L_n^{(0)}) \subseteq f_n(L_1, \ldots, L_n) = L_n$ и т. д., откуда и получаются нужные соотношения. Это дает основание назвать решение $(\tilde{L}_1, \ldots, \tilde{L}_n)$ системы (*) наименьшим.

О существовании других решений см. упражнение 6.19.

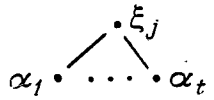
В дальнейшем нам удобно будет считать, что в многочленах «раскрыты все скобки», т. е. что каждый из них представлен в виде объединения выражений вида $x_0\xi_{k_1}x_1 \ldots x_{r-1}\xi_{k_r}x_r$, где $s = 0, 1, \ldots$; $x_0, \ldots, x_r \in V^*$ (см. стр. 25), которые мы будем называть членами f_i. Члены, не содержащие неизвестных, называются свободными.

Требуемое представление многочлена, очевидно, единственно с точностью до перестановки членов.

Сопоставим теперь системе (*) ОБ-грамматику $\Gamma = \langle V, \Xi, \xi_1, R\rangle$, где R состоит из всевозможных правил вида $\xi_j \to \omega$ таких, что $j = 1, \ldots, n$ и ω — член многочлена f_j. Обозначим через $L_j(\Gamma)$ множество всевозможных цепочек в словаре V, выводимых в Γ из ξ_j.

Ясно, что для каждого $j = 1, \ldots, n$ язык $L_j^{(1)}$ есть не что иное, как множество свободных членов f_j, а последнее совпадает с множеством правых частей заключительных правил Γ, имеющих левой частью ξ_j, т. е. цепочек в V, выводимых в Γ из ξ_j за один шаг, — иначе говоря, с помощью деревьев вывода высоты 1.

Покажем, что для любого $i = 0, 1, \ldots$ язык $L_j^{(i)}$ есть множество цепочек в V, выводимых в Γ из ξ_j с помощью деревьев вывода высоты, не превосходящей $i+1$. Действительно, пусть утверждение доказано для всех $i = 0, \ldots, i_0$ и всех $j = 1, \ldots, n$. Но $L_j^{(i_0+1)} = f_j(L_1^{(i_0)}, \ldots, L_n^{(i_0)})$ — это множество всех тех цепочек, которые получаются из членов f_j, если подставлять в них вместо ξ_1 цепочки из $L_1^{(i_0)}$, вместо ξ_2 — цепочки из $L_2^{(i_0)}$ и т. д. (причем разные вхождения одного и того же $\xi_{j'}$ — даже в один член — могут замещаться разными цепочками). Будем теперь в произвольные деревья высоты 1, имеющие вид



где $\alpha_1 \ldots \alpha_t$ — член f_j, подставлять вместо каждого вхождения каждого $\xi_{j'}$, $j' = 1, \ldots, n$, какое-либо $(\xi_{j'}, z)$-дерево грамматики Γ, имеющее высоту, меньшую или равную i_0, и такое, что $z \in V^*$. Полученное множество деревьев обозначим $M_{i_0 j}$. По предыдущему множество $\{\Pr_V u(T) \mid T \in M_{i_0 j}\}$ есть как раз $L_j^{(i_0+1)}$. Однако $M_{i_0 j}$ — это множество (ξ_j, x)-деревьев в Γ высоты, не превосходящей $i_0 + 1$, таких, что $x \in V^*$. Итак, наше утверждение доказано.

Из установленного в предыдущем абзаце факта немедленно вытекает, что $L_j(\Gamma) = \tilde{L}_j$ для любого $j = 1, \ldots, n$.

Упомянем еще об одном способе представления наименьшего решения нормальной системы уравнений — с помощью так называемых формальных степенных рядов.

Фиксируем некоторый пересчет цепочек в алфавите V: $x_0, x_1, \ldots, x_n, \ldots$, в котором более короткие цепочки предшествуют более длинным, так что, в частности, $x_0 = \Lambda$. Фиксируем также некоторую функцию $f(n)$, определенную на натуральном ряду и принимающую в качестве значений натуральные числа. Выражение $\sum_{n=0}^{\infty} f(n) x_n$, или, иначе,

$$f(0)x_0 + f(1)x_1 + \ldots + f(n)x_n + \ldots$$

будем называть некоммутативным формальным степенным рядом (или для краткости просто формальным рядом) в алфавите V. Множество $\{x_i \mid f(i) > 0\}$ назовем опорным множеством данного формального ряда.

Многочлен без переменных, коэффициентами которого служат цепочки в V, можно рассматривать как частный случай формального ряда, если заменить в нем знак $\cup$ знаком $+$, расположить члены соответствующим образом и «привести подобные».

Для заданной конечной последовательности натуральных чисел $p = (n_0, n_1, \ldots, n_N)$ будем обозначать через R_p множество всех формальных рядов в алфавите V, у которых коэффициенты при $x_0, x_1, \ldots, x_N$ равны $n_0, n_1, \ldots, n_N$ соответственно. (В частности, последовательность p может быть пустой — тогда R_p есть множество всех формальных рядов в V.) Если считать окрестностями ряда r множества $R_{p_0}, R_{p_1}, \ldots, R_{p_N}, \ldots$, где p_N — последовательность первых $N+1$ коэффициентов r, то множество всех формальных рядов в словаре V становится топологическим пространством. Сходимость в этом пространстве означает, очевидно, следующее: последовательность рядов $r_1, \ldots, r_s, \ldots$ сходится к ряду r, если для каждого $N = 0, 1, \ldots$ найдется такое $S = 1, 2, \ldots$, что для любого $s = S, S+1, \ldots$ имеют место равенства: $f_s(0) = f(0)$, $f_s(1) = f(1)$, $\ldots$, $f_s(N) = f(N)$, где $f_s(n)$ и $f(n)$ означают коэффициенты при x_n в рядах r_s и r соответственно.

Легко убедиться, что если последовательность рядов $r_1, \ldots, r_s, \ldots$ сходится к ряду r, то опорное множество r является пределом последовательности опорных множеств $r_1, \ldots, r_s, \ldots$*).

Вернемся теперь к системе уравнений (*), наложив на нее некоторые ограничения, а именно: (1) левые части уравнений не должны содержать членов вида ξ_j; (2) ни один из многочленов f_j не должен содержать

*) Множество M называется верхним (нижним) пределом последовательности множеств, если M есть множество всех элементов, принадлежащих бесконечному числу членов последовательности (соответственно всем членам последовательности, кроме, быть может, конечного их числа). Если верхний предел совпадает с нижним, он называется пределом последовательности.

переменной ξ_1; (3) при $j \neq 1$ в f_j не должно быть пустого члена. Если перейти к грамматике, отвечающей системе (*), то порождаемый такой грамматикой язык и при сформулированных сейчас ограничениях может быть произвольным ОБ-языком (следствие из теоремы 4.3, лемма 4.1).

Определим для каждого $j = 1, \ldots, n$ последовательность формальных рядов (являющихся многочленами) $g_j^{(0)}, g_j^{(1)}, \ldots, g_j^{(i)}, \ldots$ следующим образом: (I) $g_j^{(0)}$ есть сумма свободных членов многочлена f_j; (II) $g_j^{(i+1)}$ получается из f_j подстановкой $g_1^{(i)}, \ldots, g_n^{(i)}$ вместо $\xi_1, \ldots, \xi_n$ соответственно с последующим формальным раскрытием скобок и приведением подобных.

Непосредственно ясно, что для любых i, j ($i = 0, 1, \ldots$; $j = 1, \ldots, n$) опорное множество ряда $g_j^{(i)}$ есть $L_j^{(i)}$. В то же время из наложенных на систему (*) ограничений (1)—(3) следует, что каковы бы ни были $i = 0, 1, \ldots$ и $j = 1, \ldots, n$, коэффициенты при цепочках длины, не большей i, во всех рядах $g_j^{(i)}, g_j^{(i+1)}, g_j^{(i+2)}, \ldots$ совпадают. Действительно, цепочки длины, не большей единицы, в любом многочлене $g_j^{(i)}$ — это в точности свободные члены f_j длины 1; если утверждение справедливо для некоторого i_0, то для $i_0 + 1$ оно вытекает из того очевидного факта, что каждая цепочка длины $i_0 + 1$ в многочлене $g_j^{(i+1)}$ при любых i и j либо является свободным членом f_j, либо получается из некоторого члена f_j при подстановке вместо переменных каких-либо членов $g_1^{(i)}, \ldots, g_n^{(i)}$, являющихся цепочками длины, не большей i_0, а такие члены у всех многочленов $g_j^{(i_0)}, g_j^{(i_0+1)}, g_j^{(i_0+2)}, \ldots$ по индуктивному предположению одинаковы. Таким образом, если обозначить через r_j ($j = 1, \ldots, n$) формальный ряд, в котором коэффициенты при каждой цепочке длины i ($i = 0, 1, \ldots$) такие же, как в $g_j^{(i)}$, то к этому ряду будет сходиться последовательность $g_j^{(0)}, g_j^{(1)}, \ldots$ Однако опорное множество ряда $g_j^{(i)}$ есть $L_j^{(i)}$, а предел неубывающей последовательности множеств*) равен объединению всех членов последовательности. Поэтому опорное множество ряда r_j есть $\tilde{L}_j$. Что же касается коэффициента при (произвольной) цепочке x_n в ряде r_i, то он, как легко понять, равен числу (ξ_i, x_n)-деревьев в соответствующей ОБ-грамматике, т. е. числу существенно различных способов вывода в этой грамматике цепочки x_n из символа ξ_i. В частности, для однозначной грамматики коэффициенты ряда r_1 принимают только значения 0 и 1.

*) То есть такой, в которой каждый предыдущий член содержится в следующем.

§ 6.5. Каноническое представление Б-языка

Цель настоящего параграфа — доказать теорему, которая дает любопытное представление Б-языка, называемое иногда каноническим. Предварительно введем некоторые определения.

Пусть V — произвольный словарь, $V_{\mathscr{G}}$ и $V_{\mathscr{F}}$ — подмножества V и P — подмножество V^2. Обозначим через $L'(V, V_{\mathscr{G}}, V_{\mathscr{F}}, P)$ множество тех цепочек в V, которые начинаются символами из $V_{\mathscr{G}}$, кончаются символами из $V_{\mathscr{F}}$ и содержат только такие подцепочки длины 2, которые принадлежат P,— иначе говоря, $L'(V, V_{\mathscr{G}}, V_{\mathscr{F}}, P) = V_{\mathscr{G}}V^* \cap V^*V_{\mathscr{F}} \cap (V^* - V^*(V^2 - P)V^*)$. Ввиду теоремы 5.4 $L'(V, V_{\mathscr{G}}, V_{\mathscr{F}}, P)$ является А-языком; мы будем называть такой А-язык стандартным.

Сопоставим далее каждому символу $a \in V$ «двойника» — новый символ $\bar{a} \notin V$ — и определим два языка $D(V)$ и $D'(V)$, которые будем называть соответственно языком Дика и скобочным языком, следующим образом: (i) пустая цепочка принадлежит обоим языкам; (ii) если цепочки xy и $x'y'$ принадлежат $D(V)$ и $D'(V)$ соответственно, то для произвольного $a \in V$ цепочки $xa\bar{a}y$ и $x\bar{a}ay$ принадлежат $D(V)$, а цепочка $x'a\bar{a}y'$ принадлежит $D'(V)$; (iii) никакая цепочка не может принадлежать $D(V)$ или $D'(V)$ иначе, как в силу (i) или (ii). Очевидно, язык Дика есть не что иное, как множество слов, равных единице в свободной группе с системой образующих V (если считать a и $\bar{a}$ взаимно обратными); скобочный язык, если интерпретировать a и $\bar{a}$ соответственно как левую и правую скобки с меткой a, будет представлять собой множество всевозможных «правильных последовательностей» скобок с метками из V.

Теперь мы можем сформулировать теорему.

Теорема 6.7. *Для всякой* Б-*грамматики* Γ *с основным словарем* V *можно построить такой словарь* $Z \supset V$ *и такой стандартный* А-*язык* L' *в словаре,* Z, *содержащем* Z, *что* $L(\Gamma)$ *является проекцией пересечения* $L' \cap D(Z)$ *на* V *и при этом* $L' \cap D(Z) = L' \cap D'(Z)$.

Доказательство. Для упрощения конструкции будем считать, на что мы имеем право, что $\Gamma = \langle V, W, I, R\rangle$ есть стандартная бинарная Б-грамматика. Сверх того, допустим, что если для какого-либо символа $B \in W$ имеется правило вида $A \to BC$, то не может существовать правила вида $E \to FB$; требуемое для выполнения этого условия преобразование грамматики читатель проведет без труда. Символ $B \in W$ будем называть «левым», соответственно «правым», если имеются правила вида $A \to BC$, соответственно $A \to CB$. Занумеруем как-либо правила Г. Рассмотрим произвольную цепочку $x \in L(\Gamma)$ и произвольную систему составляющих этой цепочки, отвечающую некоторому ее дереву вывода. Расставим в x скобки, отвечающие всем составляющим данной системы (включая тривиальные). Каждую скобку пометим тройкой (A, i, j), где A — метка в узле, от которого «происходит» соответствующая составляющая, j — номер правила, применяемого в данном узле, и i — номер правила, применяемого в узле, подчиняющем данный; у скобок, ограничивающих полную составляющую, значение i может быть произвольным, но непременно одинаковым для обеих скобок. Левую и правую скобки с метками A, i, j будем писать в виде $[A, i, j]$ и $\overline{[A, i, j]}$ соответственно. Далее вставим непосредственно справа от каждого вхождения символа a в цепочку x (левее ближайшей скобки) новый символ $\bar{a}$ — «двойника» a. Полученную цепочку символов и скобок обозначим через $\tilde{x}$, а множество всех таких цепочек (для всех $x \in L(\Gamma)$) — через $\tilde{L}$. Имеем, очевидно, $x = \mathrm{Пр}_V \tilde{x}$, так что $L(\Gamma) = \mathrm{Пр}_V \tilde{L}$.

Пусть, например, Г содержит правила: (1) $I \to AB$, (2) $A \to a$, (3) $B \to b$; тогда одна из возможных цепочек $\widetilde{ab}$ будет иметь вид $[I, 3, 1][A, 1, 2]a\bar{a}\overline{[A, 1, 2]}[B, 1, 3]b\bar{b}\overline{[B, 1, 3]}\overline{[I, 3, 1]}$.

Положим теперь $Z = V \cup V'$, $\tilde{V} = V \cup \bar{V} \cup V' \cup \bar{V}'$, где $\bar{V}$ — множество «двойников» основных символов, а V' и $\bar{V}'$ — соответственно множества (типов) левых и правых скобок, помеченных тройками, как описано выше. Определим стандартный А-язык $L' = L'(\tilde{V}, \tilde{V}_{\mathscr{G}}, \tilde{V}_{\mathscr{F}}, P)$ следующим образом: а) Множества $\tilde{V}_{\mathscr{G}}$ и $\tilde{V}_{\mathscr{F}}$ будут состоять из всевозможных скобок вида $[I, i, j]$ и $\overline{[I, i, j]}$ (с произвольными i, j) соответственно. б) Множество P будет состоять из цепочек, сопоставляемых правилам Г следующим образом: б1) каждому правилу с номером j_0, имеющему вид $A \to BC$, сопоставляются всевозможные цепочки

$$[A, i, j_0][B, j_0, k], \quad \overline{[B, j_0, k]}[C, j_0, l], \quad \overline{[C, j_0, l]}\ \overline{[A, i, j_0]},$$

где i, k, l пробегают соответственно номера всех правил, номера всех правил с левой частью B и номера всех правил с левой частью C; б2) каждому правилу с номером j_0, имеющему вид $A \to a$, сопоставляется цепочка $a\bar{a}$ и всевозможные цепочки $[A, i, j_0]a$, $\bar{a}\overline{[A, i, j_0]}$, где i пробегает номера всех правил.

Покажем, что язык L' не пересекается с множеством $D(Z) - D'(Z)$. Действительно, пусть $\omega \in L' \cap [D(Z) - D'(Z)]$. Поскольку цепочка ω принадлежит $D(Z)$, ее можно сократить до пустой цепочки, последовательно вычеркивая пары вида $\alpha\bar{\alpha}$ или $\bar{\alpha}\alpha$, где $\alpha \in Z$. В то же время $\omega \notin D'(Z)$; поэтому при сокращении хотя бы один раз должна быть вычеркнута пара вида $\bar{\alpha}\alpha$, иначе говоря, ω должна содержать подцепочку вида $\bar{\alpha}\xi\alpha$, где ξ сокращается до пустой цепочки. Цепочку ξ можно выбрать так, чтобы при ее сокращении никакая пара вида $\beta\bar{\beta}$ не вычеркивалась. [Если ξ этим свойством не обладает, то она содержит подцепочку $\bar{\beta}\xi_1\beta$, где ξ_1 сокращается до пустой цепочки; если ξ_1 еще не обладает нужным свойством, то она содержит подцепочку такого же вида, и т. д.] Пусть цепочка ξ такова; она не может быть пустой, поскольку из самого определения языка L' ясно, что его цепочки подцепочек вида $\bar{\alpha}\alpha$ не содержат. Следовательно, ξ имеет вид $\beta\eta\bar{\gamma}$, где β, $\gamma \in Z$. Поскольку ненадчеркнутый символ β стоит непосредственно справа от надчеркнутого символа $\bar{\alpha}$, должно быть $\alpha = [B, j, k]$, $\beta = [C, j, l]$, причем в Г найдется правило

вида $A \to BC$ (действительно, по определению языка L' всякая подцепочка всякой его цепочки, имеющая вид $\bar{\alpha}\beta$, именно такова), так что C — правый символ, а B — левый. Однако совершенно аналогичным образом из того, что α стоит непосредственно справа от $\bar{\gamma}$, получается, что символ B правый. Имеем противоречие. Таким образом, $L' \cap [D(Z) - D'(Z)] = \varnothing$, т. е. $L' \cap D(Z) = L' \cap D'(Z)$. Следовательно, нам достаточно теперь доказать, что $\tilde{L} = L' \cap D'(Z)$. Однако из способа построения цепочек $\tilde{x}$ непосредственно ясно, что каждая такая цепочка принадлежит как L', так и $D'(Z)$. Остается показать, что любая цепочка $\varphi \in L' \cap D'(Z)$ принадлежит $\tilde{L}$. Для этого мы воспользуемся индукцией по длине φ, причем для удобства проведения индукции будем доказывать несколько более общий факт. Сопоставим каждому символу $A \in W$ стандартный А-язык L'_A, отличающийся от L' только тем, что $\tilde{V}_{\mathscr{G}}$ и $\tilde{V}_{\mathscr{T}}$ будут состоять теперь из всевозможных скобок вида $[A, i, j]$ и $[\overline{A, i, j}]$ соответственно. Обозначим через $L_A(\Gamma)$ множество всех цепочек в V, выводимых в Γ из A; для каждой цепочки $x \in L_A(\Gamma)$ построим $\tilde{x}$ так же, как это делалось для цепочек из $L(\Gamma)$, и множество всех таких $\tilde{x}$ обозначим $\tilde{L}_A$. Покажем теперь, что для любого $A \in W$ всякая цепочка $\varphi \in L'_A \cap D'(Z)$ принадлежит $\tilde{L}_A$. Тем самым доказательство будет завершено.

Заметим прежде всего, что если ψ — произвольная подцепочка какой-либо цепочки из $L'_A \cap D'(Z)$, сокращающаяся до пустой цепочки, то либо $\psi = \alpha\xi\bar{\alpha}$, либо $\psi = \beta\eta\bar{\beta}\gamma\zeta\bar{\gamma}$, где выделенные вхождения символов α и $\bar{\alpha}$, β и $\bar{\beta}$, γ и $\bar{\gamma}$ соответствуют друг другу (иначе говоря, сокращаются друг с другом). Действительно, в противном случае мы имели бы $\psi = \alpha\zeta\bar{\alpha}\beta\eta\bar{\beta}\gamma\zeta\bar{\gamma}\theta$; но по определению языка L'_A в его цепочке лишь тогда может содержаться подцепочка вида $\bar{\delta}\varepsilon$, когда $\delta = [B, j, k]$, $\varepsilon = [C, j, l]$, где символ B левый, а C правый. Поэтому получаем $\beta = [D, m, n]$ и символ D оказывается одновременно левым и правым, что невозможно.

Пусть теперь $\varphi \in L'_A \cap D'(Z)$. Если $|\varphi| = 4$ (наименьшая возможная длина цепочки из L'_A), то $\varphi \in L'_A$ означает $\varphi = [A, i, j]a\bar{a}[\overline{A, i, j}]$, причем в Γ должно быть правило $A \to a$ с номером j; но тогда $\varphi = \tilde{a}$, так что $\varphi \in \tilde{L}_A$. Пусть утверждение доказано для всех цепочек длины, меньшей n $(n > 4)$, и $\varphi \in L'_A \cap D'(Z)$, причем $|\varphi| = n$. Тогда $\varphi = \alpha\psi\bar{\alpha}'$, где $\alpha = [A, i, j]$, $\alpha' = [A, i', j']$. Нетрудно заметить прежде всего, что выделенные вхождения скобок α и $\bar{\alpha}'$ соответствуют друг другу, так что $\alpha = \alpha'$. Действительно, в противном случае, по сделанному выше замечанию, скобка, соответствующая α, стояла бы непосредственно слева от скобки, соответствующей $\bar{\alpha}'$, т. е. φ содержала бы подцепочку $[\overline{A, i, j}][A, i', j']$; но тогда A должен был бы быть одновременно левым и правым символом. Далее можно усмотреть, что правило с номером j должно иметь вид $A \to BC$, а цепочка ψ должна начинаться символом вида $[B, j, k]$ и кончаться символом вида $[\overline{C, j, l}]$. В самом деле, в силу определения L'_A в противном случае могло бы быть только $\psi = a\psi'\bar{a}$, причем $\psi \neq \Lambda$, поскольку $|\varphi| > 4$. Однако тогда выделенные вхождения a и $\bar{a}$ должны бы были соответствовать друг другу, иначе в ψ нашлась бы подцепочка $\bar{a}a$, чего в цепочке из L'_A быть не может, а значит, ψ' сокращалась бы до Λ. Однако из определения L' ясно, что ψ' начинается вхождением $\bar{a}$, а значит, при сокращении ψ это вхождение сократилось бы с каким-то вхождением a, стоящим правее, что невозможно. Итак, $\psi = [B, j, k]\chi[\overline{C, j, l}]$. При этом B — левый символ, а C — правый, так что во всяком случае $B \neq C$, и, значит, скобки $[B, j, k]$ и $[\overline{C, j, l}]$ не могут быть соответствующими. Следовательно, $\chi = \chi_1[\overline{B, j, k}]\,[C, j, l]\chi_2$, где χ_1 и χ_2 сокращаются до пустой цепочки. Но это означает, что цепочки $[B, j, k]\chi_1[\overline{B, j, k}]$ и $[C, j, l]\chi_2[\overline{C, j, l}]$ принадлежат соответственно $L'_B \cap D'(W)$ и $L'_C \cap D'(W)$; следовательно, по индуктивному предположению они принадлежат $\tilde{L}_B$ и $\tilde{L}_C$ соответственно. А отсюда — и из наличия в Γ правила $A \to BC$ — следует, что цепочка $\varphi = [A, i, j][B, j, k]\chi_1[\overline{B, j, k}]\,[C, j, l]\chi_2[\overline{C, j, l}][\overline{A, i, j}]$ принадлежит $\tilde{L}_A$.

Упражнения

6.1. Построить К-грамматики, определяющие:
а) язык $\{x\hat{x}|x \in \{a_1, \ldots, a_n\}^+\}$;
б) язык $\{a^n b^n | n = 1, 2, \ldots\}^*$;
в) множество ппф логики высказываний в бесскобочной записи;
г) язык Дика.

6.2. а) Показать, что для всякой К-грамматики можно построить эквивалентную ей К-грамматику с единственной элементарной категорией [Г. М. Ильин, не опубликовано].

б) Оценить происходящее при этом увеличение сложности грамматики.

6.3. Оценить увеличение мощности словаря категорий при построении по данной К-грамматике эквивалентной ей односторонней К-грамматики сложности, не превосходящей 2 (следствие из теоремы 6.1).

6.4. Показать, что класс языков, определяемых левосторонними (или правосторонними) К-грамматиками сложности, не превосходящей 1, «эффективно совпадает» с классом А-языков.

6.5. Показать, что класс языков, определяемых произвольными К-грамматиками сложности, не превосходящей 1, «эффективно совпадает» с классом линейных Б-языков.

6.6. Назовем К-грамматику G категориально однозначной, если каждой цепочке языка $L(G)$ она сопоставляет единственную цепочку категорий, и синтаксически однозначной, если всякая цепочка ее категорий, сокращающаяся до ее главной категории, имеет единственное дерево сокращения.

а) Проверить, являются ли грамматики примеров 1, 2, 3 из § 6.1 категориально и синтаксически однозначными.

б) Пусть L' — множество ппф логики высказываний в варианте, рассмотренном в примере 3 из § 6.1, но со стертыми скобками. Построить категориально однозначную, но синтаксически неоднозначную К-грамматику, определяющую L', таким образом, чтобы для каждой цепочки из L' имелось взаимно однозначное соответствие между ее деревьями сокращения и расстановками скобок, превращающими ее в ппф.

в) Показать, что всякая односторонняя К-грамматика, обладающая тем свойством, что «знаменатели» всех категорий, являющихся частями элементов значений ее приписывающей функции, элементарны, синтаксически однозначна. [Заметим, что грамматика, построенная при доказательстве теоремы 6.1, обладает указанным свойством.]

г) Можно ли распространить результат предыдущего пункта на произвольные односторонние К-грамматики?

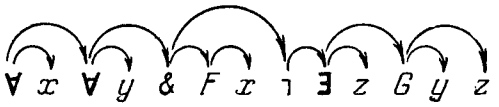
д) Построить синтаксически однозначные К-грамматики для языков примеров 1—3 из § 4.4. Показать, что ни один из этих языков не может быть определен категориально однозначной К-грамматикой.

6.7. Вывести теорему 6.1 из теоремы 6.2.

6.8. Показать, что линейный язык $\{a^m b^n \mid m, n = 1, 2, \ldots; m \leqslant n\}$ не допускается никакой однодорожечной МП-машиной, удовлетворяющей требованию теоремы 6.4.

6.9. Показать, что для всякой МП-машины, принадлежащей одному из перечисленных ниже классов, можно построить эквивалентную ей МП-машину без растяжения, принадлежащую тому же классу:
а) класс однодорожечных МП-машин;
б) класс чисто стирающих МП-машин;
в) класс МП-машин с ограниченным числом поворотов.

6.10. Назовем дерево подчинения для правильно построенной формулы узкого исчисления предикатов в бесскобочной записи «естественным», если оно построено, как в следующем примере:



Построить для множества формул указанного вида Д-грамматику, сопоставляющую каждой из них естественное дерево подчинения (и только его).

6.11. Показать, что Д-грамматика, получаемая из К-грамматики G при обратной иерархизации, имеет ограниченную ширину, а при прямой иерархизации, если язык $L(G)$ бесконечен, — неограниченную.

6.12. Показать, что Д-грамматика, получаемая из К-грамматики сложности 1 при прямой иерархизации, имеет высоту 1, а при обратной иерархизации — ширину 1.

6.13. Показать, что длины путей без ветвления *) в деревьях подчинения, сопоставляемых цепочками К-грамматикой G при прямой иерархизации, не превосходят сложности G [Е. И. Подольская, не опубликовано].

6.14. Назовем Д-грамматику $G = (\Gamma, f)$ однозначной, если никакой цепочке она не сопоставляет более одного дерева подчинения. Соответствующую функцию f будем называть «строгой», если все ее значения — одноэлементные множества. Исследовать взаимоотношение между однозначностью G и строгостью f.

6.15. Построить однозначные Д-грамматики (упражнение 6.14) для языков примеров 1—3 из § 4.4.

6.16. Показать, что всякая приведенная удлиняющая Б-ОАЕВ-грамматика (§ 5.4) имеет конечную степень.

6.17. Показать, что любая приведенная удлиняющая Б-грамматика, порождающая язык $\{a^n b^n | n = 1, 2, \ldots\}$, имеет конечную степень.

6.18. Пусть Γ — НС-грамматика без правил вида $\varphi A\psi \to \varphi B\psi$, $A, B \in W$, и $\varphi A\psi \to \varphi a\psi$, $A \in W - \{I\}$, $a \in V$. [Такие грамматики порождают не все НС-языки (см. теорему 3.7), но и не только Б-языки (достаточно слегка видоизменить пример 1 из § 3.4).]

а) Обобщить определение Д-грамматики на случай, когда ее первая компонента — НС-грамматика указанного вида.

*) Путь $\alpha_1, \ldots, \alpha_{t+1}$ называется путем без ветвления, если узлам $\alpha_1, \ldots, \alpha_t$ подчинены только $\alpha_2, \ldots, \alpha_{t+1}$ соответственно.

б) Показать, что при таком обобщении сохраняет силу пункт а) леммы 6.2.

6.19. а) Показать, что решение нормальной системы уравнений единственно, если ни один из членов левых частей не равен одной из переменных ξ_i или произведению нескольких переменных.

б) Указать примеры, из которых следует, что при нарушении одного из условий пункта а) утверждение этого пункта теряет силу.

6.20. В пространстве формальных рядов в данном алфавите определить расстояние таким образом, чтобы это пространство стало метрическим.

6.21. Показать, что пространство формальных рядов в данном алфавите является полным (т. е. в нем имеет место критерий Коши).

6.22. Определим *сложение*, *умножение* и *адамарово умножение* формальных рядов (обозначения: $+$, $\times$ и $\otimes$ соответственно) следующим образом: если $r = \sum_{n=0}^{\infty} f(n)\, x_n$, $s = \sum_{n=0}^{\infty} g(n)\, x_n$, то $r + s = \sum_{n=0}^{\infty} (f(n) + g(n))\, x_n$, $r \times s = \sum_{n=0}^{\infty} h(n)\, x_n$, где $h(n)$ — сумма всевозможных произведений $f(i)\, g(j)$ таких, что $x_i x_j = x_n$, $r \otimes s = \sum_{n=0}^{\infty} f(n)\, g(n)\, x_n$. Показать, что опорные множества рядов $r + s$, $r \times s$ и $r \otimes s$ равны соответственно объединению, произведению и пересечению опорных множеств r и s.

6.23. Показать, что дополнение к языку Дика $D(V)$ можно представить в виде $S(L'; a_1, \ldots, a_n, \bar{a}_1, \ldots, \bar{a}_n \mid a_1 D(V), \ldots, a_n D(V), \bar{a}_1 D(V), \ldots, \bar{a}_n D(V))$, где $\{a_1, \ldots, a_n\} = V$ и L' — стандартный A-язык в словаре $\{a_1, \ldots, a_n, \bar{a}_1, \ldots, \bar{a}_n\}$. Аналогично для скобочного языка.

6.24. Построить детерминированные МП-машины, допускающие языки $D(V)$, $CD(V)$, $D'(V)$ и $CD'(V)$ соответственно.

6.25. а) Показать, что теорема 6.7 останется справедливой, если взять в качестве Г линейную Б-грамматику и заменить $D(Z)$ и $D'(Z)$ языками $D_1(Z)$ и $D_1'(Z)$ соответственно, где $D_1(Z)$ и $D_1'(Z)$ определяются следующим образом: (i) $\Lambda \in D_1(Z)$, $\Lambda \in D_1'(Z)$; (ii) если $x \in D_1(Z)$, $x' \in D_1'(Z)$ и $a \in V$, то $ax\bar{a}$, $\bar{a}xa \in D_1(Z)$, $ax'\bar{a} \in D_1'(Z)$; (iii) никакая цепочка не может принадлежать $D_1(Z)$ или $D_1'(Z)$ иначе, как в силу (i) или (ii).

б) Сформулировать и доказать аналогичные утверждения для металинейных Б-грамматик, итерационно-линейных Б-грамматик и Б-ОАЕВ-грамматик.

6.26. Показать, что если в теореме 6.7 вместо стандартности языка L' потребовать выполнения более слабого условия — k-определенности для некоторого k, зависящего от Г (см. упражнение 5.7), то словарь Z и язык $D(Z)$ можно выбрать одни и те же для всех Б-грамматик с данным основным словарем.

ГЛАВА 7

СЛОЖНОСТЬ ВЫВОДА В БЕСКОНТЕКСТНЫХ ГРАММАТИКАХ

Для классификации Б-грамматик по сложности вывода «разрешающая способность» «главных» сигнализирующих операторов — временно́й сложности и емкости — оказывается недостаточной. Действительно, для всякой Б-грамматики Г функция $T_\Gamma(n)$ мажорируется линейной функцией (упражнение 4.1, б)) и сверх того для любой Б-грамматики Г можно построить эквивалентную Б-грамматику Г′ такую, что $T_{\Gamma'}(n) < cn$, где c — произвольное наперед заданное положительное число (это тривиальным образом вытекает из теоремы 6.3); но временна́я сложность грамматики, порождающей бесконечный язык, не может быть по порядку меньше линейной функции (неравенства (1) и (2) на стр. 59). Что же касается емкости, то она, как мы знаем, не позволяет классифицировать даже неукорачивающие грамматики. Зато квазисигнализирующие операторы «низших» типов — левая и правая глубина, разброс, активная емкость, для которых соответствующие относительные операторы в классе Б-грамматик являются сигнализирующими, — дают в рассматриваемом случае весьма интересную картину. К ее описанию мы и перейдем.

§ 7.1. Глубина и разброс

Левая глубина. Следующая теорема устанавливает связь между левой глубиной Б-грамматики и степенями левого ветвления отвечающих ей систем составляющих.

Теорема 7.1 (ср. упражнение 3.5). *Для всякой* Б-*грамматики* $\Gamma = \langle V, W, I, R\rangle$

$$\mathcal{Y}^{\text{Л}}_{\Gamma}(n) \leqslant \Upsilon^{\text{Л}}_{\Gamma}(n) + 1 \leqslant \mathcal{Y}^{\text{Л}}_{\Gamma}(n) + d,$$

где d *— максимум длин цепочек* $x \in V^*$, *для которых существуют правила вида* $A \to x\theta \in R$, *причем* $\theta \neq \Lambda$. *В частности, для стандартной бинарной* Б-*грамматики* $\mathcal{Y}^{\text{Л}}_{\Gamma}(n) = \Upsilon^{\text{Л}}_{\Gamma}(n) + 1$.

Имеет место даже более сильное утверждение: $\tilde{\mathcal{Y}}^{\text{Л}}(\Gamma, x) \leqslant \tilde{\Upsilon}^{\text{Л}}(\Gamma, x) + 1 \leqslant \tilde{\mathcal{Y}}^{\text{Л}}(\Gamma, x) + d$ (смысл символов $\tilde{\mathcal{Y}}^{\text{Л}}$ и $\tilde{\Upsilon}^{\text{Л}}$ ясен из систем обозначений, использовавшихся на стр. 55 и 83).

Назовем левой глубиной цепочки $x\omega$, где $x \in V^*$ и $\omega \in W(V \cup W)^* \cup \{\Lambda\}$, число $|\omega|$. Левой глубиной вывода будем называть наибольшую из левых глубин входящих в него цепочек. Легко видеть, что среди всех выводов в Б-грамматике, отвечающих одному и тому же дереву вывода, наименьшую левую глубину имеет упорядочиваемый вывод (всегда существующий и единственный — см. стр. 119). Поэтому нужное утверждение непосредственно получается из следующей леммы.

Лемма 7.1. *Пусть* (в обозначениях теоремы 7.1) $A \in W$, $x \in V^+$ *и* T *есть* (A, x)-*дерево. Тогда*

$$y^{\text{Л}} \leqslant \mathrm{y}^{\text{Л}}(T) + 1 \leqslant y^{\text{Л}} + d,$$

где $\mathrm{y}^{\text{Л}}(T)$ *— степень левого ветвления индуцируемой деревом* T *системы составляющих для* x *и* $y^{\text{Л}}$ *— левая глубина соответствующего* T *упорядочиваемого вывода.*

Доказательство. Если высота T равна единице, неравенства очевидны. Пусть они доказаны для всех (B, z)-деревьев высоты, меньшей n $(B \in W,\ z \in V^+)$, высота T равна n и $\alpha_0 > \ldots > \alpha_{k-1}$ — все узлы T, подчиненные корню. Для каждого узла α_i, помеченного вспомогательным символом, обозначим через $\mathrm{y}^{\text{Л}}_i$ степень левого ветвления системы составляющих, индуцируемой на соответствующей подцепочке x полным α_i-поддеревом T (само это поддерево обозначим T_i), и через $y^{\text{Л}}_i$ — левую глубину отвечающего этому поддереву упорядочиваемого вывода. По индуктивному предположению $y^{\text{Л}}_i \leqslant \mathrm{y}^{\text{Л}}_i + 1 \leqslant y^{\text{Л}}_i + d$. Положим, кроме того, $\mathrm{y}^{\text{Л}}_i = 0$,

если α_i помечен основным символом. Имеем, очевидно, $y^{\text{Л}} = \max(i + y^{\text{Л}}_i)$, $\mathrm{y}^{\text{Л}}(T) = \max(i + \mathrm{y}^{\text{Л}}_i)$, причем в первом случае максимум берется по тем i, для которых α_i помечены вспомогательными символами, а во втором — по всем $i = 0, \ldots, k-1$. Обозначим через $\mathrm{y}^{\text{Л}'}$ максимум чисел $i + \mathrm{y}^{\text{Л}}_i$ по тем i, для которых α_i помечены вспомогательными символами, и через $\mathrm{y}^{\text{Л}''}$ — максимум тех же чисел по остальным i (если они есть). Тогда $y^{\text{Л}} \leqslant \mathrm{y}^{\text{Л}'} + 1 \leqslant \mathrm{y}^{\text{Л}} + 1$. С другой стороны, $\mathrm{y}^{\text{Л}'} \leqslant y^{\text{Л}}(T) + d - 1$; что же касается числа $\mathrm{y}^{\text{Л}''}$, то оно, очевидно, меньше $\mathrm{y}^{\text{Л}'}$, если узел α_{k-1} помечен вспомогательным символом, и равно $k-1$ в противном случае. Но если i_0 — наибольшее i, для которого α_i помечен вспомогательным символом, то $k - 1 - i_0 \leqslant d$ и $i_0 \leqslant y^{\text{Л}} - 1$, так что $(k-1) + 1 = k < y^{\text{Л}} + d$. Этим доказательство заканчивается.

Связь между левой глубиной и степенью левого ветвления была впервые замечена В. Ингве [Yngve 1960]; именно эта связь была выдвинута им в качестве основного аргумента для обоснования гипотезы об ограниченности степеней левого ветвления в естественных языках (стр. 290). Дело в том, что если моделировать процесс произнесения («порождения») предложения человеком с помощью вывода в Б-грамматике, то для каждой промежуточной цепочки вывода ее отрезок вправо от самого левого вхождения вспомогательного символа (включая это вхождение) естественно интерпретировать как ту информацию, которую говорящий должен на соответствующем шаге процесса держать в памяти. [Такая интерпретация будет особенно наглядной, если взять грамматику с правилами вида $A \to xX$, где $x \in V^*$ и $X \in W^*$ (например, стандартную бинарную или в нормальной форме), и построить для нее эквивалентную МП-машину способом, использованным в доказательстве теоремы 6.4. Тогда каждая промежуточная цепочка упорядочиваемого вывода будет иметь вид zZ, где $z \in V^*$ и $Z \in W^*$, а при переходе к МП-машине цепочка Z (точнее, $\hat{Z}$) станет содержимым рабочей ленты, т. е. «ленты памяти», на соответствующем шаге вычисления.] Но объем «оперативной памяти» человека ограничен сверху, причем граница, по данным экспериментальной психологии, сравнительно невысока. То

обстоятельство, что гипотеза в целом не подтверждается (стр. 290), заставляет сделать вывод, что с помощью Б-грамматик — во всяком случае, без дополнительных механизмов — нельзя достаточно удовлетворительно моделировать процесс порождения предложения носителем языка *).

Тем не менее Б-грамматики, левые глубины которых ограничены константами, представляют значительный лингвистический интерес, хотя бы потому, что ограниченность степеней левого ветвления является существенной типологической характеристикой языка. Возникает вопрос о природе языков, порождаемых такими грамматиками, а также множеств цепочек, выводимых в произвольных Б-грамматиках с помощью выводов ограниченной левой глубины. Ответ на эти вопросы дает следующая

Теорема 7.2. *Для любой* Б-*грамматики* Γ *и любого натурального числа* k *множество* $L_k^{y\text{л}}$ *является* А-*языком. Более того, для всякой* Б-*грамматики* Γ *и всякого натурального* k *можно построить* А-*грамматику, порождающую* $L_k^{y\text{л}}(\Gamma)$.

Доказательство. Пусть $\Gamma = \langle V, W, I, R\rangle$. Сопоставим каждой цепочке $\omega \in (V \cup W)^+$, $|\omega| \leqslant k$, новый символ $\alpha(\omega)$ и каждой тройке (φ, ψ, x), где $\varphi \in (V \cup W)^+$, $\psi \in (V \cup W)^*$, $x \in V^*$, $|\varphi|$, $|\psi| \leqslant k$ и $\varphi \underset{\Gamma}{\models} x\psi$, новое правило $\alpha(\varphi) \to x\alpha(\psi)$, если $\psi \neq \Lambda$, и $\alpha(\varphi) \to x$, если $\psi = \Lambda$. Положим $\Gamma' = \langle V, W', \alpha(I), R'\rangle$, где W' состоит из всех новых символов и R' — из всех новых правил. Грамматика Γ' правосторонняя линейная, и для нее без труда строится эквивалентная ей автоматная (ср. стр. 169). В то же время легко видеть, что $L(\Gamma') = L(\Gamma)$.

Следствие. *Для любой* Б-*грамматики* Γ, *для которой* $\mathcal{Y}_\Gamma^{\text{л}}(n)$ *ограничена числом* k, *можно, если известно* k, *построить эквивалентную ей* А-*грамматику.*

*) Это утверждение ни в коей мере не означает отрицания лингвистической значимости Б-грамматик. Грамматики, неадекватные для моделирования процесса порождения предложения человеком, могут быть в то же время пригодны для описания структуры законченного предложения, подобно тому, как с помощью языка математической логики можно представить готовое доказательство теоремы, но не процесс его нахождения математиком.

Заметим, что если Γ — приведенная грамматика, то $\mathcal{Y}_\Gamma^{\text{л}}(n) = 1$ тогда и только тогда, когда Γ — А-грамматика. Таким образом, если C — класс всех функций-констант, то $\mathcal{L}_C^{y\text{л}}(\text{Б}) = \mathcal{L}(\text{А})$.

В общем случае для Б-грамматики Γ имеем, очевидно, $\mathcal{Y}_\Gamma^{\text{л}}(n) \leqslant n$, и существуют Б-грамматики, для которых это неравенство переходит в равенство, — примером может служить хотя бы грамматика $\Gamma = \langle \{a\}, \{I\}, I, \{I \to Ia, I \to a\}\rangle$; впрочем, язык $L(\Gamma) = a^+$ может быть порожден и А-грамматикой, т. е. грамматикой левой глубины 1. Естественно спросить, существуют ли Б-языки, не порождаемые никакими Б-грамматиками левой глубины, меньшей n. Ответ на этот вопрос отрицателен; более того, для любой Б-грамматики Γ и любого действительного числа c, $0 < c < 1$, можно построить Б-грамматику Γ', эквивалентную Γ и такую, что для всех достаточно больших n имеет место неравенство $\mathcal{Y}_{\Gamma'}^{\text{л}}(n) \leqslant cn$ (так что для любой линейной функции $f(n) = cn$, $c > 0$, имеем $\mathcal{L}_f^{y\text{л}}(\text{Б}) = \mathcal{L}(\text{Б})$. [A fortiori это утверждение верно также для разброса и активной емкости.] Действительно, нужным свойством будет обладать всякая грамматика, удовлетворяющая требованию теоремы 6.3, если только взять фигурирующее в условии этой теоремы число s достаточно большим (например, положить $s \geqslant \frac{1}{c}$).

В то же время существуют Б-языки (и даже линейные языки), удовлетворяющие тому условию, что для каждой порождающей такой язык Б-грамматики ее левая глубина мажорирует некоторую линейную функцию. Приведем пример.

Пример 1. Рассмотрим язык $L = \{a^n b^n \mid n = 1, 2, \ldots\}$ и произвольную порождающую его Б-грамматику $\Gamma = \langle \{a, b\}, W, I, R\rangle$. Мы можем считать, что Γ — приведенная грамматика без правил вида $A \to B$, A, $B \in W$; действительно, при переходе от произвольной Б-грамматики к приведенной (лемма 4.8) множество полных выводов не меняется, а переход к грамматике без правил вида $A \to B$ можно, как показывает очевидный анализ доказательства леммы 4.1, осуществить так, чтобы

между множествами полных выводов в старой и новой грамматиках существовало взаимно однозначное соответствие со следующим свойством: для каждой цепочки ω из полного вывода в старой грамматике найдется цепочка ω' из соответствующего вывода в новой такая, что $|\omega'| = |\omega|$ и на всех местах, на которых в ω стоят основные (соответственно вспомогательные) символы, в ω' также стоят основные (вспомогательные) символы, и обратно.

Покажем прежде всего, что ни из какого символа $A \in W$ не может быть выводима цепочка вида $\varphi B \psi C \chi$, где B и C — циклические вспомогательные символы. Действительно, в противном случае мы имели бы $A \vdash xByCz$, $B \vdash u_1 B u_2$, $C \vdash v_1 C v_2$, где x, y, z, u_1, u_2, v_1, $v_2 \in V^*$, $u_1 u_2 \neq \Lambda \neq v_1 v_2$. Кроме того, $I \vdash t_1 A t_2$, $B \vdash r$, $C \vdash s$, где t_1, t_2, r, $s \in V^*$. Но тогда $I \vdash t_1 x u_1 B u_2 y v_1 C v_2 z t_2$ и $t_1 x u_1 r u_2 y v_1 s v_2 z t_2 = a^n b^n$ для подходящего n. Поэтому либо $u_1 r u_2$ состоит только из a, либо $v_1 s v_2$ — только из b; и если, например, справедливо первое, то оказывается невозможным $t_1 x u_1^2 r u_2^2 y v_1 s v_2 z t_2 \in L$, что по предыдущему должно иметь место.

Пусть теперь T — произвольное дерево вывода цепочки $a^n b^n$ из I в Г. Из предыдущего следует, что, каковы бы ни были два пути в T, идущих из корня в узлы, помеченные циклическими символами, один из них является продолжением другого. Наибольший из таких путей назовем главным. Никакое поддерево с корнем в произвольном узле α главного пути, состоящее из всех узлов, зависящих от α и не входящих в главный путь, не содержит узлов, помеченных циклическими символами, так что высота такого поддерева не превосходит числа $p = \mu(W)$, и поэтому оно содержит не более $q = 1 + g + g^2 + \ldots + g^p$ узлов, где g — максимум длин правых частей правил Г. Следовательно, главный путь содержит не менее $k/(q+1)$ узлов, где k — общее число узлов дерева T. Далее для подходящего символа $A \in W$ найдется последовательность $\alpha_1, \ldots, \alpha_h$ узлов, лежащих на главном пути, упорядоченных «сверху вниз» и помеченных этим символом, такая, что для каждого $i = 0, \ldots, h-1$ длина пути из α_i в α_{i+1} не превышает p, откуда $h \geqslant ck$, где $c = \frac{1}{p(q+1)}$, и тем более $h \geqslant c \cdot 2n$; важно заметить, что c — константа, не зависящая от n и от T.

Обозначим через z_i составляющую цепочки $a^n b^n$, «происходящую» от α_i. Рассуждениями, аналогичными использованным в начале разбора примера, без всякого труда устанавливается, что для любого $i = 1, \ldots, h$ будет $z_i = a^{k_i} b^{l_i}$, причем $k_i - k_{i+1} = l_i - l_{i+1}$ для всех $i = 1, \ldots, h-1$. Поэтому, во всяком случае, $l_i - l_{i+1} > 0$. Пусть теперь $(I = \omega_0, \omega_1, \ldots, \omega_m = a^n b^n)$ — произвольный вывод, отвечающий дереву T, и $\omega_{j_1}, \ldots, \omega_{j_h}$ — те его цепочки, которые «проходят» через $\alpha_1, \ldots, \alpha_h$ соответственно. Если в цепочках ω_{j_i} и $\omega_{j_{i+1}}$ выделить те вхождения A, которые отвечают узлам α_i и α_{i+1} соответственно, то мы получим $\omega_{j_i} = \xi A \eta$, $\omega_{j_{i+1}} = \xi' \zeta A \theta \eta'$, где $\xi \vdash \xi'$, $\eta \vdash \eta'$, $A \vdash \zeta A \theta$; при этом $\theta \neq \Lambda$, поскольку $\theta \vdash b^{l_i - l_{i+1}} \neq \Lambda$. Следовательно, цепочка ω_h будет содержать подцепочку вида $A\theta_{h-1}\theta_{h-2} \ldots \theta_1$, где все θ_i непусты, так что левая глубина этой цепочки, а значит, и всего вывода не меньше $h \geqslant c \cdot 2n$. Таким образом, $\mathcal{G}^{\text{Л}}_{\Gamma}(n) \geqslant cn$. [См. также упражнение 7.2.]

Для **правой глубины** картина такая же, как для левой, с точностью до обращения.

Разброс. Нетрудно видеть, что роль, которую по отношению к глубине играют A-языки, по отношению к разбросу принадлежит линейным языкам. Именно, справедлива

Теорема 7.3. *Для любой* Б-*грамматики* Г *и любого натурального числа* k *множество* $L_k^D(\Gamma)$ *является линейным языком. Более того, для всякой* Б-*грамматики* Г *и всякого натурального* k *можно построить линейную* Б-*грамматику, порождающую* $L_k^D(\Gamma)$.

Доказательство. Пусть $\Gamma = \langle V, W, I, R \rangle$. Определим новые символы $\alpha(\omega)$, как в доказательстве теоремы 7.2. Каждой четверке (φ, ψ, x, y), где $\varphi \in (V \cup W)^+$, $\psi \in (V \cup W)^*$, x, $y \in V^*$, $|\varphi|$, $|\psi| \leqslant k$ и $\varphi \underset{\Gamma}{\models} x\psi y$, сопоставим новое правило $\alpha(\varphi) \to x\alpha(\psi)y$, если $\psi \neq \Lambda$, и $\alpha(\varphi) \to xy$, если $\psi = \Lambda$. Положим $\Gamma' = \langle V, W', \alpha(I), R' \rangle$, где W' состоит из всех новых символов и R' — из всех новых правил. Эквивалентность Г и Г′ очевидна.

Следствие. *Для любой* Б-*грамматики* Г, *для которой* $D_\Gamma(n)$ *ограничена числом* k, *можно, если известно* k, *построить эквивалентную ей линейную* Б-*грамматику.*

Остается заметить, что если Г — приведенная грамматика, то $D_\Gamma(n) = 1$ тогда и только тогда, когда Г линейна, так что $\mathscr{L}^D_C(\text{Б}) = \mathscr{L}(\text{Л})$ (Л — класс линейных Б-грамматик).

Металинейный язык — и даже произведение двух линейных языков — может уже удовлетворять тому условию, что для каждой порождающей его Б-грамматики ее разброс мажорирует некоторую линейную функцию.

Пример 2. Пусть $L = \{a^m b^m a^n b^n \mid m, n = 1, 2, \ldots\}$, $\Gamma = \langle V, W, I, R\rangle$ — Б-грамматика, и $L(\Gamma) = L$. Как и в примере 1, можно считать Г приведенной и не имеющей правил вида $A \to B$, $A, B \in W$. В произвольном дереве вывода T цепочки $a^m b^m a^n b^n$ из I в Г нетрудно, рассуждая аналогично примеру 1, найти два «главных пути», исходящих из одного и того же узла α (не обязательно совпадающего с корнем) и содержащих все узлы T, помеченные циклическими символами. На этих путях для подходящих символов $A, B \in W$ найдутся последовательности узлов $\alpha_1, \ldots, \alpha_h$ и $\beta_1, \ldots, \beta_f$ такие, что $\alpha_1, \ldots, \alpha_h$ имеют метку A, а $\beta_1, \ldots, \beta_f$ — метку B, причем $h, f \geqslant ck$, где k — общее число узлов T и c — константа, определяемая так же, как в примере 1. Обозначая через z_i и u_j составляющие цепочки $a^m b^m a^n b^n$, «происходящие» от α_i и β_j соответственно, видим, что $z_i = a^{k_i} b^{l_i}$, $u_j = a^{d_j} b^{e_j}$ и при этом $k_i - k_{i+1} = l_i - l_{i+1}$, $d_j - d_{j+1} = e_j - e_{j+1}$, так что $l_i - l_{i+1}$, $d_j - d_{j+1} > 0$. Поэтому в произвольном выводе, отвечающем дереву T, найдется цепочка, содержащая вхождение подцепочки вида $A\theta_{h-1} \ldots \theta_1 \omega \sigma_1 \ldots \sigma_{f-1} B$, где $\theta_i, \sigma_j \neq \Lambda$ (мы считаем здесь, что узлы $\alpha_1, \ldots, \alpha_h$ принадлежат левому главному пути, а $\beta_1, \ldots, \beta_f$ — правому). Но длина этой подцепочки не меньше $h + f \geqslant 2ck \geqslant 2c \cdot |a^m b^m a^n b^n|$.

[См. также упражнение 7.17].

§ 7.2. Активная емкость

Активная емкость Б-грамматик ведет себя сложнее, чем глубина и разброс. Оказывается, прежде всего, что, — в то время как все классы $\mathscr{L}^{\text{ЛЛ}}_k(\text{Б})$, где k — функции-константы, совпадают между собой, и то же верно для $\mathscr{L}^D_k(\text{Б})$, — разность $\mathscr{L}^I_{k+1}(\text{Б}) - \mathscr{L}^I_k(\text{Б})$ не пуста, каково бы ни было $k = 1, 2, \ldots$ ($\mathscr{L}^I_1(\text{Б})$ есть, очевидно, класс линейных языков). Это доказывается следующим примером.

Пример 1. Рассмотрим две бесконечные последовательности символов $A_0, A_1, \ldots, B_1, B_2, \ldots$ и построим последовательность Б-грамматик $\Gamma_0, \Gamma_1, \ldots$, причем каждая грамматика будет иметь основной словарь $\{a, b, c, d\}$, вспомогательный словарь $\{A_0, \ldots, A_k, B_1, \ldots, B_k\}$ ($\{A_0\}$ при $k = 0$) и начальный символ A_k, следующим образом:

I. $\Gamma_0 = \langle \{a, b, c, d\}, \{A_0\}, A_0, \{A_0 \to cd\}\rangle$;

II. Γ_{k+1} получается из Γ_k добавлением к вспомогательному словарю символов A_{k+1} и B_{k+1}, причем A_{k+1} объявляется начальным символом, и к схеме — правил $A_{k+1} \to cB_{k+1}d$, $A_{k+1} \to cd$, $B_{k+1} \to aB_{k+1}b$, $B_{k+1} \to aA_iA_jb$ $(i, j = 0, \ldots, k)$.

Полагая $L(\Gamma_k) = L_k$, имеем $L_0 = \{cd\}$, $L_{k+1} = \{ca^n z b^n d \mid n = 1, 2, \ldots, z \in L^2_k\} \cup \{cd\}$.

Цепочки, принадлежащие языкам $L_0, L_1, \ldots$, мы будем называть «гроздьями». Для каждой грозди w определим ее высоту $h(w)$: $h(cd) = 0$; если $w = ca^m cudb^m dca^n cvdb^n d$ и $h(cud) = i$, $h(cvd) = j$, то $h(w) = \max(i, j) + 1$. Гроздь будет называться «совершенной», если в любой ее подцепочке, имеющей вид $v_1 v_2$, где v_1 и v_2 — гроздья, высоты v_1 и v_2 равны.

Каждый язык L_k состоит, очевидно, в точности из всех гроздьев высоты, не большей k.

При $k \geqslant 1$ для любой грозди высоты, не превосходящей k, наименьшая активная емкость ее вывода в Γ_k не превосходит k. В самом деле, для Γ_1 это тривиально; пусть доказано для Γ_k; любой полный вывод в Γ_{k+1} содержит цепочку вида $ca^n A_i A_j b^n d$, $i, j \leqslant k$, причем предыдущие цепочки содержат по одному вхождению вспомогательного символа, а в последующих наименьшее число вхождений вспомогательных символов получится, если сначала полностью «обработать» одно из вхождений A_i или A_j — это делается по правилам Γ_i или Γ_j соответственно, — а потом другое. Итак, $L_k \in \mathscr{L}^I_k(\text{Б})$.

Установим теперь некоторые свойства гроздьев. Первые два из них очевидны:

I. В каждой грозди вхождений a равно числу вхождений b, а число вхождений c — числу вхождений d.

II. В любом начале грозди число вхождений b не превосходит числа вхождений a, а число вхождений d — числа вхождений c.

Следующие три свойства без труда доказываются индукцией по высоте грозди.

Назовем вхождение c в гроздь «правым», если оно непосредственно следует за вхождением d, и «левым» в противном случае.

III. В каждой грозди число левых вхождений c на единицу больше числа правых.

IV. Если гроздь содержит подцепочку $cdq_1cdq_2 \ldots \ldots cdq_hcd$, где $h \geqslant 2^t$, то длина хотя бы одной из цепочек $q_1, \ldots, q_h$ не меньше $4(t-1)$.

V. Если cqd — подцепочка грозди, содержащая одинаковое число вхождений c и d и такая, что во всяком ее начале число вхождений d не превосходит числа вхождений c, то q содержит одинаковое число вхождений a и b.

Остается одно свойство:

VI. Пусть некоторую гроздь w высоты $k > 0$ можно представить в виде $w = u_1xyzu_2$ таким образом, что из цепочек x и z хотя бы одна не пуста и для любого $n = 1, 2, \ldots$ цепочка $w_n = u_1x^nyz^nu_2$ также является гроздью. Тогда: а) x, y, z имеют вид $x = a^l$, $y = a^iv_1v_2b^j$, $z = b^l$, где v_1 и v_2 — гроздья; б) любая цепочка w_n является гроздью высоты k.

Поскольку б) следует из а) непосредственно, достаточно доказать а).

Допустим сначала, что x содержит вхождения c и d. Тогда возможны два случая: 1) некоторое вхождение c предшествует в x некоторому вхождению d; 2) все вхождения c следуют за всеми вхождениями d.

Случай 1. Рассмотрим последнее из тех вхождений c в x, за которыми следуют какие-либо вхождения d в x, и первое из вхождений d в x, следующих за этим вхождением c. Ясно, что интервал, ограниченный этими вхождениями, пуст, так что $x = rcds$, откуда $x^n = r(cdsr)^{n-1}cds$, причем n может быть сколь угодно большим. Но это противоречит свойству IV.

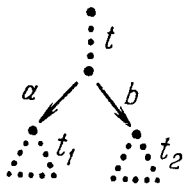
Случай 2. Рассмотрев последнее вхождение c и первое вхождение d в x, представляем x в виде $r_1cr_2 = s_1ds_2$, где r_2 и s_1 не содержат c и d; отсюда $x^2 = r_1cr_2s_1ds_2$, а поскольку x^2 — подцепочка грозди, r_2 и s_1 пусты. Далее рассуждаем, как в случае 1.

Таким образом, x не может содержать вхождений c и d одновременно. Но если x содержит вхождения d, не содержа вхождений c, то при достаточно больших n начало u_1x^n цепочки w_n не будет обладать свойством II. Если же x содержит вхождения c, не содержа вхождений d, то все вхождения c в x^n, кроме разве одного, будут левыми; в то же время, поскольку x и z с точностью до перемены местами c и d ведут себя одинаково, цепочка z тоже не может содержать вхождений c и d одновременно; если она не содержит d, то уже цепочка w_2 будет нарушать свойство I, а если она содержит d, не содержа c, то w_2 нарушит свойство III.

Итак, x не содержит ни c, ни d, и то же, разумеется, верно для z. Отсюда сразу следует, что цепочка x не может содержать a и b одновременно, иначе она содержала бы вхождение ab или ba, но таких подцепочек в гроздьях нет. То же верно для z. Если теперь предположить для определенности, что $x \neq \Lambda$, имеем $x = a^l$ или $x = b^l$, $l > 0$. Но второе невозможно из-за нарушения свойства II для цепочки w_n при достаточно большом n. Поэтому $x = a^l$, откуда $z = b^l$ ввиду свойства I. Остается показать, что y также имеет требуемый вид. Если высота w равна 2, это очевидно. Пусть предложение доказано для всех высот, меньших k, и высота w равна k, так что $w = cpt_1t_2qd$, где $p = a^m$, $q = b^m$ и t_1, t_2 — гроздья высоты, не большей $k-1$. Если x содержится в p и z — в q, наше утверждение очевидно; но при x, содержащемся в p, z должно содержаться в q, и обратно, иначе цепочка w_2 не порождалась бы грамматикой Γ_k. Если обе цепочки x и z содержатся в t_1 или в t_2, то утверждение справедливо по индуктивному предположению. Остается случай, когда x содержится в t_1 и z — в t_2; но в этом случае w_2 нарушала бы свойство V.

Теперь мы перейдем к доказательству основного утверждения: покажем, что $L_{k+1} \notin \mathscr{L}_k^l$(Б) при любом

$k = 1, 2, \ldots$ Для этого введем сначала вспомогательное понятие «совершенного бинарного дерева порядка k» (сокращенно «k-дерева»). Именно: 1-деревом называется всякое дерево ширины 1; если для некоторого k понятие k-дерева определено, то $(k+1)$-деревом называется всякое дерево вида



где t — дерево ширины 1 (возможно, единичное), a и b — дуги, t_1 и t_2 — k-деревья.

Если в дереве T полного вывода в Б-грамматике найдется поддерево, не содержащее висячих узлов T и являющееся k-деревом, то во всяком выводе, отвечающем дереву T, имеется цепочка, содержащая не менее k вхождений вспомогательных символов. Действительно, для $k = 1$ это очевидно; если для некоторого k утверждение справедливо, и T содержит поддерево T', не содержащее висячих узлов T и являющееся $(k+1)$-деревом, то во всяком выводе $(\omega_0, \ldots, \omega_s)$, отвечающем T, некоторая цепочка ω_i будет содержать вхождение вспомогательного символа, отвечающее корню T'; из определения $(k+1)$-дерева ясно, что некоторая цепочка ω_j, $j > i$, будет содержать два вхождения α и β вспомогательных символов, отвечающих корням двух k-деревьев; но тогда при дальнейшем выводе наименьшее число вхождений вспомогательных символов в одну цепочку, являющихся потомками α и β, получится, если сначала полностью «обработать» α, а потом β, или наоборот, и это число в силу индуктивного предположения будет не меньше $k+1$.

Далее нам будет удобно воспользоваться еще одним вспомогательным понятием — понятием «совершенной бинарной системы отрезков (цепочки) порядка k» (сокращенно «k-системы»), которое мы определим так: 1-система состоит из одного отрезка длины, большей 1, $(k+1)$-система получается из объединения двух k-систем таких, что никакой отрезок одной из них не пересекается ни с каким отрезком другой, добавлением еще одного отрезка, содержащего все отрезки обеих k-систем. Ясно, что если система составляющих, отвечающая некоторому дереву T полного вывода в Б-грамматике, содержит k-систему, то T содержит k-дерево.

Пусть $\Gamma = \langle V, W, I, R \rangle$ — произвольная Б-грамматика, порождающая L_k $(k \geqslant 1)$. Нам достаточно теперь доказать, что для некоторой цепочки из L_k всякая система составляющих, сопоставляемая ей выводом в грамматике Г, содержит k-систему. При этом мы можем считать, аналогично предыдущим примерам, что Г является приведенной и не имеет правил вида $A \to B$, $A, B \in W$.

Пусть α и β — два узла некоторого дерева T полного вывода в Г, помеченные одним и тем же символом $A \in W$ и такие, что β зависит от α. Если v и y — составляющие, «происходящие» от α и β соответственно, то $v = xyz$, $A \vdash xAz$, откуда $A \vdash x^n A z^n$ для любого $n = 1, 2, \ldots$, причем хотя бы одна из цепочек x и z не пуста. Поэтому ввиду свойства VI $x = a^l$, $z = b^l$, $y = a^i v_1 v_2 b^j$, где v_1 и v_2 — гроздья.

Выберем совершенную гроздь w высоты k так, чтобы для любой ее подцепочки x, являющейся гроздью высоты, большей 0 и представимой в виде $x = ca^n t_1 t_2 b^n d$ (t_1 и t_2 — гроздья), число n было не меньше $2(p+1)\cdot g^{2(p+1)}$, где p — мощность W и g — максимум длин правых частей правил Г. Фиксируем некоторую систему составляющих C, сопоставляемую цепочке w некоторым выводом в Г, — соответствующее дерево вывода обозначим T — и рассмотрим одну из подцепочек-гроздьев цепочки w: $x = cut_1 t_2 vd$, где $u = a^n$, $v = b^n$, t_1 и t_2 — гроздья. По лемме III.1 в системе C найдется $p+1$ составляющих, образующих вложенную последовательность: $y_1 \supset y_2 \supset \ldots \supset y_{p+1}$ — и таких, что либо левые, либо правые концы их попарно различны и содержатся в отрезке u. Узлы $\alpha_1, \alpha_2, \ldots, \alpha_{p+1}$ дерева T, от которых «происходят» $y_1, y_2, \ldots, y_{p+1}$ соответственно, лежат на одном пути, и по крайней мере два из них — пусть y_f и y_h, $f < h$, — помечены одним и тем же символом. По предыдущему отсюда следует, что $y_f = a^l y_h b^l$, $y_h = a^i v_1 v_2 b^j$, где v_1, v_2 — гроздья. Теперь во всяком

случае ясно, что отрезок u содержит левый конец y_h, так что $y_h = a^m t'$, причем либо t' — начало t_1t_2, либо t_1t_2 — начало t'. Поскольку и t_1t_2, и v_1v_2 начинаются символом c, имеем $m = i$. Следовательно, либо t_1 — начало v_1, либо v_1 — начало t_1; но никакая гроздь не может быть собственным началом другой, так что $t_1 = v_1$, откуда точно так же $t_2 = v_2$. Итак, $y_h = a^i t_1 t_2 b^j$. Таким образом, каждой грозди x, являющейся подцепочкой w, отвечает составляющая $y(x) = y_h$, содержащаяся в x и содержащая всякую гроздь, являющуюся собственной подцепочкой x. Пусть теперь C' — множество всех составляющих из C, сопоставленных таким способом всевозможным подцепочкам w, являющимся гроздьями (включая самое w). Ясно, что C' есть k-система. Доказательство закончено.

Замечание. Все рассуждения, относящиеся к грамматике Г, очевидно, сохраняют силу для любой Б-грамматики, которая а) порождает все гроздья высоты k и б) не порождает ни одной цепочки, не являющейся гроздью. Мы воспользуемся этим ниже при разборе примера 2.

Назовем Б-грамматику, активная емкость которой мажорируется константой, Б-грамматикой ограниченной активной емкости, сокращенно Б-ОАЕ-грамматикой, а язык, порождаемый такой грамматикой, — ОАЕ-языком. Всякая Б-ОАЕВ-грамматика (§ 5.4) является Б-ОАЕ-грамматикой (причем, если $I_0(\Gamma) = k$, то $I_\Gamma(n) \leqslant k$). Грамматики Γ_k примера 1 являются не только ОАЕ-, но и ОАЕВ-грамматиками; однако легко видоизменить этот пример так, чтобы ни один из получаемых языков не был ОАЕВ-языком. Положим, например, $L'_k = L_k L_0$, где $L_0 = \{a^n b^n \mid n = 1, 2, \ldots\}^*$. Можно показать — это предоставляется читателю, — что ни один из L'_k не является ОАЕВ-языком (ср. упражнение 5.27). В то же время для каждого L'_k легко построить порождающую его Б-грамматику Γ'_k такую, что $I_{\Gamma'_k}(n) = k + 1$; кроме того, нетрудно убедиться, опираясь на соответствующий результат, уже имеющийся для языков L_k, что $L'_k \notin \mathscr{L}^I_k$(Б).

Таким образом, уже класс $\mathscr{L}^I_2$(Б) содержит языки, не являющиеся ОАЕВ-языками. (Что касается класса $\mathscr{L}^I_1$ (Б), то он, как уже отмечалось, совпадает с классом линейных языков.)

В общем случае для активной емкости можно получить более низкую верхнюю оценку, чем для глубины и разброса: имеет место

Теорема 7.4. *Для любой* Б-*грамматики* Г *справедливо неравенство* $I_\Gamma(n) \leqslant (r-1) \cdot (\log_2 n + 1)$, *где* r *есть максимум длин проекций правых частей правил* Г *на вспомогательный словарь, если этот максимум не меньше двух, и* $r = 2$ *в противном случае.*

Доказательство. Ради удобства индукции установим несколько более общий факт: если $\Gamma = \langle V, W, I, R\rangle$ — Б-грамматика, $A \in W$, $x \in V^+$ и $A \underset{\Gamma}{\vdash} x$, то найдется вывод x из A в Г, активная емкость которого не превосходит $(r-1)(\log_2|x| + 1)$. Ограничимся сначала случаем, когда $r \leqslant 2$, т. е. правая часть каждого правила Г содержит не более двух вхождений вспомогательных символов. (Грамматику, удовлетворяющую этому условию, мы будем называть слабо бинарной.) При $|x| = 1$ утверждение очевидно. Пусть оно доказано для цепочек длины, меньшей n, и пусть $|x| = n$ и D — произвольный вывод x из A. Пусть далее ω — первая цепочка вывода D, содержащая два вхождения вспомогательных символов (если такой цепочки нет, то все доказано): $\omega = tBuCv$, $B, C \in W$, $t, u, v \in V^*$. Тогда $x = tyuzv$, где $B \underset{\Gamma}{\vdash} y$, $C \underset{\Gamma}{\vdash} z$. При этом длина хотя бы одной из цепочек y, z — пусть y — не превосходит $n/2$. По индуктивному предположению можно найти вывод D' цепочки y из B и вывод D'' цепочки z из C, активные емкости которых не превосходят соответственно $\log_2|y| + 1 \leqslant \log_2 n$ и $\log_2|z| + 1 \leqslant \log_2 n + 1$. Производя сначала все шаги вывода D до получения цепочки ω, затем все шаги D' и, наконец, все шаги D'', будем иметь вывод x из A активной емкости, не превосходящей $\log_2 n + 1$.

Доказательство для общего случая получается теперь из следующего почти очевидного замечания: заменив в произвольной Б-грамматике $\Gamma = \langle V, W, I, R\rangle$ каждое правило $A \to z_0 B_1 z_1 B_2 z_2 \ldots z_{s-1} B_s z_s$, где $B_1, \ldots, B_s \in W$; $z_0, \ldots, z_s \in V^*$, $s > 2$, системой правил $A \to z_0 B_1 z_1 C_1$, $C_1 \to B_2 z_2 C_2$, $\ldots$, $C_{s-3} \to B_{s-2} z_{s-2} C_{s-2}$, $C_{s-2} \to B_{s-1} z_{s-1} B_s z_s$, где $C_1, \ldots, C_{s-2}$ — новые символы

(разные для разных правил), мы получим слабо бинарную грамматику Γ', эквивалентную Γ и такую, что $I_\Gamma(n) \leqslant (r-1)\cdot I_{\Gamma'}(n)$.

Оценка, доставляемая только что доказанной теоремой, не может быть понижена (во всяком случае, по порядку): существуют Б-языки, не порождаемые никакими Б-грамматиками с активной емкостью, растущей медленнее логарифмической функции (и тем более не являющиеся ОАЕ-языками). Приведем пример.

Пример 2. Положим $L = \bigcup_{k=0}^{\infty} L_k$, где L_k — языки примера 1. Язык L порождается Б-грамматикой со схемой $\{I \to cAd,\ I \to cd,\ A \to aAb,\ A \to aIIb\}$. Пусть $\Gamma = \langle V, W, I, R\rangle$ — произвольная порождающая L Б-грамматика, p — мощность W и g — максимум длин правых частей правил Γ. Можно считать Γ, подобно предыдущему, приведенной и не имеющей правил вида $A \to B$, $A,\ B \in W$. Как и в примере 1, мы можем утверждать (см. замечание на стр. 234), что если цепочка $w \in L_k$, $k \geqslant 1$, выбрана так, чтобы для любой ее подцепочки x, являющейся гроздью высоты, большей 0: $x = ca^n t_1 t_2 b^n d$ (t_1 и t_2 — гроздья), число n было не меньше $2(p+1)\ g^{2(p+1)}$, то активная емкость вывода w из I в Γ не может быть меньше k. Обозначив через l_k наименьшую длину цепочки из L_k, удовлетворяющей указанному условию, и полагая $4(p+1)\cdot g^{2(p+1)} + 2 = h$, имеем $l_1 = h+4$, $l_{k+1} = 2l_k + h$, откуда при $k \geqslant 2$ получается $l_k = (2^k - 1)h + 2^{k+1} < 2^{k+1}(h+1)$, так что $k > \log_2 l_k - \log_2(h+1) - 1$. Поэтому для бесконечного числа значений n (именно, $n = l_2, l_3, \ldots$) справедливо неравенство $I_\Gamma(n) \geqslant \log_2 n - \log_2(h+1) - 1$, так что во всяком случае отношение $I_\Gamma(n)/\log_2 n$ не может стремиться к пределу, меньшему единицы. Более того, можно найти такую постоянную c, что $I_\Gamma(n) \geqslant \log_2 n - c$. Действительно, при любом $k = 2, 3, \ldots$ имеем, во всяком случае, $l_{k+1} < 2^r l_k$, где $r = \log_2(h+2)$; поэтому при $l_k \leqslant n < l_{k+1}$ получаем

$$I_\Gamma(n) \geqslant I_\Gamma(l_k) \geqslant \log_2 l_k - \log_2(h+1) - 1 > \log_2(l_{k+1}) - r - \log_2(h+1) - 1 > \log_2 n - r - \log_2(h+1) - 1.$$

В заключение параграфа мы укажем характеристику ОАЕ-языков в алгебраических терминах, сходную с полученной в § 5.4 для ОАЕВ-языков. Для этого введем одно новое понятие, представляющее и самостоятельный интерес.

Пусть T — конечное дерево. Сопоставим каждому его узлу α натуральное число $\mu(T, \alpha)$, которое будем называть густотой этого узла, следующим образом. I. Густота висячего узла равна нулю. II. Пусть для всех узлов $\beta_1, \ldots, \beta_s$, подчиненных узлу α, числа $\mu(T, \beta_1), \ldots, \mu(T, \beta_s)$ определены; тогда: а) если $\mu(T, \beta_1) = \ldots = \mu(T, \beta_s) = 0$, то $\mu(T, \alpha) = 1$; б) если $\max \mu(T, \beta_i) = m > 0$, то: б1) в случае, когда существует только одно $i = 1, \ldots, s$, для которого $\mu(T, \beta_i) = m$, полагаем $\mu(T, \alpha) = m$; б2) в противном случае $\mu(T, \alpha) = m + 1$.

Далее, густотой дерева T (обозначение: $\mu(T)$) будем называть густоту его корня.

Из определения ясно, что в дереве густоты m все узлы, имеющие ту же густоту m, образуют путь (возможно, нулевой длины), исходящий из корня. Мы будем называть этот путь старшим.

Лемма 7.2. *Если T — дерево полного вывода в слабо бинарной Б-грамматике, то наименьшая активная емкость вывода, отвечающего этому дереву, равна густоте T.*

Доказательство. Ради удобства индукции будем проводить его для любых (A, x)-деревьев, где A — вспомогательный символ и x — цепочка основных символов. Для дерева высоты 1 утверждение очевидно. Пусть оно доказано для деревьев высоты, меньшей n, и пусть высота T равна n, а густота равна m. Если среди узлов, подчиненных корню T, только один помечен вспомогательным символом и T' — полное поддерево T с корнем в этом узле, то наименьшая активная емкость выводов, отвечающих T, будет такова же, как для выводов, отвечающих T', а последняя равна $\mu(T') = m$. Пусть имеются два узла, подчиненных корню T и помеченных вспомогательными символами, и пусть T', T'' — полные поддеревья T с корнями в этих узлах. Возможны два случая: а) $\mu(T') = \mu(T'') = m - 1$; б) густота одного из деревьев T', T'' — пусть T' — равна m, а

густота второго меньше m. В обоих случаях наименьшая активная емкость вывода, отвечающего T, получится, если сначала выполнить вывод наименьшей активной емкости, отвечающий T'', а потом — отвечающий T' (впрочем, в случае а) можно и в обратном порядке); и в обоих случаях в силу индуктивного предположения получится вывод активной емкости m.

Перейдем теперь к алгебраической характеристике ОАЕ-языков. Эту характеристику можно сформулировать очень просто: класс ОАЕ-языков совпадает с замыканием класса линейных Б-языков относительно подстановки. Мы, однако, постараемся уточнить формулировку так, чтобы она стала эффективной. Для этого введем следующее определение.

Будем называть подстановочным выражением от (переменных) $\xi_1, \ldots, \xi_n$ всякое выражение, составленное из абстрактных символов $\xi_1, \ldots, \xi_n$ с помощью знаков подстановки (в обозначениях подстановок должны участвовать также некоторые элементарные символы, причем каждой переменной, выступающей в роли языка, в которой производится подстановка, сопоставляется определенный набор элементарных символов; ср. определение центрально-подстановочного выражения, являющееся частным случаем настоящего определения, на стр. 177). Например, если $\xi_1, \ldots, \xi_8$ — переменные и a, b, c, d, e — элементарные символы, то $S(\xi_1; a, b, c|\xi_2, S(\xi_1; a, b, c|\xi_3, \xi_4, \xi_5), S(\xi_4; b, d, e|S(\xi_7; a|\xi_8), \{d\}, \{e\}))$ — подстановочное выражение; переменным ξ_1, ξ_4 и ξ_7 здесь сопоставлены наборы $\{a, b, c\}$, $\{b, d, e\}$ и $\{a\}$ соответственно. Обычным способом определим также представление языка с помощью подстановочного выражения.

Теорема 7.5. а) *Для любого подстановочного выражения* $\mathfrak{A}(\xi_1, \ldots, \xi_n)$ *и любых* n *линейных* Б-*грамматик* $\Gamma_1, \ldots, \Gamma_n$ *можно построить* Б-ОАЕ-*грамматику, порождающую язык* $\mathfrak{A}(L(\Gamma_1), \ldots, L(\Gamma_n))$. б) *Для любой* Б-ОАЕ-*грамматики* Γ *можно, если известно число, мажорирующее* $I_\Gamma(n)$, *построить подстановочное выражение* $\mathfrak{A}(\xi_1, \ldots, \xi_n)$ *и линейные* Б-*грамматики* $\Gamma_1, \ldots, \Gamma_n$ *такие, что* $L(\Gamma) = \mathfrak{A}(L(\Gamma_1), \ldots, L(\Gamma_n))$.

Доказательство. а) Достаточно показать, что по любым Б-ОАЕ-грамматикам Γ, $\Gamma_1, \ldots, \Gamma_s$ можно

построить Б-ОАЕ-грамматику, порождающую язык $S(L(\Gamma); a_1, \ldots, a_s|L(\Gamma_1), \ldots, L(\Gamma_s))$. Но если $\Gamma = \langle V, W, I, R\rangle$, $\Gamma_i = \langle V_i, W_i, I_i, R_i\rangle$ $(i = 1, \ldots, s)$, $V = \{a_1, \ldots, a_s\}$, словари $V \cup W$, $W_1, \ldots, W_s$ попарно не пересекаются и при этом $I_\Gamma(n) \leqslant k$, $I_{\Gamma_i}(n) \leqslant k_i$ $(i = 1, \ldots, s)$, то, положив $\Gamma' = \langle V_1 \cup \ldots \cup V_s, W \cup W_1 \cup \ldots \cup W_s, I, \overline{R} \cup R_1 \cup \ldots \cup R_s\rangle$, где $\overline{R}$ получается из R заменой во всех правилах всех вхождений $a_1, \ldots, a_s$ на $I_1, \ldots, I_s$ соответственно, будем иметь, очевидно, $I_{\Gamma'}(n) \leqslant k + \max(k_1, \ldots, k_s) - 1$. В то же время ясно, что Γ' порождает нужный язык.

б) Пусть $\Gamma = \langle V, W, I, R\rangle$ — Б-грамматика такая, что $I_\Gamma(n) \leqslant M$. Замечание, сделанное в конце доказательства теоремы 7.4, позволяет при соответствующем увеличении M считать Γ слабо бинарной, так что M будет также верхней границей густот деревьев полных выводов в Γ (лемма 7.2). Можно также считать, что в Γ нет правил вида $A \to B$, $A, B \in W$.

Сопоставим каждой паре (A, m), где $A \in W$ и $m = 1, \ldots, M$, новый символ A_m, а также переменную $\xi_{A,m}$; число m назовем уровнем символа A_m и переменной $\xi_{A,m}$. Кроме того, нам удобно будет полагать $a_0 = a$ для каждого $a \in V$ и приписывать символам из V уровень 0.

Каждому правилу $r \in R$, содержащему в правой части вхождения вспомогательных символов, и каждому $m = 2, \ldots, M$ сопоставим систему новых правил следующим образом: если $r = A \to xBy$, $B \in W$, $x, y \in V^*$, то система состоит из одного правила $A_m \to xB_m y$; если $r = xByCz$, $B, C \in W$, $x, y, z \in V^*$, то система состоит из всевозможных правил вида $A_m \to xB_{m_1}yC_{m_2}z$, где либо $m_1 = m_2 = m - 1$, либо одно из чисел m_1, m_2 равно m, а другое меньше m. Число m назовем уровнем правил построенной системы. Кроме того, каждому правилу вида $A \to xBy$, $B \in W$, $xy \in V^+$, и вида $A \to x$, $x \in V^+$, сопоставим новое правило $A_1 \to xB_1 y$, соответственно $A_1 \to x$; этим правилам будет приписываться уровень 1. Множество всех символов (правил) уровня m будем обозначать V_m (соответственно R_m).

Далее сопоставляем каждой паре (A, m), $A \in W$, $m = 1, \ldots, M$, грамматику $\Gamma_{A,m} = \langle V_0 \cup \ldots \cup V_{m-1}, V_m,$

A_m, R_m⟩; число m будем называть ее уровнем. Из определения правил уровня m ясно, что все $\Gamma_{A,m}$ линейны.

Поскольку $L(\Gamma) = L_1 \cup \ldots \cup L_M$, где L_m, $m = 1, \ldots, M$, состоит из тех цепочек из $L(\Gamma)$, для которых существуют деревья полных выводов в Γ, имеющие густоту m, нам достаточно представить в требуемом виде каждый из языков $L_1, \ldots, L_M$ (поскольку объединение тривиальным образом сводится к подстановке в конечный язык). Удобнее доказывать несколько более общий факт, а именно представимость в том же виде каждого языка $L_{A,m}$ ($A \in W$, $m = 1, \ldots, M$), состоящего из тех цепочек x, для которых в Γ существуют (A, x)-деревья густоты m.

Мы покажем, что язык $L_{A,m}$ можно получить следующим образом: сначала подставим в язык $L(\Gamma_{A,m}) \subseteq (V_0 \cup \ldots \cup V_{m-1})^*$ вместо каждого символа $B_{m'}$, $1 \leqslant m' \leqslant m-1$, соответствующий язык $L(\Gamma_{B,m'})$; в полученный язык (в словаре $V_0 \cup \ldots \cup V_{m-2}$) вместо каждого символа $C_{m''}$, $1 \leqslant m'' \leqslant m-2$, подставим язык $L(\Gamma_{C,m''})$ и т. д., пока не получим язык в словаре $V_0 = V$. Соответствующее подстановочное выражение читатель без труда выпишет.

1) Если цепочка x принадлежит описанному только что языку — будем обозначать его $L'_{A,m}$, — то она может быть выведена из A_m с помощью новых правил уровней $1, \ldots, m$. Устранив во всех цепочках полученного таким образом вывода индексы $1, \ldots, m$, мы получим, очевидно, вывод x из A в Γ.

2) Включения $L_{A,m} \subseteq L'_{A,m}$ докажем индукцией по m. Ясно, прежде всего, что деревья вывода густоты 1 — это в точности такие деревья, что в отвечающих им выводах используются лишь правила, содержащие в правых частях не более чем по одному вхождению вспомогательного символа; а эти правила совпадают — с точностью до индекса 1 при вспомогательных символах — с новыми правилами первого уровня. Поэтому $L_{A,1} \subseteq L'_{A,1}$ для любого $A \in W$. Допустим, что $m > 1$ и утверждение доказано для густот, меньших m. Рассмотрим (A, x)-дерево T густоты m ($A \in W$, $x \in V^+$). Пусть $\alpha_0, \ldots, \alpha_t$ — старший путь в T и $\beta_0, \ldots, \beta_u$ ($u \leqslant t+1$) — все (упорядоченные «сверху вниз») узлы T, подчиненные узлам старшего пути, не лежащие на нем и

помеченные вспомогательными символами *) (рис. 14). Для каждого $j = 0, \ldots, u$ будем обозначать через T_j полное β_j-поддерево дерева T. Положим также $m_j = \mu(T_j) = \mu(T, \beta_j)$; для любого j имеем $m_j < m$.

Каждому узлу α_i старшего пути отвечает некоторое правило $r_i = A_i \to \omega_i$ грамматики Γ, где A_i — метка при α_i и цепочка ω_i содержит, если $i < t$, одно или два вхождения вспомогательных символов (один из них является меткой при α_{i+1}, а другой, если он имеется, — меткой при некотором β_j, $j \leqslant i$), а если $i = t$ — точно два вхождения (являющиеся метками при β_{u-1} и β_u). Обозначим через r'_i правило, которое получится из r_i, если приписать левой части индекс m, а в правой части: а) при $i < t$ приписать тому вхождению вспомогательного символа, которое отвечает узлу α_{i+1}, индекс m, а вхождению, отвечающему узлу β_j (если оно есть), индекс m_j; б) при $i = t$ приписать каждому из двух вхождений вспомогательных символов индекс $m-1$ (заметим, что $m_{u-1} = m_u = m-1$). Очевидно, все r'_i, $i = 1, \ldots, t$, — правила грамматики $\Gamma_{A,m}$.

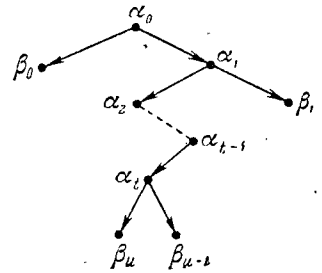


Рис. 14.

Среди отвечающих дереву T выводов имеется такой, в котором на первых $t+1$ шагах применяются правила $r_0, r_1, \ldots, r_t$ (в этом порядке) и притом как раз в точках, соответствующих узлам $\alpha_0, \ldots, \alpha_t$. После выполнения этих шагов получается цепочка ω, содержащая $u+1$ вхождений вспомогательных символов, а именно символов, являющихся метками при $\beta_0, \ldots, \beta_u$. Заменяя в ω каждое вхождение вспомогательного символа $B^{(j)}$, отвечающее узлу β_j, вхождением символа $B^{(j)}_{m_j}$, получим цепочку ω', которая, очевидно, получается из A_m последовательным применением правил $r'_0, r'_1, \ldots, r'_t$ и, следовательно, принадлежит $L(\Gamma_{A,m})$. Но цепочка x

*) Каждому из узлов $\alpha_0, \ldots, \alpha_{t-1}$ может быть подчинен самое большее один узел, помеченный вспомогательным символом и не лежащий на старшем пути; для α_t таких узлов ровно два.

получается из ω, а значит и из ω', заменой каждого вхождения вспомогательного символа, отвечающего узлу β_j, на цепочку $z_j = \mu(T_j)$; а поскольку T_j есть $(B^{(j)}, z_j)$-дерево густоты $m_j < m$, имеем $z_j \in L'_{B^{(j)}, m_j}$. Однако язык, получающийся из $L(\Gamma_{A,m})$ подстановкой языков $L'_{B,m'}$ вместо символов $B_{m'}$, $m' < m$, — это и есть $L'_{A,m}$.

Замечание. В настоящем параграфе, как и во всей главе, мы ограничились случаем Б- (а не ОБ-) грамматик, чтобы избежать дополнительного усложнения рассуждений, дающего не слишком большой выигрыш в общности. Однако соответствующие обобщения могут быть сделаны без труда.

§ 7.3. Степень гнездования и степень самовставления

В § 3.1 мы ввели НС-сигнализирующие — специфические характеристики сложности вывода в НС-грамматиках, при определении которых существенным образом используется представление вывода в виде дерева. Там же были определены три конкретных типа НС-сигнализирующих: $Y^{л}_{\Gamma}$, $Y^{п}_{\Gamma}$ и Φ_{Γ}. Для Б-грамматик, как мы видели в § 7.1, функции $Y^{л}_{\Gamma}$ и $Y^{п}_{\Gamma}$ совпадают, с точностью до аддитивной постоянной, с левой и правой глубиной соответственно. Нам остается изучить (для случая Б-грамматик) поведение функции $\Phi_{\Gamma}(n)$, которую мы будем называть степенью гнездования грамматики Г. Параллельно с ней мы будем рассматривать другую функцию — степень самовставления (обозначение: $\mathrm{X}_{\Gamma}(n)$), — определяемую следующим образом: (i) назовем гнездо, содержащееся в системе составляющих, отвечающей дереву вывода в Б-грамматике, однородным, если все входящие в него составляющие «происходят» от узлов с одинаковыми метками (причем, если какая-нибудь составляющая «происходит» от нескольких узлов, то достаточно, чтобы один из них имел нужную метку); (ii) степенью самовставления дерева полного вывода T в Б-грамматике Г (обозначение: $\tilde{\chi}(\Gamma, T)$) назовем наибольшую глубину однородного гнезда, содержащегося в отвечающей T системе составляющих; (iii) для произвольной цепочки $x \in L(\Gamma)$ положим $\tilde{\mathrm{X}}(\Gamma, x) = \min \tilde{\chi}(\Gamma, T)$ где минимум берется по всем деревьям вывода цепочки x из начального символа в Г; (iv) наконец, $\mathrm{X}_{\Gamma}(n) = \max_{x \in L(\Gamma);\, |x| \leqslant n} \tilde{\mathrm{X}}(\Gamma, x)$.

Из определений ясно, что $\mathrm{X}_{\Gamma}(n) \leqslant \Phi_{\Gamma}(n)$. В то же время, если p — мощность вспомогательного словаря Г, то во всяком гнезде глубины, не меньшей pn, содержащемся в отвечающей дереву вывода в Г системе составляющих, будет содержаться, в свою очередь, однородное гнездо глубины, не меньшей n; поэтому $\Phi_{\Gamma}(n) \leqslant p\mathrm{X}_{\Gamma}(n)$.

Как было уже замечено (стр. 84),

$$\Phi_{\Gamma}(n) \leqslant \min\left(Y^{л}_{\Gamma}(n),\ Y^{п}_{\Gamma}(n)\right).$$

Поэтому из теоремы 7.1 и ее аналога для правой глубины следует, что для Б-грамматик $\Phi_{\Gamma}(n) \leqslant \min\left(\mathcal{Y}^{л}_{\Gamma}(n),\ \mathcal{Y}^{п}_{\Gamma}(n)\right) + e - 1$, где e — максимум длин цепочек x в основном словаре Г, для которых в Г имеются правила вида $A \to x\theta$ или $A \to \theta x$, причем $\theta \neq \Lambda$ *). В частности, для стандартной бинарной Б-грамматики $\Phi_{\Gamma}(n) \leqslant \min\left(\mathcal{Y}^{л}_{\Gamma}(n),\ \mathcal{Y}^{п}_{\Gamma}(n)\right) - 1$.

В силу определения гнезда имеем (для любой НС-грамматики) $\Phi_{\Gamma}(n) \leqslant \frac{1}{2}(n-1)$; это неравенство может переходить в равенство уже для линейных грамматик (например: $\{I \to aIa,\ I \to a\}$). Для любой Б-грамматики Г и любого действительного числа c, $0 < c \leqslant 1$, можно построить Б-грамматику Г′, эквивалентную Г и такую, что (для достаточно больших n) $\Phi_{\Gamma'}(n) \leqslant cn$ (в силу аналогичного факта для левой глубины, стр. 225). Вместе с тем даже для линейного языка может оказаться, что степень гнездования (а значит, и степень самовставления) каждой порождающей его Б-грамматики мажорирует некоторую линейную функцию. В частности, это верно для языка примера 1 из § 7.1. Действительно, мы можем, как и в этом примере, ограничиться приведенными грамматиками без правил вида $A \to B$, $A, B \in W$ (ни устранение таких правил, ни переход к приведенной грамматике не меняют степеней гнездования деревьев полных выводов); но при разборе примера мы видели,

*) Впрочем, аналогичный факт верен и для произвольных НС-грамматик — см. упражнение 3.5, а).

что при любом n для всякого дерева вывода цепочки $a^n b^n$ соответствующая система составляющих будет содержать последовательность составляющих $z_1, \ldots, z_h$ такую, что: а) $h \geqslant c \cdot 2n$, где c — константа, зависящая только от грамматики; б) $z_i = a^{k_i} b^{l_i}$, причем $k_i - k_{i+1} = l_i - l_{i+1} > 0$ для всех $i = 1, \ldots, h-1$. Однако из условия б) вытекает, что данная последовательность является гнездом.

Теперь нам остается исследовать строение множеств $L_k^{\Phi}(\Gamma)$ и $L_k^{X}(\Gamma)$ (смысл этих обозначений ясен по аналогии с введенными на стр. 61). Картина здесь сходна с той, которая была получена для глубины: имеет место

Теорема 7.6. *Для любой* Б-*грамматики* Γ *и любого натурального числа* k *множества* $L_k^{\Phi}(\Gamma)$ *и* $L_k^{X}(\Gamma)$ *являются* А-*языками. Более того, для всякой* Б-*грамматики* Γ *и всякого натурального* k *можно построить* А-*грамматики, порождающие* $L_k^{\Phi}(\Gamma)$ *и* $L_k^{X}(\Gamma)$ *соответственно.*

Доказательству предпошлем лемму. Будем называть вспомогательный символ A Б-грамматики Γ самовставляющимся, если имеются такие непустые цепочки ξ и η, что $A \underset{\Gamma}{\vdash} \xi A \eta$. Б-грамматику, у которой нет самовставляющихся символов, назовем Б-грамматикой без самовставления.

Лемма 7.3. *Для всякой* Б-*грамматики без самовставления можно построить эквивалентную ей* А-*грамматику.*

Доказательство. Пусть $\Gamma = \langle V, W, I, R\rangle$ — Б-грамматика без самовставления и $p = \mu(W)$. Каков бы ни был символ $A \in W$, из A не могут быть одновременно выводимы цепочки вида ξA и $A\eta$, где $\xi \neq \Lambda$, $\eta \neq \Lambda$; и если, например, $A \underset{\Gamma}{\vdash} \xi A$, $\xi \neq \Lambda$, то ξ не содержит вхождений A. Поэтому, в частности, если $p = 1$, т. е. $W = \{I\}$, то Γ является либо правосторонней, либо левосторонней линейной грамматикой, и можно построить эквивалентную ей автоматную грамматику (стр. 169). Пусть утверждение доказано для грамматик с менее чем p вспомогательными символами. Пусть для определенности R не содержит правил вида $I \to I\xi$, $\xi \neq \Lambda$. Сопоставим каждому символу $A \in W$ новый символ $\bar{A}$ и положим $\overline{W} = \{\bar{A} \mid A \in W\}$. Положим далее

$\bar{\Gamma} = \langle V \cup (W - \{I\}), \overline{W}, I, \bar{R}\rangle$, где $\bar{R}$ состоит из всевозможных правил вида $\bar{A} \to \varphi \bar{B}$ таких, что $A \to \varphi B \in R$, и правил вида $\bar{I} \to \psi$ таких, что $I \to \psi \in R$ и ψ не содержит вхождений I. Кроме того, для каждого $A \in W - \{I\}$ будем полагать $\Gamma_A = \langle V, W - \{I\}, A, R'\rangle$, где R' получается из R изъятием правил, содержащих в левой или правой части вхождения I. Для каждой грамматики Γ_A можно по индуктивному предположению построить эквивалентную ей А-грамматику, и то же верно для правосторонней линейной грамматики $\bar{\Gamma}$, однако поскольку в цепочках вывода в Γ, начинающегося символом I, этот символ может стоять только на последнем месте, язык $L(\Gamma)$ получается из $L(\bar{\Gamma})$ подстановкой вместо каждого $A \in W - \{I\}$ языка $L(\Gamma_A)$, и мы можем воспользоваться теоремой 5.4.

Доказательство теоремы 7.6. Пусть $\Gamma = \langle V, W, I, R\rangle$ — Б-грамматика. Мы можем считать, что она не имеет правил вида $A \to B$, $B \in W$. Кроме того, нам удобно будет полагать, что основные символы могут содержаться лишь в правых частях правил вида $A \to a$, $a \in V$; действительно, если ввести для каждого $a \in V$ «двойника» — новый символ $\bar{a}$, заменить в правых частях всех правил из R не вида $A \to a$, $a \in V$, все вхождения основных символов вхождениями их «двойников» и добавить к R всевозможные правила вида $\bar{a} \to a$, то это не изменит ни порождаемого грамматикой языка, ни ее степени гнездования, ни степени самовставления.

Введем теперь для каждого символа $A \in W$ $5(k+1)$ новых символов: $A^{\text{ЛН},i}$ («левые неуглубляющие» символы), $A^{\text{ЛУ},i}$ («левые углубляющие»), $A^{\text{ПН},i}$ («правые неуглубляющие»), $A^{\text{ПУ},i}$ («правые углубляющие»), $A^{\text{С},i}$ («средние»); здесь $0 \leqslant i \leqslant k$. Каждому правилу $A \to BE_1 \ldots E_s D$, где $B, E_1, \ldots, E_s, D \in W$, $s \geqslant 0$, сопоставим $5k$ новых правил:

$$\left.\begin{array}{l} A^{\text{ЛН},i} \to B^{\text{ЛН},i} E_1^{\text{С},i+1} \ldots E_s^{\text{С},i+1} D^{\text{ПУ},i+1} \\ A^{\text{ПН},i} \to B^{\text{ЛУ},i+1} E_1^{\text{С},i+1} \ldots E_s^{\text{С},i+1} D^{\text{ПН},i} \\ A^{\text{ЛУ},i} \\ A^{\text{ПУ},i} \\ A^{\text{С},i} \end{array}\right\} \to B^{\text{ЛН},i} E_1^{\text{С},i+1} \ldots E_s^{\text{С},i+1} D^{\text{ПН},i} \Bigg\} i = 0, \ldots, k-1.$$

Кроме того, каждому правилу $A \to a$, где $a \in V$, сопоставим всевозможные правила вида $A^{Ъ, i} \to a$, где

$$Ъ = \text{ЛН, ЛУ, ПН, ПУ, С и } i = 0, \ldots, k.$$

Положим теперь $\Gamma' = \langle V, W', I^{\text{С}, 0}, R' \rangle$, где W' состоит из всех новых символов и R' — из всех новых правил.

Из самого вида новых правил непосредственно усматривается, что при любом $i = 0, \ldots, k-1$ цепочка, выводимая в Γ' из символа $A^{\text{ЛН}, i}$, соответственно $A^{\text{ПН}, i}$, может содержать вхождение того же символа только на первом, соответственно последнем, месте; символы же $A^{\text{ЛУ}, i}$, $A^{\text{ПУ}, i}$, $A^{\text{С}, i}$ $(i = 0, \ldots, k)$, а также $A^{\text{ЛН}, k}$ и $A^{\text{ПН}, k}$ не являются даже циклическими. Поэтому Γ' — грамматика без самовставления, и по лемме 7.3 можно построить эквивалентную ей А-грамматику.

Вспомнив индуктивное определение степени гнездования составляющей (стр. 289), можно без всякого труда показать индукцией по i, что в любом дереве полного вывода в Γ' степень гнездования каждой составляющей, «происходящей» от узла, помеченного символом с индексом i, равна i; поэтому степени гнездования деревьев полных выводов в Γ' (точнее, отвечающих им систем составляющих) не превосходят k. В то же время, если в дереве вывода в Γ' лишить все символы-метки индексов, получится дерево вывода в Γ, имеющее, очевидно, ту же степень гнездования; следовательно, $L(\Gamma') \subseteq L_k^{\Phi}(\Gamma)$. С другой стороны, легко показать, что во всяком дереве полного вывода в Γ, степень гнездования которого не превосходит k, можно так снабдить символы-метки индексами, чтобы получилось дерево полного вывода в Γ' (следует приписывать индексы «сверху вниз», переходя от каждого узла к подчиненным ему). Отсюда $L_k^{\Phi}(\Gamma') \subseteq L(\Gamma')$.

Займемся теперь степенью самовставления. Пусть $W = \{H_1, \ldots, H_p\}$. Каждому $A = H_i \in W$ мы сопоставим систему новых символов, которые будут записываться в виде $A^{Ъ\binom{s_1 \ldots s_p}{i_1 \ldots i_p}}$, где $Ъ = \text{ЛН, ЛУ, ПН, ПУ, С}$; $s_j = 0, 1$; $i_j = 0, \ldots, k$. (Содержательный

смысл Ъ — тот же, что и в первой части доказательства; i_j означает «степень H_j-самовставления» составляющей, происходящей от узла с меткой A, т. е. наибольшую глубину гнезда, все члены которого «происходят» от узлов с меткой H_j и содержат данную составляющую; s_j — вспомогательный «индекс готовности», смысл которого будет ясен из дальнейшего.) Далее, каждому правилу $r = A \to \omega \in R$, где $\omega = BE_1 \ldots E_sD$, $B, E_1, \ldots, E_s, D \in W$, $s \geqslant 0$, будет сопоставляться система новых правил, каждое из которых отличается от r только наличием индекса Ъ и матрицы $\binom{s_1 \ldots s_p}{i_1 \ldots i_p}$ при каждом вхождении символа в левую и правую часть; при этом в левой части возможны любые комбинации индексов и матриц, а в правой индексы и матрицы выбираются следующим образом.

I. Значение Ъ для каждого вхождения символа в ω выбирается в зависимости от расположения этого вхождения и от значения того же индекса при A точно так же, как в первой части доказательства.

II. Правила выбора значений s_j:

1) Для всех средних вхождений $s_j = 1$ $(j = 1, \ldots, p)$.

2) Если $A = H_j$ или значение s_j при A равно 0, то неуглубляющие левые и правые вхождения символов в ω получают значения s_j, равные 0, а углубляющие — равные 1.

3) Если $A \neq H_j$ и значение s_j при A равно 1, то все вхождения символов в ω получают значения s_j, равные 1.

III. Правила выбора значений i_j:

1) Если $A = H_j$, то значения i_j для тех вхождений символов в ω, для которых $s_j = 0$, такие же, как для A, а для тех, для которых $s_j = 1$, — на единицу больше.

2) Если $A \neq H_j$, то для всех вхождений символов в ω значения i_j такие же, как для A.

При этом, если $A = H_j$ и значение i_j для A равно k, то те из описанных выше правил с соответствующей левой частью, у которых в правой части имеются вхождения H_j со значениями s_j, равными 1, изымаются из числа сопоставляемых правил.

Кроме того, каждому правилу вида $A \to a$, где $a \in V$, сопоставляем всевозможные правила вида $A^{Ъ}\begin{pmatrix} s_1 & \dots & s_p \\ i_1 & \dots & i_p \end{pmatrix} \to a$.

Положим $\Gamma' = \left\langle V, W', I^{C}\begin{pmatrix} 0 & \dots & 0 \\ 0 & \dots & 0 \end{pmatrix}, R' \right\rangle$, где W' состоит из всех новых символов и R' — из всех новых правил.

Из приведенного только что описания новых правил легко усматривается, что:

а) если левая часть правила имеет вид $H_j^{\text{ЛН}}\begin{pmatrix} s_1 & \dots & s_p \\ i_1 & \dots & i_p \end{pmatrix}$, то у всех вхождений символов в правую часть, кроме первого (также имеющего ЛН в качестве значения Ъ) значения i_j больше, чем в левой части, а значения $i_1, \dots, i_{j-1}, i_{j+1}, \dots, i_p$ такие же; б) для правил с левой частью $H_j^{\text{ПН}}\begin{pmatrix} s_1 & \dots & s_p \\ i_1 & \dots & i_p \end{pmatrix}$ рассмотрение симметрично; в) если левая часть имеет вид $H_j^{Ъ}\begin{pmatrix} s_1 & \dots & s_p \\ i_1 & \dots & i_p \end{pmatrix}$, где Ъ равно ЛУ, ПУ или С, то в правой части первое и последнее вхождения символов имеют в качестве значений Ъ ЛН и ПН соответственно, и значения $i_1, \dots, i_p$ у этих вхождений такие же, как в левой части, а у средних вхождений значения i_j больше, чем в левой части, и значения остальных i такие же. Отсюда, в свою очередь, ясно, что символы, для которых Ъ равно ЛУ, ПУ или С, не циклические и что в цепочке, выводимой из символа, для которого Ъ = ЛН (соответственно Ъ = ПН), вхождение того же символа может стоять только на первом (соответственно последнем) месте. Таким образом, Γ' есть грамматика без самовставления.

Нетрудно, далее, показать, исходя опять-таки из самого определения новых правил, что если α — произвольный узел дерева полного вывода в Γ', β — узел, подчиненный α, $t_{\alpha j}$ — «степень H_j-самовставления» узла α (т. е. наибольшая глубина гнезда, все члены которого «происходят» от узлов, несущих пометку H_j — с разными, вообще говоря, индексами и матрицами, — и содержат составляющую, «происходящую» от α) и $t_{\beta j}$ — «степень H_j-самовставления» узла β, то $t_{\beta j} = t_{\alpha j}$, когда значения i_j для меток при α и β равны, и $t_{\beta j} = t_{\alpha j} + 1$ в противном случае (т. е. когда значение i_j для метки при β на единицу больше, чем для метки при α). Отсюда очевидной индукцией получается, что максимум «степеней H_j-самовставления» узлов дерева полного вывода в Γ', взятый по всем узлам и по всем $j = 1, \dots, p$, равен наибольшему из значений $i_1, \dots, i_p$ для меток в узлах данного дерева, так что ни для какого дерева полного вывода в Γ' этот максимум — обозначим его M — не может превосходить k. В то же время, лишив в таком дереве все символы-метки индексов и матриц, получим дерево полного вывода в Γ, степень самовставления которого равна M. Поэтому $L(\Gamma') \subseteq L_k^{X}(\Gamma)$. С другой стороны, нетрудно показать, что в дереве полного вывода в Γ, степень самовставления которого не превосходит k, можно так снабдить символы-метки индексами и матрицами, чтобы получилось дерево полного вывода в Γ'; отсюда $L_k^{X}(\Gamma) \subseteq L(\Gamma')$. Доказательство закончено.

Следствие. а) *Для любой* Б-*грамматики* Γ, *для которой* $\Phi_\Gamma(n)$ *ограничена числом* k, *можно, если известно* k, *построить эквивалентную ей* А-*грамматику.* б) *Аналогично для* $X_\Gamma(n)$.

Упражнения

7.1. а) Показать, что левая глубина и разброс любой Б-грамматики, порождающей язык Дика, мажорируют подходящие линейные функции.

б) То же для скобочного языка.

7.2. Показать, что левая глубина любой однозначной Б-грамматики, порождающей неавтоматный язык, мажорирует подходящую линейную функцию.

7.3. Показать, что для произвольных Б-грамматик Γ_1 и Γ_2 можно построить Б-грамматику Γ, порождающую язык $L(\Gamma_1)L(\Gamma_2)$ и такую, что $D_\Gamma(n) \leqslant \max\left(\mathcal{Y}_{\Gamma_1}^{\text{Л}}(n), \mathcal{Y}_{\Gamma_2}^{\text{П}}(n)\right)$.

7.4. Определим «центральную глубину» и «краевую глубину» цепочки $\omega \in (V \cup W)^+$ соответственно как наименьшее значение левой и правой глубины ω и как наименьшее число n, для которого ω представима в виде $\omega_1 x \omega_2$, где $x \in V^+$ и $|\omega_1| + |\omega_2| = n$. Далее обычным способом определим центральную и краевую глубину грамматики.

Показать, что для центральной и краевой глубины справедливы аналоги теоремы 7.2.

7.5. а) Показать, что для любой НС-грамматики Г и любого действительного числа c, $0 < c < 1$, можно построить НС-грамматику Г′, эквивалентную Г и такую, что $\mathcal{Y}^{Л}_{Г'}(n) \leqslant cn$.

б) Можно ли распространить на случай произвольных НС-грамматик теорему 7.4?

7.6. Показать, что для любого $k = 1, 2, \ldots$ и любого $l = k+1, k+2, \ldots$ можно построить такую Б-грамматику Г, что $L(Г) \in \mathscr{L}^{l}_{k+1}(Б) - \mathscr{L}^{l}_{k}(Б)$ и при этом $l_0(Г) = l$.

7.7. а) Показать, что если h — высота дерева подчинения, согласованного с некоторой системой составляющих густоты μ (т. е. такой, что густота соответствующего дерева составляющих равна μ), то $\mu \leqslant h$.

б) Показать, что для Б-грамматики Г тогда и только тогда существует (и может быть построена) сильно (соответственно слабо) эквивалентная ей Д-грамматика G ограниченной высоты, когда активная емкость Г ограничена константой (соответственно $L(Г)$ является ОАЕ-языком). Оценить зависимость между активной емкостью Г и высотой G.

7.8. Построить Б-грамматику, порождающую язык $\{a^n b^n \mid n = 1, 2, \ldots\}^+$ и такую, что густоты деревьев выводов в ней не превосходят двух.

7.9. Показать, что если Б-грамматика Г обладает тем свойством, что для каждой цепочки из $L(Г)$ найдется дерево полного вывода в Г, густота которого не превосходит k, где k — постоянная, зависящая только от Г, то для Г можно построить эквивалентную ей Б-грамматику Г′, в которой густоты всех деревьев выводов ограничены тем же числом k [Диковский 1972 (б)].

7.10. а) Обобщить лемму 7.2 на случай произвольных Б-грамматик.

б) Изменить определение густоты так, чтобы лемма 7.2 (в формулировке, приведенной в тексте) была справедлива для произвольных Б-грамматик.

7.11. Показать, что активная емкость итерационно-линейной грамматики не превосходит двух.

Замечание. Вместе с упражнением 5.28 отсюда следует, что класс итерационно-линейных языков является собственным подклассом класса $\mathscr{L}^{l}_{2}(Б)$.

7.12. Показать, что:

а) класс ОАЕ-языков замкнут относительно усеченной итерации;

б) ни один из классов $\mathscr{L}^{l}_{k}(Б)$ относительно усеченной итерации не замкнут.

7.13. Назовем грамматику контекстно-неукорачивающей, если каждое ее правило имеет вид $x\varphi y \to x'\psi y'$, где $\varphi \in W \cup W(V \cup W)^* W$; $\psi \in W \cup W(V \cup W)^* W \cup \{\Lambda\}$; $x, y, x', y' \in V^*$; $|x| \leqslant |x'|$; $|y| \leqslant |y'|$ (V и W — соответственно основной и вспомогательный словари Г). Показать, что для любой контекстно-неукорачивающей ОАЕ-(ОАЕВ-) грамматики Г такой, что $l_Г(n) \leqslant k$, соответственно $l_0(Г) \leqslant k$, можно построить эквивалентную ей Б-ОАЕ-(ОАЕВ-) грамматику Г′, также удовлетворяющую условию $l_{Г'}(n) \leqslant k$, соответственно $l_0(Г') \leqslant k$ [Диковский 1972 (б)].

7.14. Найти необходимое и достаточное условие равенства нулю степени гнездования приведенной удлиняющей Б-грамматики.

7.15. Показать, что для функций $\Phi^{Л}_{Г}(n)$ и $\Phi^{П}_{Г}(n)$ (стр. 84) имеют место аналоги теоремы 7.6.

7.16. Показать, что в любой Б-грамматике, порождающей нелинейный язык, из начального символа выводима цепочка, содержащая не менее двух вхождений самовставляющихся символов. [*Указание.* Использовать упражнение 5.21, б).]

7.17. Показать, что разброс любой однозначной Б-грамматики, порождающей нелинейный язык, мажорирует подходящую линейную функцию.

ГЛАВА 8

НЕРАЗРЕШИМЫЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

§ 8.1. Предварительные замечания

При изучении формальных грамматик, как и многих других математических объектов, нередко приходится сталкиваться с так называемыми массовыми проблемами. Массовая проблема — это совокупность однотипных задач, связанных с конструктивными объектами некоторого четко определенного вида, причем каждая «индивидуальная» задача состоит в построении по заданному объекту Ξ другого конструктивного объекта H, имеющего, вообще говоря, иной вид. В весьма распространенном частном случае требуется узнать, обладает ли Ξ некоторым свойством; тогда объект H должен быть одной из двух констант 1 и 0 (или, если угодно, «истина» и «ложь»), интерпретируемых как утверждения «Ξ обладает данным свойством» и «Ξ не обладает данным свойством» соответственно. Решение массовой проблемы ищется, как правило, в виде *алгоритма*, позволяющего для каждого объекта Ξ заданного вида построить соответствующий объект H; поэтому массовые проблемы часто называют *алгоритмическими*. Если нужный алгоритм существует, говорят, что данная алгоритмическая проблема *разрешима*, если нет — что она *неразрешима*. (Говорят также «проблема алгоритмически разрешима», соответственно «неразрешима».) В случае, когда все Ξ и H являются цепочками в некотором алфавите (или допускают эффективное кодирование с помощью цепочек, что́ имеет место в действительности для любых конструктивных объектов), требуемый алгоритм

можно, в силу тезиса Черча, точнее, «тезиса Тьюринга — Черча», понимать как ДЭ-машину, которая вычисляет функцию, определенную на всех цепочках Ξ заданного вида и принимающую на каждой из них значение, равное соответствующей цепочке H. Впрочем, при доказательстве разрешимости алгоритмической проблемы обычно ограничиваются указанием принципа работы алгоритма, не приводя построения соответствующей машины Тьюринга (или рекурсивной схемы, нормального алгорифма и т. п.), поскольку такое построение, коль скоро алгоритм достаточно ясно описан, не представляет трудностей *), но при доказательстве неразрешимости необходимо устанавливать именно несуществование нужной машины Тьюринга (или эквивалентного механизма четко определенного вида, например рекурсивной схемы).

В предыдущем изложении мы неоднократно встречались с алгоритмическими проблемами, получавшими положительное решение. Сюда относятся все утверждения о возможности для всякой грамматики или машины некоторого специального вида построить эквивалентную грамматику или машину другого специального вида, о перестройке вывода и т. п. Напомним также об алгоритме, позволяющем по любой неукорачивающей грамматике Γ и любой цепочке x в основном словаре этой грамматики распознать, принадлежит ли x языку $L(\Gamma)$ (§ 3.3), и об алгоритмах для распознавания пустоты и конечности языка, порождаемого Б-грамматикой (§ 4.2). В настоящей же главе мы сосредоточим внимание на тех алгоритмических проблемах (касающихся грамматик), которые решаются отрицательно.

Проблемы, с которыми мы будем иметь дело, можно разделить на два типа. К первому типу принадлежат проблемы распознавания истинности предикатов, определенных на грамматиках. По большей части, но не всегда, речь будет идти об одноместных предикатах, т. е. о *свойствах* грамматик; в случае многоместных предикатов естественно говорить об *отношениях* между

*) Возможны, вообще говоря, и неконструктивные доказательства существования алгоритма, не дающие способа его построить, например, путем приведения к противоречию утверждения о неразрешимости соответствующей проблемы.

грамматиками. Пусть $\mathscr{T}$ — некоторый класс грамматик и предикат $P(\Gamma_1, \ldots, \Gamma_n)$ определен для всех $\Gamma_1, \ldots, \Gamma_n \in \mathscr{T}$. Мы будем называть предикат P распознаваемым в классе $\mathscr{T}$, если существует алгоритм, позволяющий для любых $\Gamma_1, \ldots, \Gamma_n \in \mathscr{T}$ распознать, истинно ли утверждение $P(\Gamma_1, \ldots, \Gamma_n)$, и нераспознаваемым в классе $\mathscr{T}$, если такого алгоритма нет. В частности, при $n = 1$ будем говорить о распознаваемом (нераспознаваемом) свойстве. Наибольший интерес представляют инвариантные свойства и отношения, т. е. такие свойства и отношения, которые зависят только от порождаемых грамматиками языков; иначе говоря, инвариантность предиката $P(\Gamma_1, \ldots, \Gamma_n)$ означает, что из $L(\Gamma_1) = L(\Gamma_1'), \ldots, L(\Gamma_n) = L(\Gamma_n')$ следует $P(\Gamma_1, \ldots, \Gamma_n) \equiv P(\Gamma_1', \ldots, \Gamma_n')$. Всякому инвариантному свойству грамматик, соответственно отношению между грамматиками, естественным образом сопоставляется некоторое свойство языков (отношение между языками). Если α — некоторое инвариантное свойство грамматик (отношение между грамматиками) и α' — соответствующее свойство языков (отношение между языками), мы обычно будем вместо «α распознаваемо (нераспознаваемо) в классе $\mathscr{T}$» говорить, что α' распознаваемо (нераспознаваемо) в том же классе. Так, результаты § 4.2 могут быть теперь сформулированы следующим образом: пустота и конечность языка распознаваемы в классе Б-грамматик.

Все свойства и отношения, нераспознаваемость которых будет доказываться в основном тексте этой главы, инвариантны. Примеры проблем, относящихся к распознаванию неинвариантных свойств и отношений, см. в упражнениях 8.1, 8.2, 8.15.

Если некоторое свойство или отношение распознаваемо в классе $\mathscr{T}$, то оно, очевидно, распознаваемо и в любом классе, содержащемся в $\mathscr{T}$. Обратное, вообще говоря, неверно *); ниже мы увидим, например, что в

*) Однако если два класса грамматик «эффективно эквивалентны», т. е. для любой грамматики одного из этих классов можно построить эквивалентную грамматику другого класса, то из алгоритма, распознающего какое-либо инвариантное свойство в одном из этих классов, немедленно получается соответствующий алгоритм для другого (ср. общее замечание в конце настоящего параграфа).

классе НС-грамматик пустота и конечность языка уже нераспознаваемы.

В дальнейшем нам будет полезен также следующий термин: мы будем называть свойство языков нетривиальным в некотором классе языков, если среди языков этого класса имеются как обладающие, так и не обладающие данным свойством.

Проблемы второго типа формулируются следующим образом. Пусть V — некоторый словарь, $\mathscr{T}$ — класс грамматик и $P(L, x_1, \ldots, x_n)$ — предикат, определенный для всех языков L, порождаемых грамматиками класса $\mathscr{T}$ с основным словарем V, и всех цепочек $x_1, \ldots, x_n \in V^*$. Ставится вопрос: существует ли алгоритм, позволяющий по любой грамматике $\Gamma \in \mathscr{T}$ с основным словарем V и любым n цепочкам $x_1, \ldots, x_n \in V^*$ распознать, истинно ли утверждение $P(L(\Gamma), x_1, \ldots, x_n)$? («Общая проблема распознавания P в классе $\mathscr{T}$».) Фиксировав грамматику $\Gamma^0 \in \mathscr{T}$ с основным словарем V, можно поставить другой вопрос («проблему распознавания P для Γ^0» или «частную проблему распознавания P»): существует ли алгоритм, позволяющий по любым цепочкам $x_1, \ldots, x_n \in V$ распознать, истинно ли $P(L(\Gamma^0), x_1, \ldots, x_n)$? Из неразрешимости частной проблемы распознавания P хотя бы для одной грамматики класса $\mathscr{T}$ вытекает, очевидно, неразрешимость общей проблемы распознавания P в $\mathscr{T}$.

К этому типу относятся проблема распознавания принадлежности цепочки языку, порождаемому данной грамматикой, проблема распознавания замещаемости одной цепочки на другую в языке, порождаемом данной грамматикой, и т. п.

Заметим, что при фиксировании цепочек $x_1, \ldots, x_n$ общая проблема распознавания $P(L, x_1, \ldots, x_n)$ переходит в проблему первого типа — о распознавании некоторого свойства языков. Например, фиксируя x в предикате $x \in L$, получаем проблему распознавания свойства языка содержать данную цепочку.

В заключение параграфа сделаем одно простое замечание, которое будет чрезвычайно полезно нам в дальнейшем. Пусть $\mathfrak{A}$ и $\mathfrak{B}$ — классы конструктивных объектов, α — свойство объектов класса $\mathfrak{A}$ и β — свойство объектов класса $\mathfrak{B}$. Пусть, далее, имеется алгоритм

$\mathfrak{P}$, позволяющий для всякого $a \in \mathfrak{A}$ построить такое $b(a) \in \mathfrak{B}$, что $b(a)$ обладает свойством β тогда и только тогда, когда a обладает свойством α. Если при этом существует также алгоритм $\mathfrak{R}$, позволяющий для всякого объекта $b \in \mathfrak{B}$ распознать, обладает ли он свойством β, то есть и алгоритм, позволяющий для всякого объекта $a \in \mathfrak{A}$ распознать, обладает ли он свойством α (достаточно по данному a построить $b(a)$ и затем применить к $b(a)$ алгоритм $\mathfrak{R}$). Поэтому при наличии алгоритма $\mathfrak{P}$ неразрешимость проблемы распознавания α влечет неразрешимость проблемы распознавания β.

Если описанный алгоритм $\mathfrak{P}$ существует, мы будем говорить, что проблема распознавания α сводится к проблеме распознавания β.

Таким образом, для доказательства неразрешимости алгоритмической проблемы достаточно свести к ней какую-либо другую алгоритмическую проблему, неразрешимость которой уже доказана. Точнее, мы установили это для проблем распознавания свойств; но тот же прием мы можем использовать и для проблем распознавания отношений, поскольку отношение, например, между двумя конструктивными объектами можно рассматривать как свойство упорядоченной пары объектов, а пара конструктивных объектов также есть конструктивный объект.

§ 8.2. Инвариантные свойства произвольных грамматик

В дальнейшем нам часто придется, рассматривая некоторый класс грамматик $\mathscr{T}$, выделять те грамматики этого класса, которые имеют фиксированный основной словарь V. Совокупность всех таких грамматик мы будем обозначать через $\mathscr{T}_V$, а класс языков, ими порождаемых, — в соответствии с соглашением на стр. 30 — через $\mathscr{L}(\mathscr{T}_V)$. Напомним также, что класс произвольных грамматик мы договорились обозначать через Γ.

Целью настоящего параграфа является доказательство следующей теоремы.

Теорема 8.1. *Каков бы ни был словарь V, никакое свойство языков, нетривиальное в классе $\mathscr{L}(\Gamma_V)$ — иначе говоря, в классе рекурсивно перечислимых языков, содержащихся в V^*, — не может быть распознаваемым в классе Γ_V и тем более в классе всех грамматик.*

Вместо этого утверждения нам удобнее будет доказывать другое, равносильное ему ввиду теоремы 1.4, а именно:

Теорема 8.1′. *Каково бы ни было свойство языков α, нетривиальное в классе рекурсивно перечислимых языков, содержащихся в V^*, где V — фиксированный непустой алфавит, не существует алгоритма, позволяющего для любой Э-машины M с входным алфавитом V распознать, обладает ли язык $L(M)$ свойством α.*

Для доказательства теоремы 8.1′ установим, в свою очередь, справедливость следующей теоремы, которая понадобится нам и дальше.

Теорема 8.2 (теорема Райса). *Пусть W и V — фиксированные непустые алфавиты и $\mathfrak{F}$ — класс всевозможных частично рекурсивных функций, области определения и множества значений которых содержатся в W^* и V^* соответственно. Пусть, кроме того, α — произвольное свойство функций, нетривиальное в классе $\mathfrak{F}$, т. е. такое, что в $\mathfrak{F}$ есть как функции, обладающие этим свойством, так и функции, им не обладающие. Тогда не существует алгоритма, позволяющего для любой ДЭ-машины с входным алфавитом W распознать, обладает ли функция $\Pr_V f(x)$, где $f(x)$ — вычисляемая данной машиной функция**), *свойством α.*

Нетрудно видеть, что теорема 8.2 влечет теорему 8.1′. Действительно, по теоремам 1.2 и 1.3 для всякой Э-машины M, допускающей $L(M) \subseteq V^*$, можно построить ДЭ-машину M', вычисляющую функцию с множеством значений $L(M)$. Область определения этой функции не обязана, вообще говоря, содержаться в W^*, но она, во всяком случае, представляет собой язык в некотором (конечном) алфавите Z, и легко построить ДЭ-машину M'' (см., например, упражнение 1.19, а), б)), осуществляющую взаимно однозначное отображение W^* на Z^*. Комбинируя машины M'' и M', получаем ДЭ-машину M_1, вычисляющую функцию с множеством значений $L(M)$ и областью определения, содержащейся в W^*. С другой стороны, для любой Э-машины,

*) Очевидно, всякая ДЭ-машина вычисляет некоторую функцию (которая может, впрочем, оказаться нигде не определенной).

вычисляющей функцию $f(x)$, тривиальным образом строится Э-машина, вычисляющая функцию $\Pi p_V f(x)$, а это ввиду теоремы 1.3 позволяет построить Э-машину (с входным алфавитом V), допускающую множество значений $\Pi p_V f(x)$. Пусть теперь β — произвольное свойство языка и α — свойство функции иметь множество значений, проекция которого на V обладает свойством β. Из предыдущего ясно, что если β нетривиально в классе рекурсивно перечислимых языков в алфавите V, то α нетривиально в классе $\mathfrak{F}$, и из существования алгоритма, распознающего по любой Э-машине M с входным алфавитом V, обладает ли $L(M)$ свойством β, немедленно вытекало бы существование алгоритма, позволяющего по любой ДЭ-машине с входным алфавитом W распознать, обладает ли соответствующая функция $\Pi p_V f(x)$ свойством α.

Обратимся к доказательству теоремы 8.2. Будем рассматривать всевозможные ДЭ-машины с входным алфавитом W. Без ущерба для общности можно считать, что рабочие алфавиты и множества состояний всех рассматриваемых машин содержатся в непересекающихся счетных множествах U и R соответственно, и сверх того $R \cap W = \varnothing$ (поскольку это ограничение не сказывается на объеме класса вычисляемых машинами функций). Класс всех таких машин мы обозначим через $\mathfrak{M}$. Сопоставим каждой машине $M \in \mathfrak{M}$ цепочку $\varkappa(M) \in W^+$, которая будет называться кодом машины M, следующим образом.

а) Допустим сначала, что мощность W не меньше двух. Выберем в W произвольным образом два символа a и b. Положим $Z = V \cup U \cup R$ и $X = Z \cup \{П, Л, \rightarrow\}$, где П, Л, $\rightarrow$ — некоторые символы, не входящие в Z. Занумеруем каким-либо образом элементы X (например, элементы $V \cup \{П, Л, \rightarrow\}$ первыми $p+3$ целыми положительными числами, где p — мощность V, а элементы R и $U - V$ соответственно четными и нечетными числами, бо́льшими $p+3$). Для произвольного символа $\alpha \in X$ обозначим через $\varkappa(\alpha)$ цепочку ba^kb, где k — номер α. Для произвольной непустой цепочки $\omega = \alpha_1 \ldots \alpha_s$, где $\alpha_1, \ldots, \alpha_s \in X$, обозначим через $\varkappa(\omega)$ цепочку $\varkappa(\alpha_1) \ldots \varkappa(\alpha_s)$. Наконец, для произвольной машины $M \in \mathfrak{M}$ обозначим через ω_M цепочку $q_1 q^1 \ldots q^t I_1 \ldots I_u$,

где q_1 — начальное состояние M, $q^1, \ldots, q^t$ — все заключительные состояния M и $I_1, \ldots, I_u$ — все инструкции M, причем q^i расположены в порядке возрастания их номеров в выбранной нами нумерации X, а I_j — в лексикографическом порядке (т. е. сначала по возрастанию номеров первых символов, затем вторых и т. д.), и положим $\varkappa(M) = \varkappa(\omega_M)$.

б) Если W состоит из одного элемента a_0, то сначала построим для каждой машины $M \in \mathfrak{M}$ цепочку $\varkappa'(M)$ точно так же, как в предыдущем случае строилась $\varkappa(M)$, но используя вместо a и b произвольные различные символы a' и b', а затем положим $\varkappa(M) = a_0^{\nu^{-1}(\varkappa'(M))}$, где ν — отображение, определенное в упражнении 1.3 (при $k = 2$, $a_1 = a'$, $a_2 = b'$).

Будем называть машину $M \in \mathfrak{M}$ самоприменимой, если область определения вычисляемой этой машиной функции содержит код M, и несамоприменимой в противном случае. Имеет место

Лемма 8.1. *Не существует алгоритма, позволяющего для любой машины из $\mathfrak{M}$ распознать, является ли она самоприменимой.*

Доказательство леммы 8.1. Поскольку машина класса $\mathfrak{M}$ восстанавливается по своему коду однозначно и эффективно, существование указанного алгоритма означало бы существование ДЭ-машины M_1, вычисляющей функцию, определенную на всех кодах машин из $\mathfrak{M}$ и принимающей на кодах всех самоприменимых машин одно и то же значение ω_1, а на кодах всех несамоприменимых машин — одно и то же значение ω_2, отличное от ω_1. Допустим, что такая машина M_1 существует; без нарушения общности можно считать, очевидно, что она принадлежит $\mathfrak{M}$. По M_1 легко построить (это предоставляется читателю) другую машину M_2, тоже принадлежащую $\mathfrak{M}$ и работающую на кодах несамоприменимых машин так же, как M_1, а на кодах самоприменимых машин работающую вечно, так что вычисляемая этой машиной функция f определена на кодах всех несамоприменимых машин и не определена ни на одном коде самоприменимой машины. Но машина M_2 не может быть самоприменимой, поскольку это означало бы, что функция f определена на коде M_2, который был бы при этом кодом самоприменимой машины; в то же время и

несамоприменимой M_2 быть не может — ведь это равносильно тому, что на коде M_2 функция f не определена, и в то же время код M_2 был бы кодом несамоприменимой машины, а на таких кодах f как раз определена. Имеем противоречие.

Доказательство теоремы 8.2. Пусть $\mathfrak{F}_0$ — подкласс $\mathfrak{F}$, состоящий из всех функций, обладающих свойством α, и $\mathfrak{F}_1 = \mathfrak{F} - \mathfrak{F}_0$. По условию оба класса $\mathfrak{F}_0$ и $\mathfrak{F}_1$ не пусты. Класс $\mathfrak{F}$ содержит, в частности, нигде не определенную функцию (т. е. функцию с пустой областью определения) f_0; допустим для определенности, что $f_0 \in \mathfrak{F}_0$. Фиксируем далее некоторую функцию $f_1 \in \mathfrak{F}_1$ и вычисляющую ее машину $M_1 \in \mathfrak{M}$. Рассмотрим произвольную машину $M \in \mathfrak{M}$ и построим по ней другую машину $M' \in \mathfrak{M}$, работающую следующим образом. В начале вычисления машина M', «не обращая внимания» на содержимое входной ленты, записывает на рабочей ленте цепочку $\varkappa(M)$ и затем работает как п. о. м. машины M, «пытаясь», таким образом, вычислить значение определяемой этой машиной функции при значении аргумента, являющемся ее собственным кодом. Если соответствующее значение будет вычислено, то вслед за этим рабочая лента уничтожается, а потом машина начинает работать, как M_1, т. е. вычислять значение f_1 от входной цепочки. В противном случае M' работает вечно. Итак, машина M' вычисляет функцию $f_1 \in \mathfrak{F}_1$, если M самоприменима, и нигде не определенную функцию $f_0 \in \mathfrak{F}_0$, если M несамоприменима. При этом f_0 и f_1 совпадают со своими проекциями на V. Следовательно, проблема распознавания самоприменимости для машин класса $\mathfrak{M}$ сводится к проблеме распознавания свойства α для проекций на V функций, вычисляемых машинами того же класса, так что, согласно замечанию в конце § 8.1, неразрешимость первой проблемы влечет неразрешимость второй.

§ 8.3. Инвариантные свойства НС-грамматик

В предыдущем параграфе мы показали, что в классе произвольных грамматик все нетривиальные свойства языков нераспознаваемы. Уже в классе НС-грамматик дело обстоит иначе. Действительно, для произвольной

цепочки x свойство языка содержать x, как уже отмечалось, распознаваемо в этом классе*). Поэтому в нем распознаваемы и такие свойства, как «содержать хотя бы одну из двух данных цепочек», «содержать одну данную цепочку и не содержать другую» и т. п. Вообще, если $\Phi(L, x_1, \ldots, x_k)$ — произвольное булево выражение от предикатов «$x_1 \in L$», ..., «$x_k \in L$» (т. е. выражение, составленное из этих предикатов с помощью пропозициональных связок $\neg$, &, $\vee$, $\supset$) и $\Phi_0(x_1, \ldots, x_k)$ — свойство языка L удовлетворять условию $\Phi(L, x_1, \ldots, x_k)$, то $\Phi_0(x_1, \ldots, x_k)$ распознаваемо в классе НС. Такие свойства мы назовем булевыми.

Однако для некоторого весьма общего типа инвариантных свойств все же удается доказать их нераспознаваемость в классе НС; это и будет основной целью настоящего параграфа.

Предварительно докажем несколько лемм.

Лемма 8.2. *Для всякой* ДЭ-*машины* M, *входной алфавит которой содержит символ* a, *можно построить такие* НС-*грамматики* Γ_1^M *и* Γ_2^M *с одним и тем же одноэлементным основным словарем* $\{b\}$, *что*:

а) *если вычисляемая машиной* M *функция определена для значения аргумента, равного* a, *то* $L(\Gamma_1^M) = \{b^n \mid n \geqslant n_0\}$, $L(\Gamma_2^M) = \{b^n \mid 1 \leqslant n \leqslant n_0\}$, *где* n_0 — *целое положительное число, зависящее от* M;

б) *в противном случае* $L(\Gamma_1^M) = \varnothing$, $L(\Gamma_2^M) = b^+$.

Доказательство. Построим сначала по машине M ее детерминированную п. о. м. M_1 (лемма 1.5). Затем построим по M_1 другую ДЭ-машину M_2 так, чтобы она также представляла собой п. о. м. для M и в то же время обладала тем свойством, что в случае, когда вычисляемая машиной M функция не определена, M_2 работала бы вечно и при этом ее рабочая лента при достаточно долгой работе оказывалась бы сколь угодно длинной. [Это можно сделать, например, перестроив M_1 так, чтобы она: а) перед каждым своим шагом добавляла к рабочей ленте справа ячейку, которая уничтожалась

*) Как и в более широком классе неукорачивающих грамматик. В отношении распознаваемости инвариантных свойств эти два класса вообще ведут себя одинаково (см. сноску на стр. 254).

бы лишь после перехода машины в «заключительное M_1-состояние»; б) в случае «безрезультатной остановки» начинала добавлять к рабочей ленте ячейку за ячейкой без конца.] Кроме того, мы будем считать, что левые части инструкций M_2 не содержат заключительных состояний; это, очевидно, не уменьшает общности. В силу теоремы 3.1 достаточно построить неукорачивающие грамматики $\tilde{\Gamma}_1^M$ и $\tilde{\Gamma}_2^M$ со свойствами а) и б); мы определим их следующим образом.

1) Основной и вспомогательный словари V и W и начальный символ I — общие для обеих грамматик, а именно: $V=\{b\}$, $W=Z\cup Q\cup\{\#, I, C_0\}$, где Z, Q — соответственно рабочий алфавит и множество состояний M_2 и $\#$, I, $C_0\notin Z\cup Q\cup\{b\}$.

2) Схемы обеих грамматик содержат правило $I\to\# q_1 ad$, где q_1 — начальное состояние M_2 и d — «разделительный символ» (см. определение п. о. м., стр. 45), всевозможные правила вида $C_0\alpha\to\alpha C_0$, где $\alpha\in W$, и правила, сопоставляемые инструкциям M_2 следующим образом: каждой инструкции $Aq_\alpha\to q_\beta$ типа (ii) сопоставляется правило $Aq_\alpha\to q_\beta C_0$, каждой инструкции $q_\alpha\to Aq_\beta$ типа (iii) — правило $q_\alpha\to Aq_\beta$, каждой инструкции $q_\alpha\to q_\beta\Pi$ типа (iv) — всевозможные правила вида $Bq_\alpha\to q_\beta B$, где $B\in Z$, и каждой инструкции $q_\alpha\to q_\beta\Pi$ типа (v) — всевозможные правила вида $q_\alpha B\to Bq_\beta$, где $B\in Z$. 3) Сверх того, схема $\tilde{\Gamma}_1$ содержит всевозможные правила вида $q_f\to b$, где q_f — заключительное состояние M_1, вида $bB\to bb$, где $B\in W$, вида $Bb\to bb$, где $B\in W\cup\{\#\}$, и правило $b\to bb$, а схема $\tilde{\Gamma}_2^M$ — правила $I\to b$, $I\to bb$, $I\to bbb$ и всевозможные правила вида $B\to b$, где $B\in Z\cup Q\cup\{\#\}$.

Ясно, что если вычисляемая машиной M функция определена на a, то ситуаций машины M_2, достижимых из ситуации $S_0=(q_1, \Lambda, \# \ddagger, \Lambda, \# ad)$, имеется лишь конечное число, причем длины отвечающих этим ситуациям рабочих лент пробегают все натуральные числа от 2 до некоторого n_1 включительно; поэтому в $\tilde{\Gamma}_1^M$ будут в этом случае выводимы из I всевозможные цепочки вида b^k, где $k\geqslant n_1+2$, и только такие степени b, а в $\tilde{\Gamma}_2^M$ — всевозможные цепочки вида b^l, где

$1\leqslant l\leqslant n_1+2$, и только такие степени b. В противном случае достижимых из S_0 ситуаций будет бесконечно много, длины отвечающих им рабочих лент будут пробегать все натуральные числа, не меньшие 2, но ни одна из этих ситуаций не будет содержать заключительных состояний; поэтому в $\tilde{\Gamma}_1^M$ не будет выводима из I ни одна степень b, а в $\tilde{\Gamma}_2^M$, напротив, все степени b будут выводимы из I.

Лемма 8.3. *Для любых двух* НС-*грамматик* Γ_1 *и* Γ_2 *можно построить* НС-*грамматику, порождающую язык*

$$\{x\mid x\in L(\Gamma_1)\,\&\,\exists y\,(y\in L(\Gamma_2)\,\&\mid y\mid=\mid x\mid)\}.$$

Доказательство. В силу теоремы 3.3 достаточно построить почти неукорачивающую грамматику, порождающую нужный язык. Пусть

$$\Gamma_1=\langle V_1, W_1, I_1, R_1\rangle,\quad \Gamma_2=\langle V_2, W_2, I_2, R_2\rangle.$$

Без ограничения общности мы можем считать, что $W_1\cap W_2=\varnothing$. Сопоставим каждому символу $a\in V_1$ новый символ $\bar{a}$ и обозначим через $\bar{\Gamma}_1$ грамматику, полученную из Γ_1 заменой каждого символа $a\in V_1$ и каждого его вхождения в каждое правило на $\bar{a}$. Положим $\Gamma=\langle V_1, Z\cup\{I\}, I, R\rangle$, где а) $Z=V_2\cup W_1\cup W_2\cup\cup\{\bar{a}\mid a\in V_1\}$, $I\notin Z\cup V_1$; б) R получается из объединения схем $\bar{\Gamma}_1$ и Γ_2 добавлением правил $I\to I_1I_2$, $\bar{a}b\to b\bar{a}$, $b\bar{a}\to a$, где a и b пробегают V_1 и V_2 соответственно. Ясно, что при $V_1\cap V_2=\varnothing$ Γ есть нужная грамматика. В противном случае следует предварительно преобразовать Γ_2 так же, как было сделано с Γ_1.

Положим теперь для произвольного языка L и произвольного $n=1, 2, \ldots$

$$L^{(n)}=\{x\mid x\in L\,\&\,1\leqslant\mid x\mid\leqslant n\},\quad \overline{L}^{(n)}=\{x\mid x\in L\,\&\mid x\mid\geqslant n\}$$

и докажем следующую лемму.

Лемма 8.4. *Пусть* L_0, L_1, L_2 — НС-*языки и* $\mathscr{L}$ — *класс языков, содержащий* $L_0\cup L_1$ *и не содержащий ни одного из языков* $L_0\cup L_1^{(n)}\cup\overline{L}_2^{(n)}$. *Тогда свойство принадлежать классу* $\mathscr{L}$ *нераспознаваемо в классе* НС-*грамматик.*

Доказательство. Пусть Δ_0, Δ_1, Δ_2 — НС-грамматики, порождающие L_0, L_1, L_2 соответственно.

Фиксировав символы a и b, рассмотрим произвольну ДЭ-машину M с входным алфавитом $\{a\}$ и построим дл нее НС-грамматики Γ_1^M и Γ_1^M по лемме 8.2. Затем, применив к грамматикам Δ_1 и Γ_1^M лемму 8.3, построим НС-грамматику Γ', порождающую множество тех цепо чек из L_1, для которых в $L(\Gamma_1^M)$ имеются цепочки равной длины, и аналогичную НС-грамматику Γ'' построи для Δ_2 и Γ_2^M. Если вычисляемая машиной M функция определена для значения аргумента, равного a, т языки $L(\Gamma')$ и $L(\Gamma'')$ будут, очевидно, совпадать соот ветственно с $L_1^{(n_0)}$ и $\bar{L}_2^{(n_0)}$, где n_0 — число из леммы 8. в противном случае будет $L(\Gamma')=L_1$, $L(\Gamma'')=\varnothing$. На конец, построим НС-грамматику Γ, порождающую язы $L(\Gamma_0)\cup L(\Gamma')\cup L(\Gamma'')$ (теорема 3.4). Язык $L(\Gamma)$ будет со впадать с $L_0\cup L_1^{(n_0)}\cup\bar{L}_2^{(n_0)}$, если f определена для a, с $L_0\cup L_1$ в противном случае. Но $L_0\cup L_1\in\mathscr{L}$ $L_0\cup L_1^{(n_0)}\cup\bar{L}_2^{(n_0)}\notin\mathscr{L}$; таким образом, неразрешимая п теореме 8.2 проблема распознавания свойства ДЭ-ма шины M с входным алфавитом $\{a\}$ вычислять функ цию *), определенную для значения аргумента a, сво дится к проблеме распознавания свойства НС-грам матики порождать язык, принадлежащий классу $\mathscr{L}$.

Замечание. Из доказательства леммы 8.4 ясно что она остается справедливой и в несколько усиленно формулировке; именно, для произвольного словаря свойство принадлежать классу $\mathscr{L}$ нераспознаваем в классе HC_V. То же верно и для последующих рассмот рений этого параграфа (исключая одну специальну проблему, особо оговариваемую ниже, стр. 267).

Теперь мы можем перейти к основной теореме этог параграфа.

Скажем, что языки L_1 и L_2 почти совпадают если они отличаются друг от друга лишь конечным чис лом элементов, т. е. их симметрическая разност $(L_1-L_2)\cup(L_2-L_1)$ конечна. Отношение почти совпаде ния является, очевидно, эквивалентностью; классы инду цируемого этой эквивалентностью разбиения мы будем на

*) Собственно говоря, речь должна идти о проекции этой фун ции на некоторый алфавит (который можно выбрать произвольно) но проектирование не изменяет области определения функции.

зывать пучками. Ясно, что всякий пучок, содержащий хотя бы один НС-язык, целиком состоит из НС-языков.

Теорема 8.3. *Если $\mathscr{L}$ — класс языков, содержащий хотя бы один* НС-*язык и имеющий хотя бы с одним пучком* НС-*языков конечное (в частном случае пустое) пересечение, то свойство принадлежать классу $\mathscr{L}$ нераспознаваемо в классе* НС-*грамматик.*

Доказательство. Пусть сначала пересечение $\mathscr{L}$ с пучком конечных языков бесконечно. Тогда существует пучок П, состоящий из бесконечных языков и пересекающийся с $\mathscr{L}$ по конечному множеству. Обозначим через L_0 произвольный конечный язык, принадлежащий $\mathscr{L}$, через L_1 — пустой язык и через L — произвольный язык из П. При любом $m=1,2,\ldots$ язык $L_0\cup\bar{L}^{(m)}$ также принадлежит П, и, следовательно, найдется такое p, что ни при каком $n\geqslant p$ язык $L_0\cup\bar{L}^{(n)}$ не принадлежит $\mathscr{L}$. Положив $\bar{L}^{(p)}=L_2$, будем иметь $\bar{L}_2^{(n)}=\bar{L}^{(n)}$, если $n\geqslant p$, и $\bar{L}_2^{(n)}=\bar{L}^{(p)}$, если $n<p$; поэтому при любом $n=1,2,\ldots$ получаем $L_0\cup L_1^{(n)}\cup\bar{L}_2^{(n)}=L_0\cup\bar{L}_2^{(\max(n,\,p))}\notin\mathscr{L}$. В то же время $L_0\cup L_1=L_0\in\mathscr{L}$. Остается применить лемму 8.4.

Пусть теперь $\mathscr{L}$ пересекается с пучком конечных языков по конечному множеству. Если при этом $\mathscr{L}$ не содержит бесконечных НС-языков, воспользуемся леммой 8.4, взяв в качестве L_0 произвольный конечный язык, принадлежащий $\mathscr{L}$, в качестве L_1 — пустой язык и в качестве L_2 — произвольный бесконечный НС-язык. В противном случае, — если существует бесконечный НС-язык $\tilde{L}$, принадлежащий $\mathscr{L}$, — применяем ту же лемму при $L_0=\tilde{L}^{(p)}$, $L_1=\tilde{L}$, $L_2=\varnothing$, где p — такое натуральное число, что ни один из языков $\tilde{L}^{(n)}$, где $n\geqslant p$, в $\mathscr{L}$ не входит.

Доказанная теорема позволяет легко убедиться в нераспознаваемости ряда более конкретных типов свойств. Имеют место, в частности, такие факты:

Следствие 1. *В следующих случаях свойство, нетривиальное в классе* НС-*языков, нераспознаваемо в классе* НС-*грамматик*:

а) *если им обладает лишь конечное число* НС-*языков*;

б) *если им обладают все конечные языки, соответственно все языки с конечными дополнениями, кроме, быть может, конечного их числа*;

в) *если им обладают только* Б-*языки и, быть может, конечное число* НС-*языков, не являющихся* Б-*языками.*

Пункт а) вытекает из того, что число пучков НС-языков бесконечно, пункт б) — из того, что конечные языки, равно как и языки с конечными дополнениями, образуют пучок (и из нераспознаваемости отрицания нераспознаваемого свойства), пункт в) — из того, что всякий пучок, содержащий Б-язык, целиком состоит из Б-языков (в силу теорем 4.8 и 5.5).

В частности, нераспознаваемы в классе НС-грамматик следующие свойства языков: быть пустым, иметь пустое дополнение, быть конечным, иметь конечное дополнение, быть бесконтекстным, иметь бесконтекстное дополнение, быть автоматным, линейным, металинейным и т. п.

Следствие 2. *Какова бы ни была* НС-*грамматика* Γ_0, *не существует алгоритма, позволяющего для любой* НС-*грамматики распознать, эквивалентна ли она грамматике* Γ_0. *Тем более не существует алгоритма для решения общей проблемы эквивалентности* НС-*грамматик.*

Действительно, свойство языка совпадать с $L(\Gamma_0)$ нераспознаваемо по пункту а) следствия 1.

Следствие 3. *Если* L_0 — *произвольный бесконечный язык, то не существует алгоритмов, позволяющих по произвольной* НС-*грамматике* Γ *распознать*:

а) *содержит ли* $L(\Gamma)$ *язык* L_0;

б) *является ли пересечение* $L_0 \cap L(\Gamma)$ *пустым.*

Это вытекает из пункта б) следствия 1, так как свойством не содержать L_0 обладают все конечные языки, а свойством иметь с L_0 непустое пересечение — все языки с конечными дополнениями.

Аналогично получаем

Следствие 4. *Если* L_0 — *произвольный язык с бесконечным дополнением, то не существует алгоритмов, позволяющих по произвольной* НС-*грамматике* Γ *распознать*:

а) *содержится ли* $L(\Gamma)$ *в* L_0;

б) *имеет ли объединение* $L_0 \cup L(\Gamma)$ *пустое дополнение.*

Из следствия 3 вытекает, например, нераспознаваемость свойства содержать цепочку, содержащую вхождение данной цепочки (соответственно начинающуюся или оканчивающуюся вхождением данной цепочки) (ср. упражнение 4.7), из следствия 4 (в случае словаря не менее чем из двух символов) — нераспознаваемость свойства состоять только из цепочек, содержащих вхождение данной цепочки (соответственно начинающихся или оканчивающихся вхождением данной цепочки).

Другие примеры применения теоремы 8.3 см. в упражнениях 8.5 и 8.6.

Замечание. Если язык L_0 конечен, то указанные в следствии 3 алгоритмы существуют, так как соответствующие свойства являются в этом случае булевыми. Аналогично для следствия 4.

Итак, мы убедились, что «естественно возникающие» не булевы свойства НС-языков оказываются, как правило, нераспознаваемыми. Возникает вопрос: быть может, распознаваемые свойства исчерпываются булевыми? Следующий пример показывает, что это не так.

Пример*). Фиксируем (произвольный) словарь V и сопоставим цепочкам в этом словаре натуральные числа способом, описанным в упражнении 1.3; цепочку, которой сопоставлено число n, будем обозначать ωn. Пусть, далее, R — произвольный рекурсивный язык в словаре V, не являющийся НС-языком**). Определим класс языков $\mathscr{L}$ следующим образом:

$$L \in \mathscr{L} \equiv \exists n\,[\omega n \in (L - R)\,\&\,(\forall i < n)(\omega i \in L \equiv \omega i \in R)].$$

Свойство языка принадлежать $\mathscr{L}$ распознаваемо в классе НС (и $\mathrm{НС}_V$). В самом деле, для произвольной НС-грамматики Γ в силу выбора R имеем $L(\Gamma) \neq R$; поэтому, проверяя последовательно для каждой цепочки $\omega 1, \omega 2, \ldots$, принадлежит ли она $L(\Gamma)$ и принадлежит ли она R, мы дойдем в конце концов до такой цепочки ωn, которая принадлежит либо $L(\Gamma) - R$, либо $R - L(\Gamma)$. Если первая из таких цепочек принадлежит $L(\Gamma) - R$, имеем $L(\Gamma) \in \mathscr{L}$, если же она принадлежит $R - L(\Gamma)$, то $L(\Gamma) \notin \mathscr{L}$.

Вместе с тем свойство принадлежать $\mathscr{L}$ не булево. Действительно, если $\Phi_0 = \Phi_0(x_1, \ldots, x_k)$ — булево

*) Указан М. И. Белецким.

**) Пример такого языка при $\mu(V) \geqslant 2$ доставляется хотя бы теоремой 2.4, если положить в ней $F = S$, $f(n) = n$. При $\mu(V) = 1$ нужно еще произвести перекодирование, как на стр. 259.

свойство и p—наибольший из номеров цепочек $x_1, \ldots, x_h$, то всякие два языка L и L', для которых имеет место равенство $L \cap \{\omega 1, \ldots, \omega p\} = L' \cap \{\omega 1, \ldots, \omega p\}$, либо оба обладают свойством Φ_0, либо оба не обладают. Но если q — наименьшее число, большее p и такое, что $\omega q \notin R$ *), то язык $L = \{\omega 1, \ldots, \omega(q-1)\} \cap R$ не принадлежит $\mathscr{L}$, в то время как $L' = L \cup \{\omega q\} \in \mathscr{L}$, и при этом $L \cap \{\omega 1, \ldots, \omega p\} = L' \cap \{\omega 1, \ldots, \omega p\}$.

§ 8.4. Некоторые проблемы, связанные с Б-грамматиками

При переходе от НС-грамматик к Б-грамматикам класс нераспознаваемых свойств языков сужается еще больше. Так, в классе Б-грамматик распознаваемы пустота (теорема 4.1), конечность (теорема 4.2), свойство содержать цепочку, содержащую вхождение x, или начинающуюся вхождением x, или оканчивающуюся таковым (упражнение 4.7); в классе НС-грамматик все эти свойства, как мы видели, нераспознаваемы. Однако и для Б-грамматик можно доказать нераспознаваемость довольно обширного круга свойств. Этим, а также доказательством неразрешимости в классе Б-грамматик некоторых проблем иного типа мы сейчас и займемся. Впрочем, нам удобнее будет говорить не о Б-, а об ОБ-грамматиках (хотя все результаты останутся справедливыми и для Б-грамматик; соответствующие видоизменения конструкций очевидны, и упоминать о них мы не будем).

Нам понадобится одно вспомогательное понятие, относящееся к машинам Тьюринга, и лемма, связывающая произвольные ДЭ-машины с ОБ-грамматиками.

Пусть M — Э-машина с входным алфавитом V, рабочим алфавитом W и множеством состояний Q. Положим $Z = V \cup W \cup Q \cup \{\#, \ddagger, \nabla\}$, где ∇ — символ, не входящий в $V \cup W \cup Q \cup \{\#, \ddagger\}$. Ситуацию $(q_\alpha, x', x'', X', X'')$ машины M назовем правильной, если $x'x'' = \#x\ddagger$ и $X'X'' = \#X$ для подходящих $x \in V^*$, $X \in W^*$. Записью правильной ситуации $(q_\alpha, x', x'', X', X'')$ будем называть цепочку $q_\alpha x' \nabla x'' X' \nabla X''$. Для произвольного вычисления $C = (S_0, S_1, \ldots, S_n)$ машины M будем называть его протоколом цепочку $\pi(C) = \mathfrak{z}(S_0)\mathfrak{z}(S_1)\ldots\mathfrak{z}(S_n)$, где $\mathfrak{z}(S_i)$ — запись ситуации S_i. Множество всех протоколов машины M будем обозначать через $\Pi(M)$, дополнение $\Pi(M)$ до Z^* — через $\Pi'(M)$.

*) Если бы такого числа не нашлось, то R имел бы конечное дополнение и, значит, был бы НС-языком.

Лемма 8.5. *Для произвольной* ДЭ-*машины* M *множество* $\Pi'(M)$ *есть линейный язык, и для всякой* ДЭ-*машины* M *можно построить линейную* ОБ-*грамматику, порождающую* $\Pi'(M)$.

Предварительно докажем другую лемму.

Лемма 8.6. *Пусть* Э-*машина* M *имеет инструкции только типов* (i) *и* (iii) *и для любой ее непустой входной цепочки* x *существует самое большее одно* $[x, y]$-*вычисление, причем это вычисление происходит в такой последовательности: сначала читается символ* # *на входной ленте, затем (если* $|x| \geqslant 1$*) — первый символ входной цепочки* x*, затем создается рабочая ячейка, и далее шаги типов* (i) *и* (iii) *чередуются через один, пока не будет прочитан последний символ цепочки* x*; после этого машина либо сразу останавливается, либо, прежде чем остановиться, делает еще* один или два *шага типа* (iii). *Пусть, кроме того, входной и рабочий алфавиты машины* M *представлены соответственно в виде* $V = V' \cup \tilde{V}$, $W = W' \cup \tilde{W}$, *где* $V' \cap \tilde{V} = W' \cap \tilde{W} = \varnothing$. *Для произвольных двух языков* $L_1 \subseteq V'\tilde{V}^*$, $L_2 \subseteq W'\tilde{W}^*$ *обозначим через* $L_M(L_1, L_2)$ *множество всевозможных цепочек* xy *таких, что* $x \in L_1$, $y \in L_2$ *и не существует* $[x, y]$-*вычисления машины* M. *Тогда для любых двух* А-*грамматик* Γ_1 *и* Γ_2 *таких, что* $L(\Gamma_1) \subseteq V'\tilde{V}^*$, $L(\Gamma_2) \subseteq W'\tilde{W}^*$, *можно построить линейную* ОБ-*грамматику, порождающую* $L_M(L(\Gamma_1), L(\Gamma_2))$.

Доказательство леммы 8.6. Поскольку для данной входной цепочки x самое большее при одном значении y, которое мы будем обозначать $y(x)$, может существовать $[x, y]$-вычисление, ясно, что цепочка xy, где $x \in L(\Gamma_1)$, $y \in L(\Gamma_2)$, принадлежит $L_M(L(\Gamma_1), L(\Gamma_2))$ тогда и только тогда, когда выполняется одно из четырех условий: (а) цепочки $y(x)$ не существует; (б) $|y| < |y(x)|$; (в) $|y| > |y(x)|$; (г) существует такое $i = 1, \ldots, \min(|y|, |y(x)|)$, что i-е слева символы цепочек y и $y(x)$ различны. Дизъюнкция этих условий

равносильна дизъюнкции (б′) ∨ (в′) ∨ (г′), где (б′) = = (а) ∨ (б), (в′) = (а) ∨ (в), (г′) = (а) ∨ (г). Иначе говоря, $L_M(L(\Gamma_1), L(\Gamma_2)) = L' \cup L'' \cup L'''$, где L', L'', L''' — множества цепочек xy, $x \in L(\Gamma_1)$, $y \in L(\Gamma_2)$, удовлетворяющих (б′), (в′) и (г′) соответственно. Поэтому ввиду эффективной замкнутости класса линейных языков относительно объединения (стр. 170) нам достаточно указать способ построения линейных ОБ-грамматик для языков L', L'' и L'''. В силу теорем 5.2 и 5.8 это равносильно построению однодорожечных МП-машин, допускающих указанные языки, по конечным автоматам M_1 и M_2, допускающим $L(\Gamma_1)$ и $L(\Gamma_2)$ соответственно.

Машина M' для L' строится следующим образом: если на ее входной ленте записана цепочка xy, где $x \in V'\bar{V}^*$, $y \in W'\bar{W}^*$, то сначала она работает на входной ленте, как M_1, а на рабочей — как M; прочтя x, она начинает работать на входной ленте, как M_2, а на рабочей после каждого «входного» шага уничтожает одну ячейку; если после прочтения xy еще останутся рабочие ячейки, то машина уничтожает и их, после чего переходит в заключительное состояние; в противном случае после прочтения xy машина «ломается».

Машина M'' для L'' строится аналогично.

Машина M''' для L''' работает так. Имея на входной ленте xy, где $x \in V'\bar{V}^*$, $y \in W'\bar{W}^*$, она сначала работает так же, как M', но потом, не дойдя до конца x, переходит «в другой режим»: на входной ленте продолжает работать так же, а на рабочей не делает ничего. При этом она «запоминает» тот символ b, который должна была бы записать на рабочей ленте машина M (или M') сразу после прочтения содержимого той входной ячейки, которую M''' обозревает к началу работы в новом режиме; пусть к этому времени рабочая лента состоит из i ячеек (возможно, $i = 0$) *). Прочтя x, машина M''' снова начинает работать так же, как работала с этого момента M', пока не уничтожит всю рабочую ленту. Если к этому моменту входная цепочка будет прочитана целиком, машина ломается. В противном случае входная головка будет обозревать в данный момент $i+1$-й слева символ цепочки y, и очередной шаг

*) Разумеется, числа i машина «знать» не может.

будет состоять в следующем: как и на предыдущих шагах, машина имитирует работу M_2, но одновременно проверяет, совпадает ли обозреваемый входной символ с запомненным ранее символом b; если да — она ломается; если нет — продолжает работать просто как M_2, и если y окажется «M_2-допущенной», переходит в заключительное состояние.

Доказательство леммы 8.5. Пусть R — множество записей всевозможных ситуаций данной ДЭ-машины M. Ясно, что R есть A-язык; соответствующая А-грамматика строится без всякого труда. Обозначим через T множество всевозможных цепочек вида xy, где $x, y \in R$ и ситуация с записью y не является непосредственно достижимой из ситуации с записью x. Ввиду очевидного равенства $\Pi'(M) = R^*TR^* \cup C_zR^+$, теоремы 5.4 и замечания на стр. 170 нам достаточно построить линейную ОБ-грамматику, порождающую T.

Будем говорить, что инструкция $\mathcal{K}$ машины M применима к ситуации $S = (q, x', a\tilde{x}'', X', A\tilde{X}'')$, если левая часть $\mathcal{K}$ содержит q и при этом: если $\mathcal{K} = qb \to q'$, то $b = a$, если $\mathcal{K} = Bq \to q'$, то $B = A$; если $\mathcal{K} = q \to q'\Lambda$, то $X' \neq \Lambda$; если $\mathcal{K} = q \to q'\Pi$, то $\tilde{X}'' \neq \Lambda$. Будем обозначать через $R(\mathcal{K})$ множество записей всевозможных ситуаций машины M, к которым применима инструкция $\mathcal{K}$, и через R_0 — множество записей всевозможных ситуаций M, к которым неприменимы никакие инструкции. Как R_0, так и все $R(\mathcal{K})$ являются А-языками, и построение соответствующих А-грамматик очевидно. При этом $T = [R_0 \cap T] \cup \bigcup [R(\mathcal{K}) \cap T]$, где объединение берется по всем инструкциям M. Поскольку $R_0 \cap T = = R_0R$, достаточно указать способ построения линейной Б-грамматики для каждого из языков $R(\mathcal{K}) \cap T$.

Ввиду детерминированности машины M к каждой данной ситуации может быть применима только одна инструкция. Поэтому, если $x \in R(\mathcal{K})$ и $y \in R$, то ситуация S' с записью y тогда и только тогда не является непосредственно достижимой из ситуации S с записью x, когда S' не получается из S применением инструкции $\mathcal{K}$. А отсюда по лемме 8.6 вытекает, что достаточно построить Э-машину M, удовлетворяющую условию этой леммы и такую, что для цепочек $x \in R(\mathcal{K})$ и $y \in R$ тогда и только тогда существует $[x, y]$-вычисление

машины M, когда ситуация с записью y получается из ситуации с записью x применением инструкции $\mathscr{K}$. При этом возможны пять случаев, соответствующих возможным типам инструкций $\mathscr{K}$. Мы рассмотрим случай $\mathscr{K} = Aq \to q'$, предоставив остальные (вполне аналогичные) читателю. Машина M будет в этом случае, имея на входной ленте цепочку $qx'\nabla x''\tilde{X}'B\nabla A\tilde{X}''$ (где в силу выбора типа инструкции $A \in W$, $B \in W \cup \{\#\}$), работать следующим образом. Сначала она читает q и пишет на рабочей ленте q'. Затем до прочтения B просто переписывает входные символы на рабочую ленту. Прочтя B, она пишет на рабочей ленте ∇. (Подчеркнем, что это делается «недетерминированным образом»: машина может записать ∇ после прочтения любого символа из $W \cup \{\#\}$, но если в следующей входной ячейке не окажется ∇, то произойдет «поломка».) Затем, прочтя на входной ленте ∇, она пишет на рабочей B. Далее, прочтя A, пишет на рабочей ленте символ, который на входной непосредственно следует за A (опять «недетерминированным образом»: символ пишется наугад, и в случае несовпадения его со следующим входным символом машина ломается). Далее, читая $\tilde{X}''$, машина после прочтения каждого символа пишет на рабочей ленте (как описано выше) тот символ, который на входной следует за прочитанным, а прочтя последний символ из $\tilde{X}''$, переходит в специальное состояние $\tilde{q}$ (если она, «не угадав», что данный символ последний, вместо перехода в $\tilde{q}$ запишет что-нибудь на рабочей ленте, то на следующем шаге будет прочтен граничный маркер *) и произойдет поломка). После этого читается граничный маркер и наступает заключительное состояние. Выполнение для M условия леммы 8.6 очевидно. Итак, лемма 8.5 доказана.

Пусть теперь M — произвольная Э-машина. Обозначим через $R(M)$ множество записей всевозможных ситуаций M, через $R_1(M)$ — множество записей всевозможных начальных ситуаций M и через $R_2(M)$ — множество записей всевозможных ситуаций M, имеющих вид $(q_f, \#x, \ddagger, \Lambda, \#y)$, где q_f — заключительное состояние M. Положим, кроме того,

$$F(M) = \Pi(M) \cap [R_1(M)(R(M))^* R_2(M)];$$

иначе говоря, $F(M)$ — это множество протоколов тех вычислений машины M, с помощью которых значения аргумента вычисляемой ею функции f переводятся в значения самой f. Поскольку множество $C_Z F(M)$ (обозначение Z введено на стр. 268) представимо в виде

$$C_Z F(M) = \Pi'(M) \cup C_Z [R_1(M)(R(M))^* R_2(M)],$$

а $R_1(M)$ и $R_2(M)$ суть А-языки, причем А-грамматики для них строятся очевидным образом, из только что доказанной леммы непосредственно вытекает следующая

Лемма 8.7. *Для произвольной* ДЭ-*машины M множество $C_Z F(M)$ есть линейный язык, и для всякой* ДЭ-*машины M можно построить линейную* ОБ-*грамматику, порождающую $C_Z F(M)$.*

Нам понадобится еще следующая простая конструкция. Пусть $G = \{g_1, \ldots, g_n, \ldots\}$ — счетное множество символов и a, b — произвольные символы. Для каждой цепочки $\omega = g_{i_1} \ldots g_{i_s}$ положим $\lambda(\omega) = ba^{1+i_1}b \ldots$ $\ldots ba^{1+i_s}b$; кроме того, положим $\lambda(\Lambda) = bab$. Для каждого языка L в произвольном словаре, содержащемся в G, положим $\lambda(L) = \{\lambda(\omega) \mid \omega \in L\}$. Мы можем допустить, что для каждой рассматриваемой нами Э-машины M соответствующий словарь Z содержится в G, и рассмотреть язык $H(M) = \lambda(F(M))$. Имеет место

Лемма 8.8. *Для произвольного словаря V, содержащего a и b, и для произвольной* ДЭ-*машины M множество $C_V H(M)$ есть линейный язык, и для всякого словаря $V \supseteq \{a, b\}$ и всякой* ДЭ-*машины M можно построить линейную* ОБ-*грамматику, порождающую $C_V H(M)$.*

Доказательство. Имеем $C_V H(M) = C_V(\lambda(Z))^* \cup$ $\cup ((\lambda(Z))^* - H(M))$. Поскольку $\lambda(Z)$ конечно, для первого члена объединения по теореме 5.4 строится ОА-грамматика; второй член порождается линейной ОБ-грамматикой, схема которой получается из схемы соответствующей грамматики для $C_Z F(M)$ заменой в каждом правиле каждого вхождения каждого основного символа g_i вхождением цепочки $ba^{1+i}b$.

*) Мы можем, разумеется, считать граничные маркеры M отличными от $\#$.

Теперь для некоторых инвариантных свойств ОБ грамматик легко доказать нераспознаваемость.

Теорема 8.4. Пусть V — *произвольный словарь содержащий не менее двух символов,* E — *произвольно непустое множество натуральных чисел и* α_E — *свойств языка в словаре* V *иметь дополнение (до* V^**), число элементов которого конечно и является элементом* E*. Тогда* α_E *нераспознаваемо в классе линейных* ОБ-*грамматик и тем более в классе всех* ОБ-*грамматик с основным словарем* V.

Доказательство. Для произвольной ДЭ-машины M мощность множества $H(M)$ равна мощности $F(M)$, а эта последняя — мощности области определения вычисляемой данной машиной функции. Поэтому к проблеме распознавания α_E сводится проблема распознавания свойства машины M вычислять функцию *), область определения которой имеет конечную мощность являющуюся элементом E.

Следствие 1. *Следующие свойства языков нераспознаваемы в классе линейных (и тем более всех)* ОБ-*грамматик с заданным основным словарем мощности, большей или равной* 2:

а) *иметь конечное дополнение;*

б) *иметь дополнение, состоящее в точности из заданного конечного числа элементов (в частности, пустое).*

Действительно, для пункта а) достаточно взять в качестве E множество всех натуральных чисел, для пункта б) — множество, состоящее из одного числа.

Следствие 2. *Не существует алгоритма, позволяющего для любой* ОБ-*грамматики (или даже для любой линейной* ОБ-*грамматики) с данным основным словарем мощности, большей или равной* 2, *распознать, эквивалентна ли она заданной грамматике, порождающей* V^**. Тем более неразрешима общая проблема эквивалентности* ОБ-*грамматик.*

Следствие 3. *Не существует алгоритма, позволяющего для любых двух* ОБ-*грамматик* Γ_1 *и* Γ_2 *с данным основным словарем мощности, большей или равной* 2, *распознать, содержится ли* $L(\Gamma_1)$ *в* $L(\Gamma_2)$.

*) Точнее следовало бы говорить о машинах с некоторым фиксированным входным алфавитом и о проекции функции на некоторый фиксированный алфавит (ср. сноску на стр. 264).

В самом деле, по предыдущему следствию такой алгоритм невозможен даже для фиксированной грамматики Γ_1, если она порождает V^*.

Замечание. Ограничение на мощность словаря в теореме 8.4 и ее следствиях существенно (см. упражнение 8.16).

Для доказательства нераспознаваемости ряда других свойств мы воспользуемся следующей теоремой.

Теорема 8.5. *Пусть* U, V — *словари такие, что* $U \cap V = \varnothing$, $\mu(V) \geqslant 2$, *и* $\mathscr{T}$ — *класс* ОБ-*грамматик, содержащий все линейные* ОБ-*грамматики и такой, что соответствующий класс* $\mathscr{L}(\mathscr{T})$ *эффективно замкнут относительно объединения и относительно умножения слева на* U^* *и справа на* V^**. Тогда всякое свойство языков, нетривиальное в* $\mathscr{L}(\mathscr{T}_U)$, *справедливое для* U^*V^* *и сохраняющееся при операциях пересечения с* А-*языком и проектирования, нераспознаваемо в* $\mathscr{T}_{U \cup V}$*. При этом сохранение свойства при проектировании может быть заменено более слабым требованием: если* $L \subseteq U^*$, $y \in V^*$ *и язык* Ly *обладает данным свойством, то им обладает и язык* L.

Доказательство. Пусть α — свойство языков, удовлетворяющее описанным условиям. В силу одного из этих условий должен существовать язык $L_0 \in \mathscr{L}(\mathscr{T}_U)$, не обладающий свойством α. Для произвольного языка $L \in \mathscr{L}(\mathscr{T}_V)$ положим $L' = U^*L \cup L_0V^*$. Если при этом $L = V^*$, то $L' = U^*V^*$, так что α справедливо для L'. Если же $L \neq V^*$, то, фиксировав цепочку $y \in V^* - L$ и допустив, что свойство α имеет место для L', получим ввиду условий, наложенных на α, что этим свойством должен обладать и язык $L' \cap U^*y = L_0y$, а значит, и $\mathrm{Пр}_U(L_0y) = L_0$, что неверно. Таким образом, L' обладает свойством α тогда и только тогда, когда $L = V^*$. А это позволяет заключить, — поскольку L' строится по L эффективно, — что проблема распознавания совпадения языка класса $\mathscr{L}(\mathscr{T}_V)$ с V^* сводится к проблеме распознавания α в классе $\mathscr{T}_{U \cup V}$. Но свойство совпадать с V^* нераспознаваемо в $\mathscr{T}_V$ (следствие 1 из теоремы 8.4, пункт б)).

Следствие. *Следующие свойства языков нераспознаваемы в классе* ОБ-*грамматик с заданным основным словарем мощности, большей или равной* 4: (α) *быть однозначным* Б-*языком*; (β) *быть детерминированным языком* (*т. е. допускаться детерминированной* МП-*машиной*); (γ) *иметь бесконтекстное дополнение*; (δ) *быть* ОА-*языком*; (ε) *быть линейным языком*; (ζ) *быть металинейным языком*; (η) *быть итерационно-линейным языком*; (θ) *быть* ОАЕВ-*языком. Свойства* (α)—(η) *нераспознаваемы также в классе* ОБ-ОАЕВ-*грамматик, свойства* (α)—(ζ) *и* (θ)—*в классе итерационно-линейных грамматик, свойства* (α)—(ε)—*в классе металинейных грамматик, свойства* (α)—(δ)—*в классе линейных грамматик* (*при том же ограничении на словарь*).

В самом деле, все перечисленные классы грамматик дают классы языков, эффективно замкнутые относительно нужных операций (теорема 4.8, замечания на стр. 170, лемма 5.1); справедливость названных свойств для U^*V^* очевидна, сохранение свойства (δ) при нужных операциях обеспечивается теоремой 5.4, сохранение свойств (ε)—(θ)—замечаниями 2 и 3 в конце § 5.4, нетривиальность свойств (α)—(θ) в соответствующих классах при $\mu(U) \geqslant 2$—примерами из § 4.4 (для (α) и (β)), простой модификацией примера 2 из § 4.3*) (для γ), примером к следствию из теоремы 5.7 (для (δ)), примерами из упражнений 5.26, 5.28 (для (ε)—(θ)). Свойства (α) и (β) при проектировании могут не сохраняться, но удовлетворяют указанному в конце формулировки теоремы 8.5 более слабому требованию (упражнение 4.35). То же верно, как нетрудно убедиться, для (γ). Сохранение свойства (α) при пересечении с А-языком вытекает из упражнения 5.17, а), свойства (β)—из теорем 5.5 и 5.3, свойства (γ)—из теорем 4.8 и 5.4.

Замечание. Утверждения следствия из теоремы 8.5 справедливы для любого словаря мощности, большей или равной 2 (упражнение 8.11). В случае одноэлементного словаря свойства (α)—(θ) тривиальны.

*) Именно, нужно убрать разделительный символ b.

Другие примеры нераспознаваемых в классе Б-грамматик свойств и отношений см. в упражнениях 8.12, 8.13, 8.15, 8.22.

В заключение мы рассмотрим некоторые проблемы иного характера (второго из двух типов, указанных в § 8.1), а именно проблемы распознавания замещаемости, взаимозамещаемости и конфигураций. Мы покажем, что они неразрешимы уже для линейных языков; более того, для подходящих линейных языков неразрешимы и соответствующие частные проблемы.

Укажем способ построения этих языков. Пусть M—произвольная ДЭ-машина, в рабочем алфавите которой выделен символ a_0 такой, что все значения вычисляемой машиной M функции f являются цепочками в $\{a_0\}$. Множество всех тех n, для которых a_0^n суть значения f, будем обозначать E_M. При надлежащем выборе M мы можем, очевидно, получить в качестве E_M произвольное рекурсивно перечислимое множество. Будем, кроме того, считать, что машина M имеет единственное заключительное состояние q_0, причем оно отлично от начального и не встречается в левых частях инструкций; понятно, что класс множеств E_M при таком ограничении не сужается. Положим далее $V = \{a, b\}$ и построим по лемме 8.8 линейную ОБ-грамматику Γ_M, порождающую язык $C_V H(M)$. Рассмотрим произвольную цепочку $\omega \in \lambda(R_2(M))$ (см. стр. 272, 273). Эта цепочка единственным образом представляется в следующем виде: $\lambda(q_0)\lambda(\#x\nabla\mp)\lambda(\#)\lambda(y)$, где x и y—цепочки во входном и рабочем алфавитах M соответственно; обратно, всякая цепочка такого вида принадлежит $\lambda(R_2(M))$. Обозначим через $Q^1(M)$ и $Q^2(M)$ соответственно множества всевозможных цепочек вида

$$\lambda(q_0)\lambda(\#x\nabla\mp\nabla)c \quad \text{и} \quad b^{k+4}\lambda(\#x\nabla\mp\nabla)\lambda(\#)\lambda(y)\,bb,$$

где x, y—цепочки во входном и рабочем алфавитах M соответственно; $k = |y|$; c—символ, отличный от a и b. Положим $S^i(M) = \lambda(R_1(M)(R(M))^*)Q^i(M)$ $(i = 1, 2)$ и $L^i(M) = [\lambda(R_1(M)(R(M))^* R_2(M)) - H(M)]\,bb \cup S^i(M)$. Ясно, что $S^1(M)$—ОА-язык, а $S^2(M)$—линейный язык; соответствующие грамматики строятся без труда. Поэтому

лемма 8.8 дает возможность по машине M построить линейные ОБ-грамматики, порождающие $L^1(M)$ и $L^2(M)$.

Непосредственно ясно, что в языке $L^1(M)$ любая цепочка вида $\lambda(\#)[\lambda(a_0)]^k bb$, $k = 0, 1, \ldots$, замещаема на символ c, в то время как этот последний замещаем на $\lambda(\#)[\lambda(a_0)]^k bb$ тогда и только тогда, когда никакая цепочка из $H(M)$ не оканчивается вхождением $\lambda(\#)[\lambda(a_0)]^k bb$ — иначе говоря, когда a_0^k не является значением f. Что же касается языка $L^2(M)$, то для него принадлежность a_0^k множеству значений f равносильна замещаемости b^{k+4} на $\lambda(q_0)$. Заметим еще, что а) кодирование λ можно произвести так, чтобы было $\lambda(q_0) = baab$; б) в $L^1(M)$ символ c можно с тем же результатом заменить цепочкой $bbbb$.

Подобрав машину M так, чтобы множество значений f было нерекурсивным, т. е. чтобы не существовало алгоритма, позволяющего по числу k узнать, принадлежит ли оно этому множеству*), мы получаем конкретные линейные ОБ-грамматики Γ_1 с основным словарем $\{a, b, c\}$, Γ_1' и Γ_2 — обе с основным словарем $\{a, b\}$ — такие, что не существует алгоритмов, позволяющих для произвольной цепочки $x \in \{a, b\}^*$ распознать: а) является ли x конфигурацией языка $L(\Gamma_1)$ с результирующим c; б) является ли x конфигурацией первого ранга языка $L(\Gamma_1)$ с результирующим c; в) замещаема ли цепочка $bbbb$ на x относительно $L(\Gamma_1')$; г) замещаема ли x на $baab$ относительно $L(\Gamma_2)$; д) являются ли x и $bbbb$ взаимозамещаемыми относительно $L(\Gamma_1')$.

По поводу возможности уменьшить здесь мощности словарей см. упражнения 8.19 и 8.20, по поводу распознавания конфигураций высших рангов — упражнение 8.21.

*) Таким свойством будет обладать, например, следующая ДЭ-машина (ниже описывается, как обычно, только принцип ее работы). Входной алфавит машины есть $\{a, b\}$. В начале работы машина переписывает входную цепочку x на рабочую ленту; затем она проверяет, является ли эта цепочка кодом в алфавите $\{a, b\}$ некоторой Э-машины M' (см. стр. 258—259), и если да, то переписывает ее еще раз в другой зоне рабочей ленты (если нет, машина ломается) и далее работает в этой зоне, как M', используя «первый экземпляр» x на рабочей ленте для «получения инструкций» (это значит, что перед каждым очередным преобразованием «второго экземпляра» головка должна найти в «первом экземпляре» код той инструкции, которую следует применить на данном шаге). Если этот процесс окончится вычислением значения соответствующей функции от x, то все содержимое рабочей ленты заменяется однобуквенным кодом M' в алфавите $\{a_0\}$, после чего машина останавливается, перейдя предварительно в заключительное состояние; в противном случае машина работает вечно. Таким образом, цепочка a_0^k тогда и только тогда принадлежит множеству значений функции, которую вычисляет построенная машина, когда эта цепочка есть код некоторой самоприменимой машины M'. Остается воспользоваться леммой 8.1.

Упражнения

8.1. Закодировав всевозможные грамматики, как в доказательстве теоремы 2.4, назовем грамматику Г самопорождающей, если наименьший (в лексикографическом порядке) из ее кодов принадлежит $L(\Gamma)$. Показать, что самопорождаемость нераспознаваема в классе Г*). [*Указание.* Показать, что множество наименьших кодов несамопорождающих грамматик не порождается никакой грамматикой.]

8.2. Показать, что следующие свойства грамматик нераспознаваемы в классе Г:

а) иметь ограниченное растяжение (т. е. емкость, мажорируемую линейной функцией);

б) быть грамматикой без растяжения.

8.3. Можно ли перенести результат упражнения 8.1 на класс НС?

8.4. а) Пусть L_0 — произвольный НС-язык, L — произвольный непустой НС-язык и $\mathscr{L}$ — класс языков, содержащий $L_0 \cup L$ и не содержащий ни одного языка вида $L_0 \cup (L - \{x\})$, где $x \in L$. Показать, что свойство принадлежать $\mathscr{L}$ нераспознаваемо в классе НС.

б) То же с заменой языков $L - \{x\}$ на языки вида $L - L'$, где L' — непустое конечное подмножество L.

8.5. Показать, что в классе НС нераспознаваемо свойство принадлежать $\mathscr{L}_n^T$ (НС).

8.6. Язык L в словаре V имеет нетривиальную замещаемость, если найдется хотя бы одна пара цепочек $x, y \in V^*$, $x \neq y$, такая, что x является подцепочкой некоторой цепочки из L и при этом $x \underset{V, L}{\Rightarrow} y$. Показать, что свойство иметь нетривиальную замещаемость нераспознаваемо в классе НС.

8.7. Усилить лемму 8.1 и теорему 8.2, показав, что следующие множества не могут быть рекурсивно перечислимыми:

а) множество кодов несамоприменимых Э-машин;

б) множество кодов таких ДЭ-машин, что вычисляемые ими функции обладают некоторым свойством, справедливым для нигде не определенной функции, но не для всех частично рекурсивных функций.

8.8. Используя упражнение 8.7, б), показать, что следующие множества не могут быть рекурсивно перечислимыми:

*) Г — класс всех грамматик (стр. 30).

а) множество кодов грамматик, обладающих некоторым инвариантным и нетривиальным в классе Г свойством, справедливым для грамматик, порождающих пустой язык;

б) множество кодов НС-грамматик, для которых порождаемые ими языки принадлежат некоторому классу, удовлетворяющему условию леммы 8.4 или теоремы 8.3 (но не дополнение к такому множеству!).

8.9. Назовем свойство языков в словаре V «булевски перечислимым», если оно представимо в виде $B_{i_1} \vee \ldots \vee B_{i_k} \vee \ldots$, где B_i — булевы свойства, занумерованные с помощью некоторого эффективного кодирования булевых выражений (от переменных $X_1, \ldots, X^n \ldots$, где X_n означает «$\omega n \in L$»; ωn — цепочка с номером n, см. стр. 267), и $\{i_1, \ldots, i_k, \ldots\}$ — рекурсивно перечислимое множество натуральных чисел. Показать, что если $\mathscr{T}$ — класс грамматик такой, что соответствующий класс $\mathscr{L}(\mathscr{T})$ содержит вместе с каждым конечным языком все его НС-надъязыки и множество кодов НС-грамматик из $\mathscr{T}$ рекурсивно перечислимо, то свойство быть НС-языком, принадлежащим $\mathscr{L}(\mathscr{T})$, булевски перечислимо. [*Указание.* Использовать упражнение 8.8, б).]

8.10. Опираясь непосредственно на лемму 8.8, доказать, что в классе линейных языков нераспознаваемы свойства быть ОА-языком и иметь бесконтекстное дополнение.

8.11. Показать, что все результаты следствия из теоремы 8.5 справедливы для любого словаря мощности, большей или равной 2. [*Указание.* Установить сводимость проблем распознавания свойств (α) — (θ) для словаря мощности 4 к соответствующим проблемам для словаря мощности 2.]

8.12. Доказать нераспознаваемость в классе $Б_V$, где V — словарь мощности, большей или равной 2, следующих свойств языков:

а) порождаться Б-грамматикой, активная емкость которой не превышает заданного числа k;

б) быть ОАЕ-языком;

в) удовлетворять условию $x \underset{V,L}{\Rightarrow} y$, где x и y — произвольные фиксированные различные цепочки в V;

г) иметь заданную конфигурацию заданного ранга с заданным результирующим;

д) содержать бесконечный ОА-язык [Ginsburg 1966, лемма 4.3.4].

8.13. Доказать нераспознаваемость в классе $Б_V$, где V — словарь мощности, большей или равной 2, следующих отношений между языками L_1 и L_2:

а) пустоты пересечения $L_1 \cap L_2$;

б) бесконтекстности пересечения $L_1 \cap L_2$;

в) существования конечного автомата с выходом (упражнение 5.8), переводящего L_1 в бесконечное подмножество L_2 [Ginsburg 1966, теорема 4.3.2].

8.14. Доказать, что для словаря V мощности, большей или равной 2, невозможен алгоритм, позволяющий для любых двух систем n цепочек $x_1, \ldots, x_n$ и $y_1, \ldots, y_n$ в V узнать, существует ли такая (непустая) последовательность чисел $i_1, \ldots, i_k$ ($i_1, \ldots, i_k = 1, \ldots, n$), что $x_{i_1} \ldots x_{i_k} = y_{i_1} \ldots y_{i_k}$ (теорема Поста о проблеме соответствий; см. [Post 1946, Марков 1954]).

[*Указание.* Доказать нераспознаваемость в классе линейных ОБ-грамматик с основным словарем $\{a, b, c, d, e\}$ свойства языка иметь непустое пересечение с $L_0\{d, e\}^*$, где $L_0 = \{xc\hat{x} \mid x \in \{a, b\}^*\}$. При этом можно рассуждать по аналогии с доказательством теоремы 8.5.]

8.15. Используя упражнение 8.14, доказать нераспознаваемость однозначности грамматики в классе $Б_V$, где $\mu(V) \geqslant 2$.

8.16. Указать алгоритмы, распознающие для любых ОА-грамматик (а тем самым и для любых ОБ-грамматик с одноэлементным основным словарем — см. замечание к теореме 4.6) свойства и отношения, перечисленные в следствиях из теоремы 8.4.

8.17. Показать, что если L — рекурсивный язык, то для любой цепочки x дополнения к множествам $\Phi_x(L)$ и $\Psi_x(L)$ (см. стр. 318) рекурсивно перечислимы.

8.18. Пусть R — рекурсивное множество точек целочисленной плоскости (m, n) $(m, n = 0, 1, \ldots)$, проекция которого на ось m не рекурсивна. Положим

$$L_1 = \{bba^{n+1} \mid n = 0, 1, \ldots\} \cup \{a^{m+1}ba^{n+1} \mid (m, n) \notin R\};$$

$$L_2 = \{(ba^{m+1}b)^p (b^2a^{n+1}b^2)^q \mid (m, n) \notin R;\ p, q = 1, 2, \ldots\} \cup$$
$$\cup \{(ba^{m+1}b)^{2p} (b^2a^{n+1}b^2)^q \mid (m, n) \in R;\ p, q = 1, 2, \ldots\}.$$

Показать, что:

а) множества $\Phi_{bb}(L_1)$ и $\Psi_{bb}(L_1)$ не рекурсивны;

б) для любой цепочки $x \in \{a, b\}^*$ множества $\Phi_x(L_2)$ и $\Psi_x(L_2)$ рекурсивны, однако $\Phi(L_2)$ и $\Psi(L_2)$ не рекурсивны.

[Гладкий 1963 (б)].

8.19. Показать, что для любого рекурсивного языка L в одноэлементном словаре $\Phi(L)$ и $\Psi(L)$ рекурсивны [Гладкий 1963 (б)].

8.20. Построить линейный язык в двухэлементном словаре с неразрешимой проблемой распознавания конфигураций.

8.21. Указать примеры линейных языков, для которых не существует алгоритмов, позволяющих по заданной цепочке x распознать:

а) является ли x конфигурацией заданного ранга r (для каждого r — свой язык);

б) является ли x конфигурацией.

[Лучкин 1966].

[*Указание.* Использовать конструкцию упражнения ПП. 7.]

8.22. Показать, что не существует алгоритма, позволяющего для любой Б-грамматики Г найти наименьшее число вспомогательных символов Б-грамматики, эквивалентной Г. [*Указание.* Использовать упражнение 4.21.]

ПРИЛОЖЕНИЕ I

СИСТЕМЫ СОСТАВЛЯЮЩИХ И ДЕРЕВЬЯ СИНТАКСИЧЕСКОГО ПОДЧИНЕНИЯ

В современной лингвистике используются различные способы описания синтаксической структуры предложения, из которых наиболее употребительны два — описание с помощью систем составляющих и описание с помощью деревьев синтаксического подчинения. Мы дадим здесь краткое изложение этих способов, а также правил перехода от одного способа к другому.

§ П1.1. Системы составляющих

Пусть x — произвольная непустая цепочка в словаре V. Множество C отрезков цепочки x называется системой составляющих этой цепочки, если оно удовлетворяет следующим двум условиям:

I. C содержит отрезок, состоящий из всех точек цепочки x, и все одноточечные отрезки x.

II. Любые два отрезка из C либо не пересекаются, либо один из них содержится в другом.

Элементы C мы будем называть составляющими. Одноточечные отрезки будут называться точечными составляющими, отрезок, состоящий из всех точек цепочки, — полной составляющей; полную и точечные составляющие мы будем иногда называть тривиальными.

Для наглядного изображения системы составляющих можно пользоваться следующим простым способом: заключать каждую нетривиальную составляющую в скобки, причем левую и правую скобки, отвечающие одной составляющей, помечать одной и той же меткой так, чтобы разные пары скобок были помечены разными метками;

в качестве меток можно использовать хотя бы натуральные числа. Можно, впрочем, обойтись и без меток, так как в силу пункта II определения системы составляющих для каждой левой скобки можно однозначно указать соответствующую ей правую и обратно (упражнение П1.1). Мы будем пользоваться скобочным изображением как с метками, так и без меток — где как удобнее.

Пример. Две разные системы составляющих (из многих возможных — см. упражнение П1.2) одной и той же цепочки *babcadebaccdabab* можно изобразить так:

(1) $\underset{1\,2\,3}{(}\underset{3}{(}\underset{2}{(}ba)\,\underset{4}{b)}\,\underset{4}{c}a\,\underset{1\,5\,6}{(}d\,\underset{6\,7}{(}e\,\underset{8}{(}ab)\underset{7\,9}{)}\underset{9\,5}{)}\underset{\cdot}{)}\underset{\cdot}{(}\underset{\cdot}{(}acc)\,\underset{\cdot}{(}d\,\underset{\cdot}{(}ab)\underset{\cdot}{)}\underset{\cdot}{)}\underset{\cdot}{(}ab)\underset{\cdot}{)}$;

(2) $\underset{1}{b}\,\underset{2}{(}a\,\underset{3}{(}b\,\underset{4}{(}c\,\underset{5}{(}ad\,\underset{6}{(}e\,\underset{7}{(}ba\,\underset{7}{(}cc)\,\underset{6}{d)}\,\underset{5}{a)}\underset{4}{)}\underset{3}{)}\underset{2}{)}\underset{1}{)}\,bab\underset{\cdot}{)}\underset{\cdot}{)}\underset{\cdot}{.}$

Если цепочка интерпретируется как предложение естественного языка, то система составляющих может быть использована в качестве способа выражения информации о его синтаксической структуре (или, как обычно говорят, способа изображения синтаксической структуры предложения). Именно, такая информация может представлять собой список словосочетаний, т. е. тех «кусков» предложения, которые в каком-то интуитивном смысле являются «синтаксически связными». Эмпирические соображения позволяют сделать допущение, что словосочетания, во-первых, образуют отрезки и, во-вторых, не имеют «частичных пересечений», т. е. удовлетворяют пункту II определения системы составляющих. Таким образом, нетривиальные составляющие — это (при подходящем выборе системы составляющих!) как раз словосочетания. (Тривиальные добавляются для придания системе формальной законченности.)

Пример. Предложение

(3) *Онегин, добрый мой приятель,*
родился на брегах Невы

допускает следующую «естественную» систему составляющих:

(4) (*Онегин*, (*добрый* (*мой приятель*))),
(*родился* (*на* (*брегах Невы*))).

Аналогичную роль играют составляющие в искусственных формализованных языках, используемых в математике и в программировании. Так, формула логики предикатов

(5) $$\forall x (F(x) \& \neg (G(x) \vee \neg H(x)))$$

допускает следующую «естественную» систему составляющих (мы записываем ее с помощью квадратных скобок, так как круглые входят здесь в словарь языка):

$$\underset{1}{[}\forall x\underset{1\,2\,3}{]}\underset{3}{[}\underset{3\,4}{(}\underset{5\,6}{[}F(x)\underset{6}{]}\&\underset{5\,6}{[}\neg\underset{7}{[}\underset{6}{(}\underset{6}{[}G(x)\underset{6}{]}\vee\underset{7}{[}\neg\underset{8}{[}H(x)\underset{8}{]}\underset{8\,7}{)}\underset{7}{]}\underset{5\,4}{)}\underset{4\,2}{]}\underset{2}{)}\underset{1}{]}.$$

Та же формула в бесскобочной записи

(5′) $$\forall x \,\&\, Fx \neg \vee Gx \neg Hx$$

допускает «естественную» систему составляющих

$$\underset{1}{[}\forall x\underset{1\,2\,3}{]}\underset{3\,4}{[}\&\underset{5\,6}{[}Fx\underset{6}{]}\underset{5\,6\,7}{[}\neg\underset{8}{[}\vee\underset{8\,7}{[}Gx\underset{7}{]}\underset{5\,4}{[}\neg\underset{8}{[}Hx\underset{8}{]}\underset{7\,5\,4}{]}\underset{4\,2}{]}\underset{2}{]}\underset{1}{]}.$$

Среди многочисленных систем составляющих, которые имеет предложение естественного языка, лишь весьма немногие «правильны», т. е. адекватно отражают синтаксическое строение предложения. При этом понятие «правильной» системы составляющих не абсолютно — оно зависит от соглашений лингвистического характера, отражающих определенные содержательные представления о синтаксической структуре предложения данного языка. Так, из двух возможных систем составляющих для одного и того же предложения

(6) (*Все* (*эти известия*)) (*произвели* (*на меня*) (*удручающее впечатление*))

и

(6′) (*Все* (*эти известия*)) *произвели* (*на меня*) (*удручающее впечатление*)

первая отвечает традиционному представлению о подлежащем и сказуемом как главных членах предложения, вторая — более новому взгляду на подлежащее и дополнения как равноправно подчиненные сказуемому «актанты».

При фиксированной системе соглашений предложение также может иметь несколько «правильных» систем

составляющих. Нередко эти системы соответствуют различным толкованиям смысла предложения. В лингвистике такое явление — наличие у одного предложения двух или более правильных «синтаксических анализов» — известно под названием синтаксической омонимии*). Например, предложение

(7) *Геологи и студенты из Тюмени уехали в Москву*

допускает по меньшей мере три толкования: 1. «Тюменские геологи и тюменские студенты уехали откуда-то (не обязательно из Тюмени) в Москву». 2. «Какие-то (не обязательно тюменские) геологи и тюменские студенты уехали откуда-то (не обязательно из Тюмени) в Москву». 3. «Какие-то (не обязательно тюменские) геологи и какие-то (не обязательно тюменские) студенты уехали из Тюмени в Москву». Этим толкованиям отвечают соответственно следующие системы составляющих:

(8) ((*Геологи и студенты*) (*из Тюмени*)) (*уехали* (*в Москву*)).

(9) (*Геологи и* (*студенты* (*из Тюмени*))) (*уехали* (*в Москву*)).

(10) (*Геологи и студенты*) ((*из Тюмени*) *уехали* (*в Москву*)).

Заметим, что синтаксическая омонимия нередко встречается и в таких предложениях, которые обычно не кажутся нам двусмысленными, поскольку мы в состоянии однозначно выбрать нужное толкование, опираясь на имеющиеся у нас сведения о действительности. Однако при автоматическом синтаксическом анализе такие случаи синтаксической омонимии также должны учитываться. Рассмотрим, например, предложения:

(11) *Гастроли балета на Кавказе проходят успешно*

и

(12) *Гастроли балета на льду проходят успешно.*

Если носитель русского языка, научившийся строить системы составляющих и понимающий смысл этих

*) Что касается искусственных формализованных языков, то они обычно строятся так, чтобы синтаксической омонимии не было.

предложений, получит задание указать для них «пр вильные» системы составляющих, то он, надо полагат без труда построит следующие системы:

(а) ((*Гастроли балета*) (*на Кавказе*)) (*проходят успешно*),

(б′) (*Гастроли* (*балета* (*на льду*))) (*проходят успешно*).

Однако такой анализ существенным образом оп рается на знание некоторых внеязыковых фактов (имее ся нечто, называемое «балет на льду», но нет ничег называемого «балет на Кавказе»; понятие «гастроли н Кавказе» имеет смысл, а «гастроли на льду» — нет). Ч ловек, не знающий этих фактов, должен будет допусти для первого предложения еще одну систему

(а′) (*Гастроли* (*балета* (*на Кавказе*))) (*проходят успешно*),

аналогичную (б′), а для второго — систему

(б) ((*Гастроли балета*) (*на льду*)) (*проходят успешно*),

аналогичную (а).

Так же будет действовать, разумеется, и машин (вернее, алгоритм синтаксического анализа).

Наконец, следует заметить, что различные «правил ные» системы составляющих (при фиксированной сист ме лингвистических соглашений) могут и не соответс вовать каким-либо ощутимым различиям в толковани смысла. Примером могут служить следующие две сист мы составляющих:

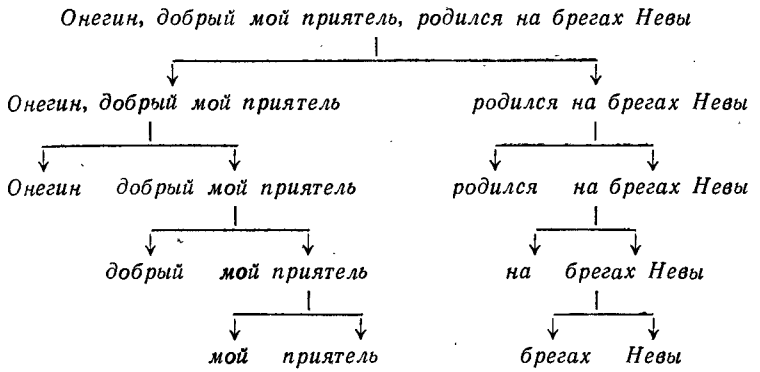
(13) *Мы увидели* (*древние* (*стены города*));

(14) *Мы увидели* ((*древние стены*) *города*).

Введем теперь следующее определение.

Если A и B — составляющие некоторой системы то выражение «B непосредственно вложен в A» будет означать, что $A \subset B$ и в C нет составляюще вложенной в A, содержащей B и отличной от A и от B Будем писать в этом случае $B \subset\subset A$ или $A \supset\supset B$.

Нетрудно видеть (упражнение П1.3), что граф $\langle C; \supset \supset\rangle$ является деревом, корнем которого служит полная составляющая и висячими узлами — точечные составляющие. Это дерево называется деревом составляющих; его графическое изображение может служить еще одним наглядным способом записи системы составляющих. Например, система составляющих (4) предложения (3) представляется следующим деревом:



Рассматривая составляющую как узел дерева составляющих, мы можем определить ее высоту, ранг и степень, как на стр. 18. Вместо «ширина (высота, степень) дерева $\langle C; \supset \supset\rangle$» будем говорить «ширина (высота, степень) системы C».

Систему составляющих, имеющую бинарное дерево, мы также будем называть бинарной.

Заметим, что если ширина системы C равна l, то длина составляющей ранга r не превосходит l^r. Поэтому каждый отрезок цепочки, длина которого больше или равна l^r, содержит точку, являющуюся левым или правым концом какой-либо составляющей ранга, не меньшего r.

При изучении систем составляющих, связанных с грамматиками, оказывается полезной следующая

Лемма П1.1. *Пусть C — система составляющих цепочки xyz, l — ширина C и $|y| \geqslant 2n \cdot l^{2n}$. Тогда найдутся точки $\alpha_1 \leqslant \ldots \leqslant \alpha_n < \beta_n \leqslant \ldots \leqslant \beta_1$ цепочки xyz такие, что $[\alpha_1, \beta_1], \ldots, [\alpha_n, \beta_n] \in C$ и либо $\alpha_1 < \ldots < \alpha_n$*

и $\alpha_1, \ldots, \alpha_n$ — *точки отрезка* y, *либо* $\beta_1 > \ldots > \beta_n$ $\beta_1, \ldots, \beta_n$ — *точки отрезка* y.

Доказательство. Представим y в виде $y = y_1 \ldots y_{2n}$, где $|y_i| \geqslant l^{2n}$, $i = 1, \ldots, 2n$. Каждый от зок y_i содержит хотя бы одну точку, являющуюся кон некоторой составляющей ранга, большего или равн $2n$. Среди этих $2n$ точек найдется либо n левых кон либо n правых — пусть для определенности первое. резок y содержит, таким образом, точки $\gamma_1 < \ldots <$ являющиеся левыми концами некоторых составляющ рангов, бóльших или равных $2n$. Обозначим прав концы этих составляющих через $\delta_1, \ldots, \delta_n$ соответ венно. Если все они лежат вне y, то, очевидно, $\delta_n \leqslant \ldots \leqslant \delta_1$, так что последовательность $\gamma_1, \ldots, \gamma_n, \delta_n, \ldots, \delta_1$ удовлетворяет нужному условию. Если же как либо точка δ_j принадлежит y, то составляющая $A = [\gamma_j, \delta_j]$ содержится в y целиком; она в свою очер содержит вложенную последовательность составля щих $A = [\varepsilon_1, \eta_1] \supset [\varepsilon_2, \eta_2] \supset \ldots \supset [\varepsilon_{2n}, \eta_{2n}]$, и либо сре точек $\varepsilon_1, \ldots, \varepsilon_{2n}$, либо среди точек $\eta_1, \ldots, \eta_{2n}$ име ся не менее n попарно различных; если, например, им место первое и при этом $\varepsilon_{i_1} < \ldots < \varepsilon_{i_n}$, то последо тельность $\varepsilon_{i_1}, \ldots, \varepsilon_{i_n}, \eta_{i_n}, \ldots, \eta_{i_1}$ является искомой.

Важными характеристиками составляющей являют также степень левого ветвления, степен правого ветвления и степень гнездов ния (обозначения: $y^{\text{л}}(A)$, $y^{\text{п}}(A)$, $\phi(A)$).

Степени левого и правого ветвления определяют так: I. Если A — полная составляющая, то $y^{\text{л}}(A) = y^{\text{п}}(A) = 0$. II. Если для составляющей A значен функций $y^{\text{л}}$ и $y^{\text{п}}$ определены и если $A_0, A_1, \ldots, A_k$ все непосредственно вложенные в A составляющие, з нумерованные слева направо, то

$$y^{\text{л}}(A_i) = y^{\text{л}}(A) + k - i; \quad y^{\text{п}}(A_i) = y^{\text{п}}(A) + i.$$

Для определения степени гнездования введем предв рительно следующее понятие. Назовем последовател ность составляющих $[\alpha_0, \beta_0], [\alpha_1, \beta_1], \ldots, [\alpha_n, \beta_n]$ гне дом глубины n, охватывающим составляющ $[\alpha_n, \beta_n]$, если $\alpha_0 < \alpha_1 < \ldots < \alpha_n \leqslant \beta_n < \ldots < \beta_1 <$ Степень гнездования составляющей A — это, по опред лению, наибольшая глубина гнезда, охватывающего A.

Наибольшие значения $y^{\text{л}}(A)$, $y^{\text{п}}(A)$ и $\phi(A)$ для $A \in C$ называются соответственно степенью левого ветвления, степенью правого ветвления и степенью гнездования системы C и обозначаются $\tilde{y}^{\text{л}}(C)$, $\tilde{y}^{\text{п}}(C)$, $\tilde{\phi}(C)$. Величина $\tilde{\phi}(C)$ есть, таким образом, наибольшая глубина гнезда, содержащегося в системе C.

Степень гнездования составляющей можно определить также индуктивно следующим образом. Пусть $A_0, \ldots, A_k$ — все составляющие, непосредственно вложенные в составляющую A, занумерованные слева направо. Будем называть составляющую A_0 левой, A_k — правой, $A_1, \ldots, A_{k-1}$ (если $k > 1$) — средними. Кроме того, полную составляющую будем считать средней. Среди левых и правых составляющих будем различать углубляющие и неуглубляющие, которые определим индуктивно. Именно: если составляющая A средняя, то непосредственно вложенные в нее левая и правая составляющие будут неуглубляющими; если A левая, то непосредственно вложенная в A левая составляющая — неуглубляющая, а непосредственно вложенная в A правая составляющая — углубляющая, когда A неуглубляющая, и неуглубляющая, когда A углубляющая; если A правая, симметрично. Кроме тотого, все средние составляющие считаем углубляющими. Теперь $\phi(A)$ определяется так: I. Если A — полная, то $\phi(A) = 0$. II. Пусть $\phi(A)$ определено и B непосредственно вложена в A. Тогда $\phi(B) = \phi(A)$, если B — неуглубляющая составляющая, и $\phi(B) = \phi(A) + 1$, если B — углубляющая составляющая. Эквивалентность двух определений степени гнездования без труда доказывается индукцией по высоте.

Назовем далее последовательность составляющих $[\alpha_0, \beta_0], [\alpha_1, \beta_1], \ldots, [\alpha_n, \beta_n]$ левым (правым) полугнездом глубины n, охватывающим составляющую $[\alpha_n, \beta_n]$, если $\alpha_0 < \alpha_1 < \ldots < \alpha_n \leqslant \beta_n \leqslant \ldots \leqslant \beta_1 \leqslant \beta_0$, соответственно $\alpha_0 \leqslant \alpha_1 \leqslant \ldots \leqslant \alpha_n \leqslant \beta_n < \ldots < \beta_1 < \beta_0$. Наибольшую глубину левого (правого) полугнезда, охватывающего данную составляющую A, обозначим $\phi^{\text{л}}(A)$, соответственно $\phi^{\text{п}}(A)$. Очевидно,

$$\phi(A) \leqslant \min(\phi^{\text{л}}(A), \phi^{\text{п}}(A)).$$

В то же время непосредственной индукцией по высоте A легко показать, что $\phi^{л}(A) \leqslant y^{п}(A)$, $\phi^{п}(A) \lessapprox y^{л}(A)$. Поэтому $\phi(A) \leqslant \min(y^{л}(A), y^{п}(A))$, откуда $\tilde{\phi}(C) \lessapprox \min(\tilde{y}^{л}(C), \tilde{y}^{п}(C))$.

Широко известна гипотеза В. Ингве [Wngve 1960], согласно которой в естественных языках имеются специальные механизмы, обеспечивающие ограниченность степеней левого ветвления предложений (в то время как степени правого ветвления ничем в принципе не ограничены — ср. хотя бы конструкции типа *Дом, который построил Джек*) *). Имеются, однако, языки, для которых это предположение заведомо не подтверждается (например, венгерский — см. [Гладкий — Мельчук 1969, стр. 101]); более того, и в языках, с фактами которы данная гипотеза на первый взгляд хорошо согласуется, можно, по-видимому, при более детальном исследовании обнаружить конструкции довольно большой левой глубины, вполне приемлемые не только грамматически, но и стилистически (ср., в частности, приводимые в [Падучева 1966] примеры для русского языка).

Значительно более адекватной характеристикой «синтаксической громоздкости» предложения является, видимо, степень гнездования. В русском языке, например, хотя и можно строить грамматически правильные предложения со сколь угодно большими значениями $\tilde{\phi}(C)$, но уже очень скоро они начинают восприниматься как стилистически недопустимые ввиду их крайней громоздкости; ср. хотя бы предложение ((*Прочитавший* (1 2 (*полученное* (3 4 5 (*от* (6 *опубликовавшего* (7 8 (*единственную* (9 *посвященную* (10 9 8 *Иванова*)) *монографию*)) 7 6 *ученого*)) 5 4 *письмо*)) 3 2 *брат*) (1 *был недовольен*)), для которого степень гнездования равна 5. См. об этом, например, [Мартыненко 1971].

Построенная для предложения система составляющих указывает в нем словосочетания разных «уровней» (составляющие разной высоты), не вводя при этом никакой иерархии среди словосочетаний одного «уровня».

*) По поводу аргументации, выдвигаемой в пользу этой гипотезы, см. § 7.1.

Между тем, в предложении естественного языка часто интуитивно ощущается «главенствование» некоторого словосочетания над другими, в нем не содержащимися. Чтобы в некоторой степени отразить этот факт, можно поступить следующим образом. Пусть C — система составляющих цепочки x. Для каждой неточечной составляющей $A \in C$ выделим в множестве всех составляющих, непосредственно вложенных в A, какую-либо одну составляющую A', которую будем называть **главной**. Множество всех главных составляющих обозначим C' и назовем **иерархизацией** системы C. Упорядоченную пару $\langle C, C' \rangle$ назовем **иерархизованной системой составляющих**.

Например, система (4) предложения (3) может быть иерархизована следующим образом (именно такую иерархизацию можно считать интуитивно естественной):

(15) (*Онегин*, (*добрый* (*мой приятель*))),
(*родился* (*на* (*брегах Невы*))).

(Главные составляющие здесь подчеркнуты.)

Вообще говоря, «естественная» система составляющих может допускать более одной «естественной» иерархизации. Например, в системе

(16) *Студенты встретили* (*больного* (*врача Иванова*))

можно взять в качестве главной составляющей (внутри составляющей *больного врача Иванова*) либо *больного*, либо *врача Иванова*; при первом выборе естественно считать, что речь идет о «больном врача Иванова», во втором — о «больном враче Иванове».

При описании предложения естественного языка с помощью системы составляющих часто используется еще один способ введения в эту систему дополнительной информации: рассматривается ее отображение в множество всех подмножеств некоторого конечного множества, элементы которого называются **метками** и содержательно интерпретируются как символы «синтаксических классов» слов и словосочетаний. Упорядоченную тройку $\langle C, W, \varphi \rangle$, где C — система составляющих, W — множество меток и φ — отображение C в 2^W, мы будем называть **размеченной системой составляющих**.

Пример. Пусть множество W содержит, в частности, следующие элементы (одновременно с их перечислением приводится содержательная интерпретация):

ПРЕДЛ — «предложение»;

$\widetilde{V}^{t}_{xyvw}$ — «группа непереходного глагола в роде x, числе y, времени v и лице w»;

V^{t}_{xyvw} — «непереходный глагол в роде x, числе y, времени v и лице w»;

$\widetilde{\widetilde{V}}^{t}_{xyvw}$ — «группа переходного глагола в роде x, числе y, времени v и лице w»;

$\widetilde{V}^{t}_{xyvw}$ — «переходный глагол в роде x, числе y, времени v и лице w»;

$\widetilde{S}_{xyz}$ — «группа существительного рода x в числе y и падеже z»;

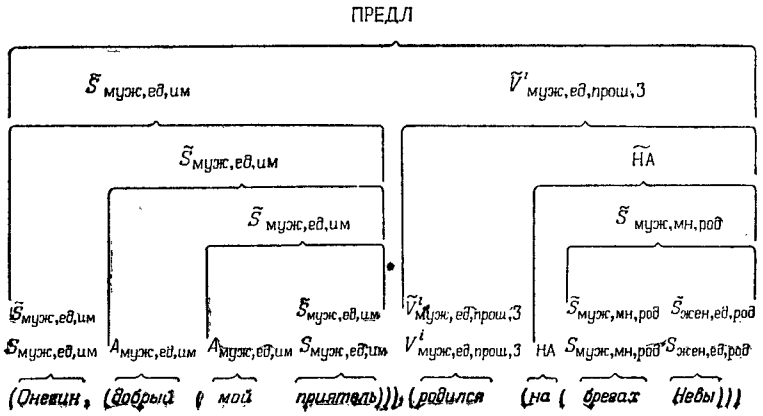
S_{xyz} — «существительное рода x в числе y и падеже z»;

A_{xyz} — «прилагательное в роде x, числе y и падеже z»;

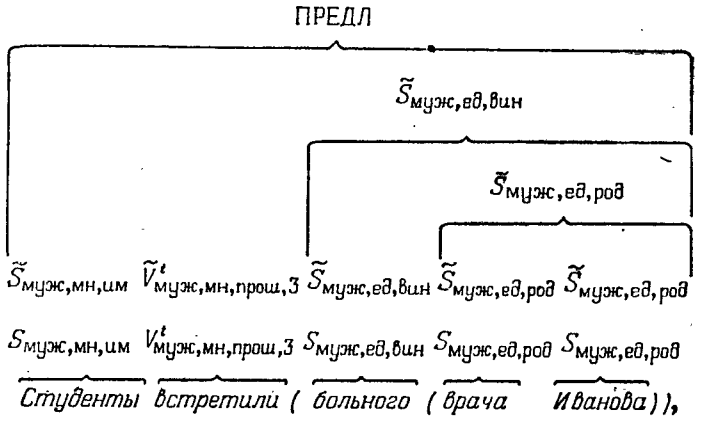
$\widetilde{\text{НА}}$ — «предложная группа с предлогом *на*»;

НА — «предлог *на*».

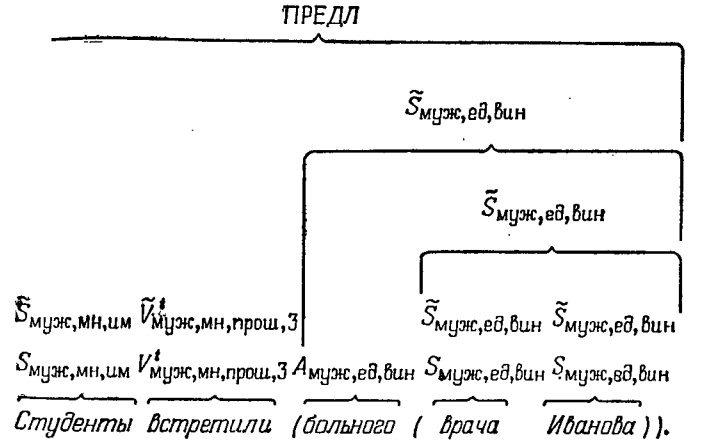
Тогда система составляющих (4) предложения (3) допускает следующую «естественную» разметку (в записи каждая составляющая A снабжена акколадой, около которой выписаны все элементы множества $\varphi(A)$):



Одна и та же «естественная» система составляющих может допускать более одной «естественной» разметки. Так, система (16) может быть размечена либо таким образом:



либо таким:



Заметим, что в последнем примере каждая из двух разметок по смыслу совместима только с одной из приведенных иерархизаций. Однако вполне возможны

различные разметки, совместимые с одной и той же иерархизацией — например, в системе *Я (сел (в метро)* составляющей *метро* можно приписать либо мет $\tilde{S}_{\text{ср, ед, вин}}$ и $S_{\text{ср, ед, вин}}$, либо $\tilde{S}_{\text{ср, ед, предл}}$ и $S_{\text{ср, ед, предл}}$, и оба варианта совместимы с иерархизацией

Я (сел в метро)).

§ П1.2. Деревья подчинения

Пусть x — произвольная непустая цепочка в слова V и X — множество всех точек x. Произвольное бина ное отношение $\rightarrow$ на X такое, что граф $\langle X; \rightarrow\rangle$ являетс деревом, мы будем называть отношением синта сического подчинения (или просто отнош нием подчинения) для x. Само дерево $\langle X; \rightarrow\rangle$ б дет называться деревом (синтаксического подчинения для x.

При графическом изображении дерева подчинени мы будем помещать точки цепочки x на горизонтально прямой так, как условлено в § 1.1, и для всякой пар точек α, β, для которой $\alpha \rightarrow \beta$, будем проводить из α β стрелку, причем таким образом, чтобы все стрелк были по одну сторону от прямой.

Для дерева подчинения обычным образом опред ляются подчинение, зависимость, группа зависимости ширина куста узла, ширина дерева, ранг узла, высот узла и дерева (см. стр. 18—19). Сверх того всякое мно жество, полученное из группы зависимости точки α вы брасыванием групп зависимости некоторых (или всех точек, подчиненных α, мы будем называть усеченно группой зависимости точки α.

Пример. На рис. 15 изображено несколько раз личных деревьев подчинения для одной и той же цепоч ки *abcdefg*.

Деревья подчинения могут быть использованы ка один из способов изображения синтаксической структу ры предложения. Именно, информация о синтаксическо строении предложения может представлять собой набо сведений о «главенствовании» одних слов (точнее, вхож дений слов) в предложении над другими; задать тако набор — значит задать некоторый граф на множеств

точек цепочки (предложения). Из интуитивных соображений, обсуждать которые мы здесь не будем, вытекает, что этот граф можно считать деревом.

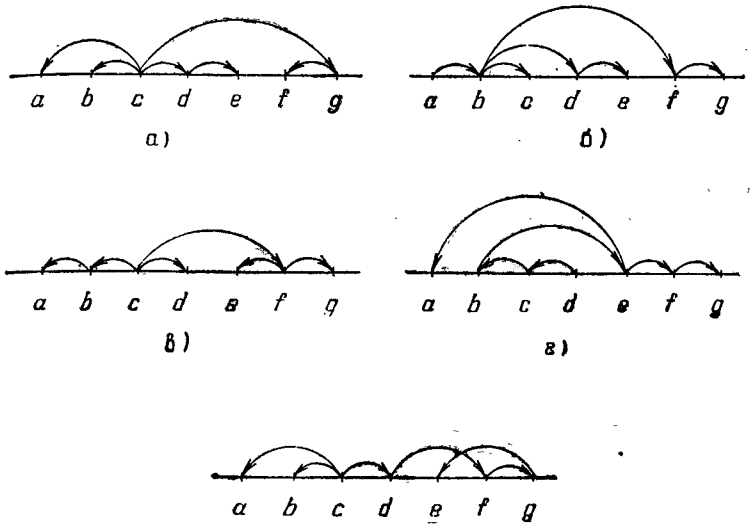
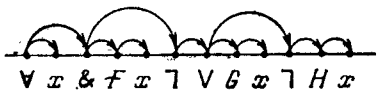


Рис. 15.

Пример. Следующее дерево подчинения для предложения (3) может считаться «естественным»:

(17) 

Аналогичным образом можно строить деревья подчинения для выражений формализованных языков. Так, для формулы (5′) (стр. 284) «естественное» дерево подчинения будет иметь вид



Понятие «правильного» дерева подчинения для предложения естественного языка зависит, вообще говоря,

от некоторых лингвистических соглашений (ср. анал гичное замечание о системах составляющих в § 1). Н пример, дерево (17) отвечает современной точке зрен на сказуемое как на корень (или, как говорят некот рые лингвисты, «вершину») предложения. Более трад ционной концепции отвечало бы дерево с корнем в по лежащем:

(18) 

При описании предложений с помощью деревьев по чинения мы можем, как и при использовании систем с ставляющих, столкнуться с явлением синтаксическ омонимии. Так, следующие два дерева одного и того предложения:

(19) 

и

(20) 

соответствуют двум его разным смыслам (ср. стр. 285)

Аналогично для предложений (11) и (12) «естестве ные» деревья будут иметь вид

(А) 

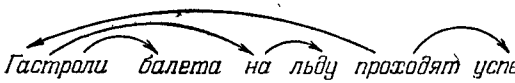
и

(Б′) 

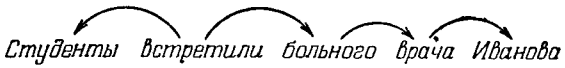
однако, не располагая полным набором необходимых внеязыковых фактов (ср. стр. 286), мы должны бы были допустить также деревья

(А′) 

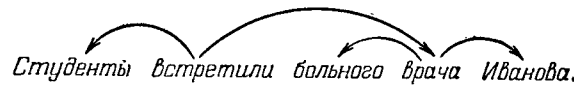
и

(Б) 

При этом возможны случаи, когда бесспорно синтаксически омонимичное — с содержательной точки зрения — предложение допускает различные «естественные» описания с помощью деревьев подчинения, но только одно с помощью системы составляющих, и наоборот. Например, двум различным смыслам предложения, допускающего лишь одну «естественную» систему составляющих (16), отвечают соответственно деревья

(21) 

и

(22) 

В то же время предложение

(23) *Он только подписывает эти бумаги*

допускает по меньшей мере два смысла («эти бумаги он только подписывает, а не составляет их сам» и «он ничего вообще не делает, кроме как подписывает эти бумаги»), и этим смыслам отвечают разные системы составляющих

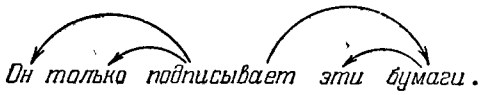
(24) *Он ((только подписывает) (эти бумаги))*

и

(25) *Он (только (подписывает (эти бумаги)))*,

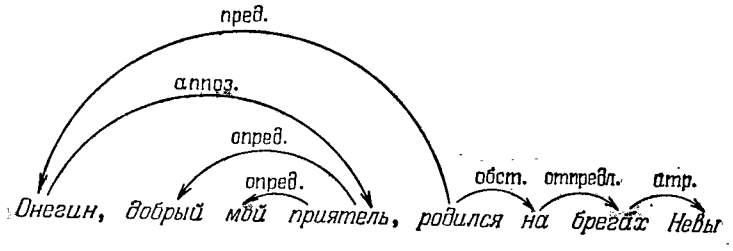
но «естественное» дерево при обоих смыслах одно и то же:

(26)

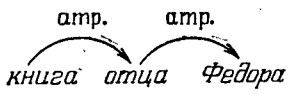


Для анализа предложений естественного языка часто используются так называемые размеченные деревья подчинения. Размеченное дерево подчинения для цепочки x — это четверка $\langle X; \rightarrow, Z, \psi\rangle$, где $\langle X; \rightarrow\rangle$ — дерево подчинения для x, Z — конечное множество (его элементы называются метками) и ψ — отображение множества дуг дерева $\langle X; \rightarrow\rangle$ в Z.

Пусть, например, Z содержит символы пред., аппоз., опред., атр., обст., отпредл., интерпретируемые соответственно как отношения между: сказуемым и подлежащим, существительным и приложением к нему, существительным и его согласованным определением, существительным и его несогласованным определением, глаголом и его обстоятельством, предлогом и управляемым им существительным (каждый раз в указанном порядке). Тогда дерево (17) естественно будет разметить так:



Возможны, вообще говоря, различные разметки одного и того же дерева, отвечающие разным смыслам. Так, конструкция *книга отца Федора* имеет не менее двух смыслов — при одном речь идет об отце некоего Федора, при другом о некоем отце Федоре. Этим смыслам можно сопоставить следующие две разметки одного и того же дерева:



Проективность и слабая проективность. В классе всевозможных деревьев подчинения можно выделить подкласс, который содержит подавляющее большинство «естественных» деревьев для предложений реальных языков и, по-видимому, практически все «естественные» деревья выражений наиболее важных искусственных формализованных языков. Это класс так называемых проективных деревьев.

Именно, дерево подчинения $\langle X; \rightarrow\rangle$ для цепочки x (а также соответствующее отношение подчинения $\rightarrow$) называется проективным, если для любых трех точек α, β, γ цепочки x из того, что $\alpha \rightarrow \beta$ и γ лежит между α и β, следует, что γ зависит от α.

Будем называть подстрелочным всякий интервал (α, β) цепочки x, для которого $\alpha \rightarrow \beta$ или $\beta \rightarrow \alpha$. Подстрелочный интервал (α, β) будет называться правильным, если все его точки зависят от α или от β (в частности, пустой интервал правилен). Таким образом, проективность означает правильность всех подстрелочных интервалов.

Деревья рис. 15, *а*), *б*), *в*) проективны, деревья рис. 15, *г*), *д*) не проективны. Все деревья предложений на стр. 295—298 проективны.

Еще один важный класс деревьев подчинения (являющийся, как мы скоро увидим, расширением предыдущего) — класс слабо проективных деревьев. Именно, дерево подчинения $\langle X; \rightarrow\rangle$ для цепочки x (и отношение $\rightarrow$) называется слабо проективным, если для любых четырех точек α, β, γ, δ цепочки x из $\alpha \rightarrow \beta$

и $\gamma \to \delta$ следует, что пары α, β и γ, δ не разделяют др друга.

При нашем способе графического изображения д ревьев подчинения слабая проективность равносиль возможности провести все стрелки так, чтобы никак две из них не пересекались *).

Деревья рис. 15, *а*) — *г*) слабо проективны, дере рис. 15, *д*) не является слабо проективным. Все прив денные на стр. 295—298 деревья предложений слаб проективны.

Укажем простые необходимые и достаточные услови проективности и слабой проективности.

Будем называть циклическим отрезком ц почки x всякое множество точек этой цепочки, котор является либо отрезком, либо объединением двух отре ков, прилегающих к концам цепочки (иначе говоря, л бое множество, которое при «склеивании» начала конца цепочки — цепочка при таком склеивании оказ вается «записанной на окружности» — переходит в о резок).

Теорема П1. 1. *Дерево подчинения для цепочки тогда и только тогда проективно, соответственно сла проективно, когда группы зависимости всех узлов этог дерева являются отрезками, соответственно циклическ ми отрезками, цепочки x.*

Доказательство. а) Если дерево подчинения н является слабо проективным, то существуют четыр точки α, β, γ, δ такие, что $\alpha \to \beta$, $\gamma \to \delta$ и пары α, β γ, δ разделяют друг друга. Из точек α, γ хотя бы одн не зависит от другой; пусть для определенности γ не з висит от α. Тогда и δ не зависит от α, так что и внутр интервала, ограниченного точками α и β, и вне е имеются точки, не принадлежащие группе зависимост точки α. Но если множество M точек цепочки x содер жит такие две точки, что и внутри и вне ограниченно ими интервала имеются точки, не входящие в M, то не может быть циклическим отрезком.

Если дерево не проективно, то существуют три то ки α, β, γ такие, что $\alpha \to \beta$, γ лежит между α и β и

*) Доказательство этого геометрического факта выходит рамки настоящего изложения.

не зависит от α. Но тогда группа зависимости точки α не может, очевидно, быть отрезком.

б) Пусть дерево подчинения $\langle X; \to\rangle$ цепочки x проективно (слабо проективно). Покажем, что группа зависимости произвольной точки α цепочки x является отрезком (циклическим отрезком). Доказательство проведем индукцией по рангу α.

Если α — узел ранга 0 (т. е. висячий), то группа зависимости для α состоит из одной точки α, стало быть, она является отрезком и тем более циклическим отрезком. Пусть утверждение доказано для всех точек ранга, не большего n ($n \geqslant 0$), α — точка ранга $n+1$ и K_α — ее группа зависимости. Тогда $K_\alpha = \{\alpha\} \cup K_{\beta_1} \cup \ldots \cup K_{\beta_p}$, где $\beta_1, \ldots, \beta_p$ — все точки, подчиненные α, и $K_{\beta_1}, \ldots, K_{\beta_p}$ — их группы зависимости.

Если дерево проективно, то множества $K_{\beta_1}, \ldots, K_{\beta_p}$ — отрезки, и если бы при этом K_α не было отрезком, то нашлась бы точка γ, не принадлежащая K_α и в то же время лежащая между α и некоторым K_{β_i} и, следовательно, между α и β_i; а это противоречит проективности.

Пусть дерево слабо проективно. Покажем прежде всего, что в этом случае для каждого β_i либо все точки, лежащие между α и β_i, либо все точки, не лежащие между α и β_i, принадлежат K_α. Действительно, пусть, например, как между, так и не между α и β_1 имеются точки, не входящие в K_α. Обозначим через ε корень дерева $\langle X; \to\rangle$. Очевидно, $\varepsilon \notin K_\alpha$. Пусть для определенности ε лежит между α и β_1, и пусть λ — произвольная точка, не принадлежащая K_α и не лежащая между α и β_1. Рассмотрим путь $\varepsilon = \mu_1, \ldots, \mu_s = \lambda$ из ε в λ; ни одна из его точек не может совпасть с α или β_1. Если μ_i — последняя точка этого пути, лежащая между α и β_1, то пары точек α, β_1 и μ_i, μ_{i+1} нарушают условие слабой проективности.

Допустим теперь, что K_α не является циклическим отрезком. Тогда существуют точки γ, $\delta \in K_\alpha$ такие, что как внутри, так и вне интервала (γ, δ) имеются точки, не принадлежащие K_α. Пусть λ, $\mu \notin K_\alpha$, $\lambda \in (\gamma, \delta)$, $\mu \notin (\gamma, \delta)$ (рис. 16). Тогда как внутри, так и вне

интервала Δ, ограниченного точками λ, μ, имеются точки множества K_α. Это, однако, невозможно. В самом деле: пусть сначала $\alpha \notin \Delta$; тогда по ранее доказанному все точки $\beta_1, \ldots, \beta_p$ тоже не принадлежат Δ, а отсюда следует, что каждое из множеств $K_{\beta_1}, \ldots, K_{\beta_p}$ целиком лежит вне Δ (так как эти множества — циклические отрезки); совершенно аналогично для случая $\alpha \in \Delta$.

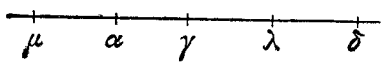


Рис. 16.

Следствие. Всякое проективное дерево подчинения слабо проективно.

Замечания. 1. В слабо проективном дереве подчинения каждый неправильный подстрелочный интервал содержит корень.

Действительно, если $\alpha \to \beta$ и γ — произвольная точка, принадлежащая подстрелочному интервалу Δ, ограниченному точками α, β, и не зависящая от α, и если при этом корень не принадлежит Δ, то пары α, β и δ_i, δ_{i+1}, где δ_i — последняя точка пути из корня в γ, не принадлежащая Δ, нарушают условие слабой проективности.

2. Слабо проективное дерево подчинения тогда и только тогда проективно, когда никакой подстрелочный интервал не содержит корня.

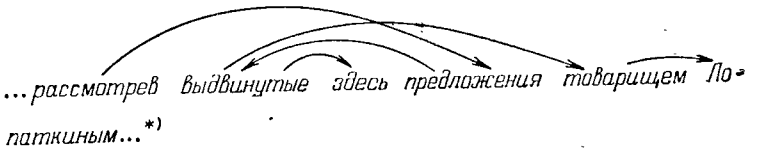
Это сразу следует из предыдущего замечания.

Следствие из теоремы П1.1 и замечание 2 дают возможность получить для проективности столь же наглядное представление, как для слабой проективности (при нашем способе графического изображения): проективность означает возможность провести все стрелки так, чтобы никакие две из них не пересекались и корень не лежал ни под одной из них.

Мы не можем сколько-нибудь подробно обсуждать здесь вопросы, связанные с лингвистической интерпретацией понятий проективности и слабой проективности, или разбирать различные случаи выполнения и невыполнения этих условий в естественных языках. Ограничимся следующими замечаниями:

1. Содержательный смысл условий проективности и слабой проективности может быть, как видно уже из самих определений и еще яснее из теоремы П1.1, охарактеризован приблизительно так: при выполнении этих условий слова, близкие синтаксически, близки и по положению в тексте. При этом проективность обеспечивает «более тесную» текстовую близость.

2. В так называемой научной и деловой прозе (по крайней мере русской) «естественные» деревья подчинения подавляющего большинства предложений слабо проективны и даже проективны. При этом едва ли не все непроективные деревья принадлежат тем предложениям, для которых дерево подчинения вообще не является достаточно естественным средством представления синтаксической структуры (см. ниже, стр. 309—310). За вычетом таких случаев непроективность в «деловом» тексте — верный признак недостаточной грамотности его сочинителя. Ср. хотя бы знаменитый оборот вида

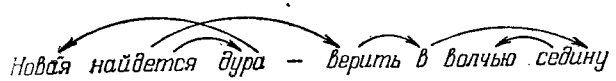


3. Напротив, в художественной литературе, особенно в поэзии, отклонения от слабой проективности и тем более от проективности вполне обычны и в тех случаях, когда дерево подчинения достаточно адекватно представляет синтаксическую структуру. Такие отклонения, как правило, значимы, т. е. служат задаче создания определенного художественного эффекта **); они могут, например, использоваться для подчеркивания особой важности каких-либо частей предложения, для придания предложению экспрессивности, живости, непринужденности, для создания нужной стилистической окраски

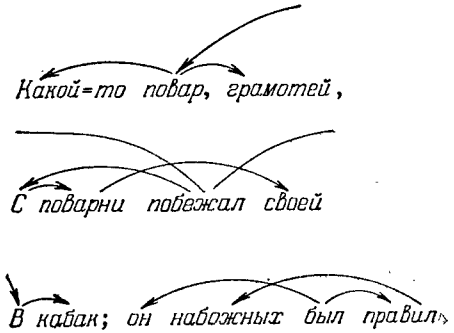
*) Пример из романа В. Дудинцева «Не хлебом единым». Эти слова вложены автором в уста одного из эпизодических персонажей — доктора наук Тепикина, и их оказывается достаточно для портрета малограмотного «научного» карьериста.

**) Как и другие отступления от «абсолютной правильности», встречающиеся в художественной литературе.

и т. п. Ср. выделение слова *новая* во фразе из стихотворения М. И. Цветаевой



или впечатление непринужденности, «разговорности», создаваемое непроективностями в басне И. А. Крылова:



§ ПI.3. Связь между системами составляющих и деревьями подчинения

Пусть C — система составляющих цепочки x и $\langle X; \rightarrow\rangle$ — дерево подчинения для той же цепочки. Будем говорить, что система C и дерево $\langle X; \rightarrow\rangle$ (или система C и отношение $\rightarrow$) согласованы, если: 1) группы зависимости всех узлов дерева $\langle X; \rightarrow\rangle$ являются составляющими системы C; 2) каждая составляющая системы C является группой зависимости или усеченной группой зависимости некоторого узла дерева $\langle X; \rightarrow\rangle$.

Пример. Дерево (17) согласовано с системой (4).

Для дальнейшего будет полезно следующее очевидное замечание. Если система составляющих C согласована с деревом $\langle X; \rightarrow\rangle$ и $A \in C$ — группа зависимости или усеченная группа зависимости точки α, то среди составляющих, непосредственно вложенных в A, та, которая содержит точку α, является для нее усеченной группой зависимости, а остальные являются группами зависимости некоторых точек, подчиненных α.

Ввиду теоремы ПI.1 всякое дерево подчинения, согласованное с какой-либо системой составляющих, проективно. С другой стороны, для всякого проективного отношения подчинения существуют согласованные с ним системы составляющих; одной из них будет множество C_0, состоящее из всех точек *) цепочки и их групп зависимости. (Всякая точка совпадает со своей группой зависимости, если она является висячим узлом дерева, и с одной из своих усеченных групп зависимости в противном случае; две группы зависимости пересекаются лишь тогда, когда одна из них содержится в другой.) В общем случае согласованная с деревом подчинения система составляющих не единственна; строение множества таких систем рассматривается в упражнениях ПI.7—ПI.10.

Перейдем к обратному вопросу — о деревьях подчинения, согласованных с заданной системой составляющих. Здесь оказывается удобнее исходить из понятия иерархизованной системы составляющих.

Введем следующее определение.

Иерархизованная система составляющих $\langle C, C'\rangle$ цепочки x и дерево подчинения $\langle X; \rightarrow\rangle$ для той же цепочки (или $\langle C, C'\rangle$ и отношение $\rightarrow$) связаны, если: 1) множество групп зависимости узлов дерева совпадает с $C \cup C'$; 2) все элементы C' являются усеченными группами зависимости узлов дерева.

Очевидно, всякое дерево, связанное с $\langle C, C'\rangle$, согласовано с C.

Пример. Дерево (17) связано с иерархизованной системой (15).

Лемма ПI.2. а) *Если C — система составляющих для цепочки x, $A \in C$ и x_A — подцепочка x, соответствующая A, то множество $C_A = \{B \mid B \in C \,\&\, B \subseteq A\}$ есть система составляющих для x_A.*

б) *Если в условиях пункта* а) *C' есть иерархизация системы C, то $C'_A = C' \cap C_A$ есть иерархизация системы C_A.*

*) Точнее — одноточечных подмножеств.

в) *Если в условиях пункта* а) $\rightarrow$ *есть отношение подчинения для* x, *согласованное с* C, *то отношение* $\rightarrow$ *индуцируемое отношением* $\rightarrow$ *на* x_A, *есть отношение подчинения для* x_A, *согласованное с* C_A.

г) *Если в условиях пунктов* а), б), в) *отношение* $\rightarrow$ *связано с иерархизованной системой* $\langle C, C'\rangle$, *то* $\rightarrow$ *связано с* $\langle C_A, C'_A\rangle$.

Лемма П.3. *Пусть* $\langle C, C'\rangle$ — *иерархизованная система составляющих и* $B_0, B_1, \ldots, B_p$ — *все составляющие высоты* 1, *причем* $B_0 \in C'$, $B_1, \ldots, B_p \notin C'$. *Если при этом* $\langle X; \rightarrow\rangle$ — *дерево подчинения, связанное с* $\langle C, C'\rangle$, *то* $\langle X; \rightarrow\rangle$ *получается из объединения деревьев* $\langle B_0; \rightarrow_{B_0}\rangle$, $\langle B_1; \rightarrow_{B_1}\rangle, \ldots, \langle B_p; \rightarrow_{B_p}\rangle$ *добавлением дуг* $(\alpha_0, \alpha_1), \ldots, (\alpha_0, \alpha$ *где* $\alpha_0, \alpha_1, \ldots, \alpha_p$ — *корни деревьев* $\langle B_0; \rightarrow_{B_0}\rangle, \langle B_1; \rightarrow_{B_1}\rangle, \ldots, \langle B_p; \rightarrow_{B_p}\rangle$ *соответственно.*

Доказательства лемм П.2 и П.3 очевидны.

Теорема П.2. а) *Для всякой иерархизованной системы составляющих существует единственное связанное с ней дерево подчинения.*

б) *Для всякого дерева подчинения, согласованного с системой составляющих* C, *существует единственная иерархизация* C' *системы* C *такая, что иерархизованная система* $\langle C, C'\rangle$ *связана с данным деревом.*

Доказательство. а) Пусть $\langle C, C'\rangle$ — иерархизованная система составляющих для цепочки x и X множество всех точек цепочки x. Для каждой неточечной составляющей A будем обозначать через $a(A)$ непосредственно вложенную в A главную составляющую и через $b_1(A), \ldots, b_{p_A}(A)$ $(p_A \geqslant 1)$ — непосредственно вложенные в A не главные составляющие. Для каждой составляющей $A \in B$ определим ее главную точку $\alpha(A$ следующим образом. Если ранг A равен нулю, то единственная точка A будет главной. Если главные точки определены для всех составляющих ранга, меньшего и A — составляющая ранга n, то главной точкой A будет главная точка $a(A)$.

Заметим, что каждая точка α цепочки x будет главной для одной и только одной не главной составляющей. Действительно, если A — точечная составляющая, стоящая из единственной точки α, $A_0, A_1, \ldots, A_k = A$ путь в дереве составляющих из его корня (полной составляющей) в A и последняя на этом пути не главная составляющая есть A_{i_0} $(0 \leqslant i_0 \leqslant k)$, то $A_{i_0}, A_{i_0+1}, \ldots, A_k$ — в точности все составляющие системы C, для которых α служит главной точкой; поэтому составляющая A_{i_0} будет искомой.

Положим теперь $\alpha \rightarrow \beta$, если α — главная точка какой-либо составляющей A и β — главная точка одной из составляющих $b_1(A), \ldots, b_{p_A}(A)$. Из только что сделанного замечания немедленно вытекает, что: а) ни в одну точку цепочки x не входит более одной дуги; б) во всякую точку x, кроме главной точки полной составляющей, входит дуга; в) в графе $\langle X; \rightarrow\rangle$ нет замкнутых путей ненулевой длины. Итак, $\langle X; \rightarrow\rangle$ — дерево подчинения для x (с корнем в главной точке полной составляющей). Покажем, что это дерево связано с $\langle C, C'\rangle$. Для этого достаточно установить следующий факт: для любого r, $0 \leqslant r \leqslant R$, где R — высота системы C, каждая не главная составляющая ранга r является группой зависимости своей главной точки, а каждая главная составляющая ранга r — усеченной группой зависимости своей главной точки. (Действительно, отсюда, поскольку каждая точка цепочки главная для некоторой не главной составляющей, будет следовать, что $C - C'$ совпадает с множеством групп зависимости.) Докажем это утверждение сначала для $r = 0$. Пусть A — точечная составляющая; тогда $A = \{\alpha\}$ и α — главная точка A. Если при этом A — не главная составляющая, то ни для какой составляющей, отличной от A, точка α не будет главной, так что (по определению отношения $\rightarrow$) эта точка окажется в дереве $\langle X; \rightarrow\rangle$ висячим узлом и ее группа зависимости будет состоять лишь из нее самой. Если же A — главная составляющая, то непременно найдутся точки, подчиненные α (такими будут хотя бы главные точки других составляющих, непосредственно вложенных в ту же составляющую, что и A), так что A не может быть группой зависимости точки α и поэтому является ее усеченной группой зависимости. Пусть теперь утверждение доказано для всех r, меньших r_0 $(r_0 \geqslant 1)$, A — составляющая ранга r_0 и α — ее главная точка. Если $B_0, B_1, \ldots, B_p$ — все непосредственно вложенные в A составляющие и B_0 является главной, то по индуктивному предположению B_0 есть усеченная групп

зависимости точки α, а $B_1, \ldots, B_p$ — группы зависимости своих главных точек. Таким образом, A есть объединение усеченной группы зависимости точки α с группами зависимости некоторых точек, подчиненных α, — а такое множество есть либо группа зависимости, либо усеченная группа зависимости для α. Если при этом A — главная составляющая, то существуют точки, подчиненные α и лежащие вне A, если же A — не главная составляющая, то таких точек нет (так как α не является тогда главной точкой ни для какой составляющей, не содержащейся в A). Поэтому в первом случае A — усеченная группа зависимости для α, во втором — группа зависимости.

Единственность дерева, связанного с иерархизованной системой составляющих $\langle C, C'\rangle$, мы докажем индукцией по высоте r системы C. При $r = 0$ утверждение тривиально. Пусть оно доказано для всех $r < r_0$ и C имеет высоту r_0. Обозначим через $B_0, B_1, \ldots, B_p$ составляющие глубины 1. Пусть $\langle X, \rightarrow_1\rangle$ и $\langle X, \rightarrow_2\rangle$ — два дерева подчинения, связанных с $\langle C, C'\rangle$. По лемме II.2 для каждого $j = 0, \ldots, p$ отношение $\rightarrow_{iB_j} (i = 1, 2)$, индуцируемое отношением $\rightarrow_i$ на B_j, связано иерархизованной системой $\langle C_{B_j}, C'_{B_j}\rangle$. Отсюда по индуктивному предположению следует, что для каждого $j = 0, \ldots, p$ деревья $\langle B_j, \rightarrow_{1B_j}\rangle$ и $\langle B_j, \rightarrow_{2B_j}\rangle$ совпадают. А это в силу леммы II. 3 влечет совпадение деревьев $\langle X, \rightarrow_1\rangle$ и $\langle X, \rightarrow_2\rangle$.

б) Пусть C — система составляющих, согласованная с некоторым деревом подчинения. Для каждой неточечной составляющей $A \in C$, являющейся группой зависимости или усеченной группой зависимости некоторой точки α, объявим главной ту из непосредственно вложенных в A составляющих, которая содержит α. При такой иерархизации в силу замечания, сделанного после определения согласованности (стр. 304—305), все главные составляющие будут усеченными группами зависимости, а не главные—группами зависимости; но все группы зависимости по условию являются составляющими так что множество не главных составляющих совпадет множеством групп зависимости. Единственность нужной иерархизации сразу следует из упомянутого замечания

Итак, для каждой системы составляющих имеется взаимно однозначное соответствие между ее иерархизациями и согласованными с ней деревьями подчинения.

Замечания. 1) В рассуждениях этого параграфа проективность нужна только для того, чтобы все группы зависимости были отрезками. Если изменить определение системы составляющих, допустив в качестве ее элементов циклические отрезки или даже произвольные множества точек цепочки (конечно, при сохранении пунктов I, II определения на стр. 282), то все наши рассуждения будут годиться для любых слабо проективных, соответственно вообще для любых, деревьев подчинения.

2) Системы составляющих и деревья подчинения характеризуют синтаксическую структуру предложения в разных аспектах. С помощью первых описываются в явном виде словосочетания, но игнорируется ориентация связей (т. е. не различаются «главные» и «зависимые» элементы); вторые дают возможность рассматривать направленные связи, но только между отдельными словами. Между тем во многих случаях было бы естественнее описывать структуру предложения, рассматривая направленные связи не только между словами, но и между словосочетаниями. В какой-то мере здесь могут помочь иерархизованные системы составляющих; однако из теоремы II. 2 видно, что в некотором смысле их возможности не шире, чем у проективных отношений подчинения: с помощью иерархизованных систем составляющих удается описать направленные связи словосочетаний только тогда, когда в каждом словосочетании достаточно естественным образом выделяется «главное слово» так, что ему можно передать пассивную связь словосочетания (т. е. вместо того, чтобы подчинить какому-то элементу все словосочетание, подчинить тому же элементу главное слово); это означает, что система направленных связей между словосочетаниями, т. е. иерархизованная система составляющих, должна быть содержательно равносильна системе связей между их главными словами, а последняя есть не что иное, как дерево подчинения, связанное с исходной иерархизованной системой. Однако бывают словосочетания, в которых либо вообще не удается выделить естественным образом главное

слово, либо «подвешивание» словосочетания за главное слово не оправдано. Например, в конструкциях с однородными членами нет оснований считать то или иное слово главным; в сложных формах глагола (типа *буду читать*) естественно считать все их внешние связи относящимися только ко всему словосочетанию как целому, поскольку эти формы синтаксически ведут себя так же, как простые. Сходным образом можно трактовать сочетания модальных глаголов и иных модальных слов с инфинитивами, конструкции, содержащие определенные стандартные лексические функции [Жолковский — Мельчук 1967] (например, Oper_i, Incep), и некоторые другие типы конструкций. Именно в таких конструкциях при попытках описать их с помощью деревьев подчинения часто возникает непроективность:

Эти и другие случаи содержательной неадекватности деревьев подчинения и систем составляющих привели к попыткам разработать другие, более сложные способы описания синтаксической структуры предложения. Один из таких способов, являющийся одновременным обобщением систем составляющих и деревьев подчинения, предложен в [Гладкий 1969, 1971]. Останавливаться на этих вопросах подробнее мы не будем, поскольку они не имеют прямого отношения к предмету книги.

Упражнения

П1. 1. Показать, что если в цепочке расставлены каким-либо образом левые и правые скобки, то существует не более одного взаимно однозначного отображения множества левых скобок на множество правых, при котором каждая левая скобка расположена левее своего образа и отрезки, ограниченные левыми скобками и соответствующими им правыми, образуют систему составляющих данной цепочки.

*) *Дать* = Oper_1(*отзыв*).

П1. 2. Пусть s_n — число систем составляющих одной цепочки длины $n \geqslant 2$ и s_{ni} $(1 \leqslant i \leqslant n)$ — число тех из этих систем, для которых крайняя справа точка цепочки служит концом i различных составляющих. Показать, что

а) $s_{n+1} = \sum_{i=2}^{n} (2i-1) \cdot s_{ni}$;

б) $s_{n+1,\,i} = s_{n,\,i-1} + 2 \sum_{j=i}^{n} s_{nj}$.

П1. 3. Доказать, что граф $\langle C; \supset \supset \rangle$ (стр. 287) является деревом.

П1. 4. Пусть b_n — число бинарных систем составляющих одной цепочки длины $n \geqslant 2$.

а) Указать рекуррентную формулу, выражающую b_{n+1} через $b_2, \ldots, b_n$.

б) Показать, что для каждой системы составляющих C цепочки x $(|x| \geqslant 2)$ существует точно $\prod_{A \in C} b_{l(A)}$ содержащих ее бинарных систем составляющих той же цепочки, где $l(A)$ — ширина куста составляющей A в системе C (b_0 считается равным единице).

П1.5. Пусть C — система составляющих ранга r и ширины l. Обозначим через I_C число различных иерархизаций системы C и через I_{rl} — число $i \cdot i^i \cdot (i^i)^i \cdot \ldots \cdot (\ldots (i^i)^i \ldots)^i$, где последний сомножитель содержит $r-1$ возведений в степень. Показать, что:

а) если ширина куста каждой неточечной составляющей системы C равна l, то $I_C = I_{rl}$;

б) в общем случае $I_{r2} \leqslant I_C \leqslant I_{rl}$.

П1. 6. Построить «естественное» дерево подчинения для формулы (5) (стр. 284).

П1.7. Показать, что всякая небинарная система составляющих, согласованная с некоторым деревом подчинения, содержится в некоторой другой системе составляющих, согласованной с тем же деревом.

Таким образом, в множестве всех систем составляющих, согласованных с данными деревом подчинения, система C_0 (стр. 305) является наименьшим элементом (относительно теоретико-множественного включения), а бинарные системы — максимальными элементами.

П1. 8. Показать, что согласованная с деревом $\langle X; \rightarrow \rangle$ система составляющих тогда и только тогда единственна, когда $\langle X; \rightarrow \rangle$ — дерево без ветвления (т. е. каждая его вершина имеет не более одной подчиненной).

П1. 9. Пусть S — множество всех систем составляющих, согласованных с деревом $\langle X; \rightarrow \rangle$, и для $C_1, C_2 \in S$ запись $C_1 \Rightarrow C_2$ означает, что $C_1 \subset C_2$ и не существует системы $C' \in S$ такой, что $C_1 \subset C' \subset C_2$. Показать, что длина любого максимального пути в графе $\langle S; \Rightarrow \rangle$ (такой путь, вообще говоря, не один) равна сумме уменьшенных на единицу ширин кустов невисячих узлов дерева $\langle X; \rightarrow \rangle$.

ПI. 10. Показать, что бинарная система составляющих, согласованная с деревом $\langle X; \rightarrow \rangle$, тогда и только тогда единственна, когда для каждого узла этого дерева все подчиненные ему узлы расположены по одну сторону от него.

ПI. 11. Показать, что высота дерева подчинения, согласованного с системой составляющих, не превосходит высоты этой системы.

ПI. 12. Показать, что система составляющих C и дерево подчинения $\langle X; \rightarrow \rangle$ тогда и только тогда согласованы, когда: а) группы зависимости всех узлов дерева $\langle X; \rightarrow \rangle$ являются составляющими системы C; б) каждая составляющая системы C совпадает с множеством узлов некоторого поддерева дерева $\langle X; \rightarrow \rangle$.

ПI. 13. Показать, что объединение множества подформул (в обычном смысле) произвольной формулы F исчисления предикатов (как в «скобочной», так и в бесскобочной записи) с множеством вхождений в F элементарных символов является системой составляющих для F.

ПI. 14. Назовем системы составляющих C и C_1 цепочек x и x_1 п о д о б н ы м и, если изображающие их скобочные последовательности совпадают.

а) Сформулировать определение подобия, не упоминая о скобочных или иных изображениях.

б) Показать, что если F — формула исчисления предикатов в «скобочной» записи и F_1 — та же формула в бесскобочной записи, то описанные в упражнении ПI. 13 системы составляющих цепочек F и F_1 подобны.

ПI. 15. Показать, что если дерево подчинения $\langle X; \rightarrow \rangle$ связано с иерархизованной системой составляющих (C, C') цепочки x, α — точка цепочки x и $B_0 \supset B_1 \supset \ldots \supset B_t = \{\alpha\}$ — все составляющие, для которых α служит главной точкой (так что B_0 — группа зависимости узла α), то ширина куста узла α в дереве $\langle X; \rightarrow \rangle$ равна $(l_0 - 1) + \ldots + (l_{t-1} - 1)$, где l_i — ширина куста составляющей B_i.

З а м е ч а н и е. Если s_0 — степень составляющей B_0, то $s_0 \leqslant t \leqslant (l_0 - 1) + \ldots + (l_{t-1} - 1)$, так что ширина куста α не может быть меньше s_0. Поэтому степень системы C не превосходит ширины дерева $(X; \rightarrow)$.

ПI. 16. Построить «естественные» системы составляющих для предложений:

а) *Века пожирая, стояли шпалеры бессониц в горячечном гаме рубанков*;

б) *По-прежнему, схлынувши с лиц очевидцев, безумствует быль, притворяясь незнающей*;

в) *...впивая впотьмах это благо, бежала на чашечку с чашечки грозой одуренная влага*;

г) *Я креплюсь на пере у творца крупной каплей густого свинца.*

Для построенных систем вычислить значения $y^{\text{Л}}$, $y^{\text{П}}$, ф. Указать для них «естественные» иерархизации.

ПI. 17. Построить «естественные» деревья подчинения для предложений из упражнения ПI. 16. Проверить эти дервья на связанность с построенными при выполнении упражнения ПI. 16 иерархизованными системами составляющих.

ПI. 18. Иерархизовать каким-либо способом каждую из систем составляющих (1), (2) (стр. 283). Построить связанные с полученными иерархизациями деревья подчинения.

ПI. 19. Для каждого из деревьев подчинения рис. 15, *а*), *б*), *в*) построить все связанные с ним иерархизованные системы составляющих.

ПI. 20. Найти не менее 32 правильных анализов фразы: *Назначение дежурным инструктором учителя Иванова вместо Петрова подтверждено Сидоренко.*

Каким из этих анализов отвечают различные деревья подчинения и каким — одинаковые? Показать, что с помощью размеченных деревьев подчинения могут быть различены все 32 анализа.

ПРИЛОЖЕНИЕ II

ЗАМЕЩАЕМОСТЬ

Наряду с формальными грамматиками, представляющими собой «собственно модели языка», или, по терминологии Г. С. Цейтина [Цейтин 1959], «синтезирующие модели», существуют так называемые «анализирующие модели языка», или «модели лингвистического исследования». В синтезирующих моделях задаются некоторые сведения о строении механизма языка (например, в виде правил преобразования), и работа модели состоит в выполнении преобразований языковых выражений или порождении некоторых «текстов». В анализирующих моделях, напротив, считаются заданными некоторые совокупности «текстов», и задача состоит в извлечении из них той или иной информации о строении языка; таким образом, мы можем говорить в этом случае о «моделировании деятельности лингвиста». (Подробнее см. [Гладкий — Мельчук 1969, стр. 162—164].) Наиболее разработанный класс анализирующих моделей составляют те, которые основаны на понятии замещаемости цепочек; им и посвящено настоящее приложение.

Предварительно необходимо сделать следующее замечание. При лингвистической интерпретации анализирующих моделей уже не безразлично, что́ понимать под элементарными символами — сегменты или словоформы (ср. введение, стр. 15—16). При втором понимании в число исходных данных включаются грамматические значения, т. е. считаются заранее заданными такие понятия, как часть речи, падеж, род и т. п. Но это обесценивает модель — ведь именно построение формальных аналогов подобных понятий является ее целью, исходные же данные должны быть логически более простыми.

С другой стороны, опираясь лишь на понятие сегмента, относящееся целиком к внешней стороне языка, можно рассчитывать получить лишь довольно «бедные» модели (ср. ниже, стр. 327). Поэтому представляется целесообразным включить в число исходных данных лексические значения слов. Для этого мы введем понятие лексически значимого сегмента (ЛЗ-сегмента), промежуточное между понятиями сегмента и словоформы; именно, ЛЗ-сегмент — это сегмент, снабженный лексическим значением (но не грамматическими). Например, в предложении *Он говорит только о физике, читает книги только по физике, и о нем говорят только как о физике* первые два вхождения сегмента *физике* являются вхождениями одного и того же ЛЗ-сегмента — но разных словоформ! — а третье отвечает другому ЛЗ-сегменту. В дальнейшем мы будем понимать элементарные символы как ЛЗ-сегменты. Впрочем, до некоторого момента будет допустимо также и понимание их как сегментов; место, начиная с которого введение лексических значений существенно, будет явно указано (стр. 327—328).

§ ПII.1. Свободные полугруппы

Этот параграф вспомогательный; в нем излагаются некоторые простые алгебраические понятия, которые понадобятся нам в дальнейшем.

Множество с определенной на нем ассоциативной бинарной операцией называется полугруппой. Для обозначения полугрупповой операции обычно используется мультипликативная запись: результат операции над элементами x и y (в этом порядке) обозначается xy. Элемент e полугруппы S называется единицей, если для любого $x \in S$ имеет место $xe = ex = x$. Легко видеть, что если полугруппа содержит единицу (что не обязательно), то только одну. Пусть S — полугруппа и $M \subseteq S$; если S обладает единицей e, обозначим через $S^{\wedge}$ множество $S - \{e\}$; в противном случае положим $S^{\wedge} = S$. Если замыкание множества M относительно полугрупповой операции совпадает с $S^{\wedge}$, говорят, что M есть система образующих для S.

Для произвольного алфавита (словаря) V множество V^* является, очевидно, полугруппой относительно операции конкатенации с системой образующих V; цепочка Λ является единицей этой полугруппы. Такая полугруппа называется свободной.

Отображение φ полугруппы S в полугруппу S' называется гомоморфным, если $\forall x, y \in S\,[\varphi(xy) = \varphi(x)\varphi(y)]$. (Для случая, когда S и S' — свободные полугруппы, понятие гомоморфного отображения фактически было уже определено в § 1.1.) Взаимно однозначное гомоморфное отображение называется изоморфным. Если φ — изоморфное отображение полугруппы S на полугруппу S', то обратное отображение φ^{-1}: $S' \to S$ также является изоморфным. В этом случае говорят, что S и S' изоморфны между собой.

Отношение эквивалентности R на полугруппе S называется конгруэнцией, если имеет место $\forall x, y, x', y' \in S\,[xRx' \,\&\, yRy' \supset xyRx'y']$. Если R — конгруэнция на полугруппе S, то любым двум классам $X, Y \in S/R$ однозначно отвечает класс, который мы будем обозначать XY, состоящий из всевозможных произведений xy, где $x \in X$, $y \in Y$. Легко убедиться, что определенная таким образом на S/R бинарная операция ассоциативна, т. е. S/R является относительно этой операции полугруппой; эта полугруппа называется факторполугруппой полугруппы S по конгруэнции R. Отображение полугруппы S на полугруппу S/R, сопоставляющее каждому элементу S содержащий его класс, является, очевидно, гомоморфным — это так называемый естественный гомоморфизм.

Пусть V — алфавит и ρ — произвольное отношение эквивалентности на V. Определим отношение ρ^* на свободной полугруппе V^* следующим образом: $x\rho^* y$ означает, что либо $x = y = \Lambda$, либо $x = a_1 \ldots a_k$, $y = b_1 \ldots b_k$, где $a_i, b_i \in V$ и $a_i \rho b_i$ $(i = 1, \ldots, k)$. Ясно, что ρ^* есть конгруэнция. Естественный гомоморфизм полугруппы V^* на полугруппу V^*/ρ^* мы обозначим через φ.

Пусть теперь X — произвольный класс по конгруэнции ρ^* и x — произвольная цепочка, принадлежащая X. Если $\Lambda \notin X$, т. е. $x = a_1 \ldots a_k$, где $a_1, \ldots, a_k \in V$, $k \geqslant 1$, мы обозначим цепочку $\varphi(a_1) \ldots \varphi(a_k)$ (не зависящую, в силу определения конгруэнции ρ^*, от выбора цепочки $x \in X$) через $\psi(X)$. Если же $\Lambda \in X$ (тогда $X = \{\Lambda\}$), положим $\psi(X) = \Lambda$. Мы получили, таким образом, отображение полугруппы V^*/ρ^* в свободную полугруппу $(V/\rho)^*$ с системой образующих V/ρ. Из способа определения этого отображения ясно, что оно гомоморфно; очевидно также, что каждая цепочка из $(V/\rho)^*$ имеет при отображении ψ прообраз, и притом единственный. Таким образом, ψ есть изоморфное отображение V^*/ρ^* на $(V/\rho)^*$; следовательно, отображение $\theta = \varphi\psi$, сопоставляющее каждой непустой цепочке $a_1 \ldots a_k$, где $a_1, \ldots, a_k \in V$, цепочку $\varphi(a_1) \ldots \varphi(a_k)$ и единице полугруппы V^* — единицу полугруппы $(V/\rho)^*$, есть гомоморфное отображение V^* на $(V/\rho)^*$; мы будем называть его алфавитным гомоморфизмом.

Если $\mathscr{R}_1$ и $\mathscr{R}_2$ — два отношения эквивалентности на произвольном множестве M такие, что каждый $\mathscr{R}_1$-класс содержится в некотором $\mathscr{R}_2$-классе, мы будем называть $\mathscr{R}_2$ укрупнением $\mathscr{R}_1$. Пусть ρ_1 — отношение эквивалентности на словаре V и ρ_2 — укрупнение ρ_1; тогда, как легко убедиться, ρ_2^* будет укрупнением ρ_1^*. Определим на V/ρ_1 отношение эквивалентности ρ_{12}, полагая $A\rho_{12}B$ тогда и только тогда, когда A и B — ρ_1-классы, содержащиеся в одном и том же ρ_2-классе. Обозначим через θ_1, θ_2 и θ_{12} алфавитные гомоморфизмы V^* на $(V/\rho_1)^*$, V^* на $(V/\rho_2)^*$ и $(V/\rho_1)^*$ на $((V/\rho_1)/\rho_{12})^*$ соответственно. Определим далее гомоморфное отображение ζ_{12}: $((V/\rho_1)/\rho_{12})^* \to (V/\rho_2)^*$, полагая $\zeta_{12}(\alpha)$ для каждого ρ_{12}-класса α равным объединению всех ρ_1-классов, принадлежащих α. Поскольку таким образом между $(V/\rho_1)/\rho_2$ и V/ρ_{12} устанавливается взаимно однозначное соответствие, ζ_{12} есть изоморфное отображение $((V/\rho_1)/\rho_{12})^*$ на $(V/\rho_2)^*$. Поэтому, полагая $\eta_{12} = \theta_{12}\zeta_{12}$, получаем гомоморфное отображение $(V/\rho_1)^*$ на $(V/\rho_2)^*$; это отображение сопоставляет каждой цепочке $\alpha_1 \ldots \alpha_k$, где $\alpha_1, \ldots, \alpha_k \in V/\rho_1$, цепочку $\beta_1 \ldots \beta_k$, где $\beta_1, \ldots, \beta_k \in V/\rho_2$, такую, что $\alpha_1 \subseteq \beta_1, \ldots, \alpha_k \subseteq \beta_k$. Ясно также, что $\theta_1\eta_{12} = \theta_2$. Отображение η_{12} мы также будем называть алфавитным гомоморфизмом. (Понятие, рассматривавшееся до сих пор под этим названием, становится частным случаем только что введенного, если заменить V на $V/=$, где $=$ — отношение тождества.)

§ ПII.2. Замещаемость и взаимозамещаемость. Конфигурации

Пусть V — произвольный словарь, L — язык в словаре V и $x, y \in V^*$. Будем говорить, что цепочка замещаема на цепочку y относительно словаря V и языка L, и писать $x \underset{V,L}{\Rightarrow} y$, если $\forall u, v \in V^* [uxv \in L \supset uyv \in L]$ *). Если каждая из цепочек x, y замещама на другую относительно V и L, говорим, что x и взаимозамещаемы относительно V и L, пишем $x \underset{V,L}{\Leftrightarrow} y$. (Буквы V и L будут опускаться, если недоразумение невозможно.) Запись $\Phi_x(V, L)$ будет означать $\{y \mid x \underset{V,L}{\Rightarrow} y\}$, запись $\Phi(V, L)$ будет означать $\{(x, y) \mid x \underset{V,L}{\Rightarrow} y\}$ аналогично, с заменой $\Rightarrow$ на $\Leftrightarrow$, определим $\Psi_x(V, L)$ и $\Psi(V, L)$. При невозможности недоразумения буква в этих обозначениях будет опускаться.

Легко убедиться, что для произвольного языка в произвольном словаре V отношение $\underset{V,L}{\Leftrightarrow}$ является конгруэнцией на свободной полугруппе V^*.

Будем говорить, что отношение эквивалентности на свободной полугруппе V^* согласовано с языком $L \subseteq V^*$, если для любых двух цепочек $x, y \in V^*$ из $x \in L$ и $x \mathfrak{R} y$ следует $y \in L$.

Лемма ПII.1. *Отношение $\underset{V,L}{\Leftrightarrow}$ согласовано с языком L и является укрупнением любой конгруэнции на V согласованной с L.*

Доказательство. Согласованность $\underset{V,L}{\Leftrightarrow}$ с L непосредственно вытекает из определений. Пусть $\mathfrak{R}$ — конгруэнция на V^*, согласованная с L, $x, y \in V^*$ и $x \mathfrak{R} y$. Тогда для любых $z, u \in V^*$ будет $zxu \mathfrak{R} zyu$, и поэтому $zxu \in L$ влечет $zyu \in L$; таким образом, $x \underset{V,L}{\Rightarrow} y$. Аналогично имеем $y \underset{V,L}{\Rightarrow} x$. Итак, всякий $\mathfrak{R}$-класс содержится в некотором $\underset{V,L}{\Leftrightarrow}$-классе.

*) Ясно, что если $L \subseteq V_1^*$, $L \subseteq V_2^*$, $x, y \in V_1^*$, $x, y \in V_2^*$, то $x \underset{V_1,L}{\Rightarrow} y$ равносильно $x \underset{V_2,L}{\Rightarrow} y$. Однако обозначение $\underset{V,L}{\Rightarrow}$ удобно в случаях, когда отношение рассматривается «в целом», как множество пар.

Лемма ПII.1 позволяет утверждать, что среди всех конгруэнций на V^*, согласованных с L, отношение $\underset{V,L}{\Leftrightarrow}$ имеет наименьший индекс.

Лингвистический смысл понятия взаимозамещаемости состоит приблизительно в следующем. Если интерпретировать V как множество ЛЗ-сегментов *) некоторого естественного языка и L — как множество грамматически правильных фраз этого языка, то взаимозамещаемые цепочки будут представлять собой «синтаксически эквивалентные», т. е. выполняющие одни и те же синтаксические функции, словосочетания. Так, в русском языке цепочку *высокому дому* можно, видимо, считать взаимозамещаемой с цепочкой *умному совету, стене* — с *доске, исключительно важными* — с *важными*.

Пусть теперь $a \in V$, $x \in VV^+$ (так что $|x| > 1$) и $a \underset{L}{\Leftrightarrow} x$. Мы будем говорить в этом случае, что x есть конфигурация 1-го ранга языка L с результирующим a.

При указанной выше лингвистической интерпретации словаря V и языка L конфигурации 1-го ранга будут «потенциальными составляющими», т. е. цепочками, которые могут входить в лингвистически естественные системы составляющих грамматически правильных предложений данного языка. Так, в русском языке цепочка *исключительно важными* является, видимо, конфигурацией 1-го ранга с результирующим *важными* (а также с любым из результирующих *ценными, вредными, скверными*, ...); существуют, очевидно, правильные русские предложения, для которых эта цепочка входит в число «естественных» составляющих [например, (*Эти работы*) (*являются* (*исключительно важными*))].

Однако «потенциальные составляющие» заведомо не исчерпываются конфигурациями 1-го ранга. Например, словосочетание *важными работами* является «потенциальной составляющей» [ср. *Я* (((*не был*) *знаком*) ((*с* (*важными работами*)) *Иванова*))]; в то же время

*) Ср., впрочем, сноску *) на стр. 327.

ясно, что на это словосочетание могут быть замещаемы только такие цепочки, как *работами, лекциями, шапками, докладами, собаками, ушами* *) и т. п., и поэтому, если бы цепочка *важными работами* была конфигурацией 1-го ранга, то только с такими результирующими. Но, например, в фразе *Я не был знаком с исключительно важными работами Иванова* нельзя без нарушения грамматической правильности заменить *важными работами* ни на *работами*, ни на *лекциями*, ни на *собаками*, *ушами* и т. п. Такая замена возможна лишь в тех контекстах, где нет словосочетаний вроде *исключительно важными* или *невероятно важными*, перекрывающихся с заменяемым вхождением словосочетания *важными работами*. Эти соображения приводят к следующему определению.

Пусть $r > 1$ — натуральное число, и для каждого i, $1 \leqslant i < r$, определено понятие конфигурации ранга i языка L. Тогда цепочка $x \in VV^+$ называется конфигурацией ранга r языка L с результирующим a, где $a \in V$, если выполняются следующие два условия: 1) $a \Rightarrow x$; 2) если $z_1xz_2 \in L$ и цепочка z_1xz_2 не содержит вхождений конфигураций рангов, меньших r, перекрывающихся с выделенным вхождением x, но не содержащихся в нем целиком, то $z_1az_2 \in L$.

В русском языке словосочетание *важными работами* является, по-видимому, конфигурацией ранга 2 с результирующим *работами*, или *шапками*, или *лекциями* и т. п.

Замечания. 1) Из определения ясно, что всякая конфигурация ранга r является и конфигурацией любого большего ранга с тем же результирующим.

2) Понятие конфигурации зависит, вообще говоря, от словаря, в котором рассматривается язык: при добавлении к словарю новых символов, не входящих в цепочки языка, могут возникать новые конфигурации (см. упражнение ПII.3). В дальнейшем мы всегда, если не оговорено противное, будем считать, что язык рассматривается в своем «наименьшем» словаре, состоящем в точности из тех символов, которые встречаются в его цепочках. Если рассматривается одновременно несколько языков, то подразумевается словарь, являющийся объединением соответствующих «наименьших».

*) Впрочем, при достаточно широком охвате русских предложений последние три цепочки не будут замещаемы на *важными работами* (ср. ниже, § ПII.3, пример 2).

Будем называть цепочку, принадлежащую языку L, неприводимой, если она не содержит вхождений конфигураций этого языка. Множество неприводимых цепочек языка L будет обозначаться $Б(L)$.

Конфигурацию ранга r языка L назовем простой, если она не содержит вхождений конфигураций ранга r (или, что то же самое, рангов, меньших или равных r) этого языка, отличных от нее самой.

Пусть T — некоторое множество конфигураций языка L и T_1 — множество всевозможных упорядоченных пар вида (x, a), где $x \in T$ и a — результирующий конфигурации x. Если T — множество всех конфигураций, соответственно всех простых конфигураций, языка L, то T_1 будет обозначаться $K(L)$, соответственно $П(L)$. Упорядоченную пару $(Б(L), K(L))$, соответственно $(Б(L), П(L))$, мы будем называть полной, соответственно приведенной, конфигурационной характеристикой языка L.

Лемма ПII.2. *Пусть* $L_1, L_2 \subseteq V^*$. *Если* $Б(L_1) \subseteq Б(L_2)$ *и* $П(L_1) \subseteq K(L_2)$, *то* $L_1 \subseteq L_2$.

Доказательство. Пусть $x \in L_1$. Индукцией по $|x|$ покажем, что $x \in L_2$. Если $|x| = 1$, то из $x \in L_1$ следует $x \in Б(L_1) \subseteq Б(L_2) \subseteq L_2$. Допустим, что для цепочек длины, меньшей $n (n > 1)$, утверждение доказано и $|x| = n$. Если при этом $x \in Б(L_1)$, то $x \in Б(L_2) \subseteq L_2$. Если же $x \notin Б(L_1)$ и r — наименьший ранг конфигураций языка L_1, содержащихся в x, то x содержит и простую конфигурацию w ранга r языка L; для некоторых $z_1, z_2 \in V^*$ и $b \in V$ имеем $x = z_1wz_2$ и $(w, b) \in П(L_1)$. Поскольку x не содержит конфигураций языка L рангов, меньших r, получаем $z_1bz_2 \in L_1$; но $|z_1bz_2| < n$, откуда по индуктивному предположению $z_1bz_2 \in L_2$ и, поскольку $(w, b) \in П(L_1) \subseteq K(L_2)$, $x = z_1wz_2 \in L_2$.

Из леммы ПII.2 непосредственно вытекает

Теорема ПII.1. *Пусть* $L_1, L_2 \in V^*$. *Если* $Б(L_1) = Б(L_2)$ *и либо* $K(L_1) = K(L_2)$, *либо* $П(L_1) = П(L_2)$, *то* $L_1 = L_2$.

Иначе говоря, язык в заданном словаре вполне определяется своей полной или приведенной конфигурационной характеристикой.

Для лингвистических приложений особо интересным представляется класс языков, у которых $Б$ и $П$ конечны. Такие языки мы будем называть конечно характеризуемыми. Кажется в высшей степени правдоподобным, что в естественных языках при всяком разумном уточнении понятия грамматической правильности все правильные предложения и все конфигурации будут содержать вхождения некоторых словосочетаний стандартного вида («прилагательное + наречие», «существительное + прилагательное» и т. п.), являющихся простыми конфигурациями, и поэтому естественные языки (точнее — множества правильных предложений естественных языков) конечно характеризуемы. (В то же время множество всех конфигураций естественного языка, видимо, бесконечно. Так, в русском языке сочетание существительного с любым числом определяющих его прилагательных будет конфигурацией.)

Проиллюстрируем нахождение конфигурационных характеристик на примерах.

Пример 1. Пусть $L = \{ge, aae, abde, bdae, bdbde, cf, def, efe, fff\}$. Заметим прежде всего, что для произвольного языка и произвольного символа α из его «наименьшего» словаря множество Φ_α содержит только такие цепочки, которые входят в цепочки данного языка в качестве подцепочек. Поэтому для конечного языка все Φ_α конечны, и, следовательно, $К$ и $П$ (равно как и $Б$) тоже конечны и могут быть найдены перебором. Положим $\Phi'_\alpha(L) = \{x \mid x \in \Phi_\alpha(L) \,\&\, |x| > 1\}$. Из предшествующего замечания ясно, что если язык L конечен и символ α входит в какую-либо цепочку языка L, имеющую наибольшую возможную для цепочек из L длину, то $\Phi'_\alpha(L)$ пусто. Поэтому в нашем случае $\Phi'_b = \Phi'_d = \Phi'_e = \varnothing$. Множество Φ'_f также пусто, так как иначе язык L содержал бы цепочки вида dex, где $|x| > 1$. Непосредственно проверяется, что $bd \in \Phi'_a$; других цепочек Φ'_a не содержит, иначе язык L содержал бы цепочки вида $xbde$, где $|x| > 1$ и $x \neq bd$. Аналогичным образом легко убедиться, что $\Phi'_c = \{de, ff\}$, $\Phi'_g = \{aa, abd, bda, bdbd, ef\}$. Непосредственно ясно, что цепочки aa, abd, bda и $bdbd$ замещаемы на g и bd — на a, а также, что de и ff не замещаемы на c и ef — на g. Итак, L имеет пять конфигураций 1-го ранга: bd, aa, abd, bda, $bdbd$. Далее, de есть конфигурация 2-го ранга с результирующим c; действительно, из всех цепочек языка L только в def имеется вхождение de, не перекрывающееся с вхождениями конфигураций 1-го ранга, и при этом $cf \in L$. Цепочка ef не является конфигурацией 2-го ранга с результирующим g, поскольку def не содержит вхождений конфигураций 1-го ранга, и все же $gf \notin L$. Аналогично устанавливается, что ff не есть конфигурация 2-го ранга с результирующим c. Следующий шаг будет состоять в проверке оставшихся цепочек ef и ff на «конфигурационность 3-го ранга»; поступая, как выше, убеждаемся, что ef является конфигурацией 3-го ранга (с результирующим g), а ff — нет. Наконец, устанавливаем, что ff не является и конфигурацией 4-го ранга. Итак, имеем

$$Б(L) = \{ge,\ cf,\ fff\},$$

$$K(L) = \{(bd,\ a)(aa,\ g),\ (abd,\ g),\ (bda,\ g),\ (bdbd,\ g),\ (de,\ c),\ (ef,\ g)\},$$

$$П(L) = \{(bd,\ a),\ (aa,\ g),\ (de,\ c)(ef,\ g)\}.$$

Пример 2. Пусть $L' = \{b^n c b^n \mid n = 0, 1, \ldots\}$, $L = a^+ L' a^+$. Легко видеть, что $\Phi_a(L) = \Psi_a(L) = a^+$, $\Phi_b(L) = \{b\}$, $\Phi_c(L) = \Psi_c(L) = L'$. Поэтому $Б(L) = \{aca\}$, $K(L) = \{(a^m,\ a) \mid m = 2, 3, \ldots\} \cup \{(x,\ c) \mid x \in L'\}$, $П(L) = \{(aa,\ a),\ (bcb,\ c)\}$.

Дальнейшие примеры см. в упражнениях ПII. 2, 3, 7; см. также ниже пример 3.

Теорема ПII. 2. *Всякий конечно характеризуемый язык является бесконтекстным. При этом по заданным конечным $Б(L)$ и $П(L)$ можно построить Б-грамматику, порождающую L.*

Доказательство. Пусть L — конечно характеризуемый язык в словаре V. Сопоставим каждому $a \in V$ новый символ $\bar{a}$ и положим $\bar{V} = \{\bar{a} \mid a \in V\}$. Для каждой цепочки $x = a_1 \ldots a_k$, где $a_1, \ldots, a_k \in V$, $k \geqslant 1$, будем полагать $\bar{x} = \bar{a}_1 \ldots \bar{a}_k$; если $y = \bar{x}$, будем писать $x = \tilde{y}$. Введем еще один символ $I \notin V \cup \bar{V}$ и рассмотрим грамматику $\Gamma = \langle V,\ \bar{V} \cup \{I\},\ I,\ R_1 \cup R_2 \cup R_3 \rangle$, где $R_1 = \{I \to z \mid z \in Б(L)\}$, $R_2 = \{\bar{a} \to \bar{x} \mid (x,\ a) \in П(L)\}$, $R_3 = \{\bar{a} \to a \mid a \in V\}$. Покажем, что $L(\Gamma) = L$.

I. Пусть $w = L(\Gamma)$. Тогда найдется такой вывод $(I = \omega_0, \ldots, \omega_k, \ldots, \omega_n = w)$ в Γ, что на первом его

шаге применяется правило из R_1, на всех следующих шагах до k-го включительно $(1 < k \leqslant n-1)$ — правила из R_2 и на всех остальных шагах — правила из R_3. Поскольку $\tilde{\omega}_1 \in Б(L) \subseteq L$ и для каждого $i = 2, \ldots, k$ цепочка $\tilde{\omega}_i$ получается из $\tilde{\omega}_{i-1}$ подстановкой некоторой конфигурации вместо ее результирующего, имеем $\tilde{\omega}_k \in L$; но $\tilde{\omega}_k = w$.

II. Чтобы показать, что $L \subseteq L(\Gamma)$, достаточно убедиться, что всякая цепочка $\bar{w} \in \bar{L}$ выводима в Γ из I. Но это легко сделать, применив индукцию по $|\bar{w}|$ и воспользовавшись рассуждением, аналогичным примененному в доказательстве леммы ПII. 2.

Теорема ПII. 2 не может быть обращена. Более того, как показывает следующий пример, даже А-язык может не быть конечно характеризуемым.

Пример 3. Пусть $L = \{bab\} \cup bcb^+cb \cup (a^+ \cup cb^+c)^+$. Ясно, что $\Phi_a(L) = \{a\} \cup cb^+c$, $\Phi_b(L) = \{b\}$, $\Phi_c(L) = \{c\}$ (например, в $bcbcb$ первое вхождение b нельзя заменить ничем, а первое вхождение c можно заменить только цепочкой вида cb^k, но такой цепочкой при $k \geqslant 1$ нельзя заменить второе вхождение c). Поэтому $K(L) = \{(cb^mc, a) \mid m = 1, 2, \ldots\}$, откуда $Б(L) = \{bab\} \cup a^+$; все конфигурации здесь, очевидно, простые, так что $П(L)$, как и $Б(L)$, бесконечно.

Один частный класс А-грамматик, приводящий к конечно характеризуемым языкам, указан в упражнении ПII. 4.

§ ПII.3. Окрестности, классы и типы

Рассмотренная в предыдущем параграфе система понятий может служить примером анализирующей лингвистической модели. Эта модель описывает некоторые отношения, возникающие между элементами языка в речи (более конкретно — в предложении), или, как говорят лингвисты, «синтагматические отношения»; модели, описывающие такие отношения, принято также называть синтагматическими в отличие от парадигматических моделей, описывающих отношения между элементами языка в его системе — «парадигматические отношения». В настоящем параграфе будет

рассмотрена одна из анализирующих моделей этого последнего типа, также основанная на понятии замещаемости.

Пусть L — язык в словаре V, ρ — отношение эквивалентности на V и θ — алфавитный гомоморфизм V^* на $(V/\rho)^*$. Пусть, кроме того, ρ' — некоторое укрупнение ρ. Мы будем говорить, что ρ' есть L-регулярное укрупнение ρ, если любые два ρ-класса, содержащиеся в одном ρ'-классе, взаимозамещаемы относительно $\theta(L)$. В частности, мы получим L-регулярное укрупнение ρ, если будем считать два элемента V эквивалентными тогда и только тогда, когда они содержатся в ρ-классах, взаимозамещаемых относительно $\theta(L)$. Такую эквивалентность мы будем называть L-производной для ρ; она является, очевидно, наибольшим L-регулярным укрупнением ρ, т. е. служит укрупнением любого L-регулярного укрупнения ρ.

В дальнейших утверждениях L будет означать язык в словаре V, ρ_1 и ρ_2 — эквивалентности на V такие, что ρ_2 — укрупнение ρ_1, и θ_1, θ_2, η_{12} — алфавитные гомоморфизмы V^* на $(V/\rho_1)^*$, V^* на $(V/\rho_2)^*$ и $(V/\rho_1)^*$ на $(V/\rho_2)^*$ соответственно.

Лемма ПII. 3. *Если ρ_2 — L-регулярное укрупнение ρ_1, то $X \in \theta_1(L)$ тогда и только тогда, когда $\eta_{12}(X) \in \theta_2(L)$.*

Доказательство. а) Поскольку $\theta_2(L) = \eta_{12}(\theta_1(L))$, из $X \in \theta_1(L)$ вытекает $\eta_{12}(X) \in \theta_2(L)$.

б) Из определения L-регулярного укрупнения следует — так как взаимозамещаемость является конгруэнцией, — что если X, $Y \in (V/\rho_1)^*$ и $\eta_{12}(X) = \eta_{12}(Y)$, то $X \underset{\theta_1(L)}{\Leftrightarrow} Y$. Поэтому, если $\eta_{12}(X) = Z \in \theta_2(L)$ и z — произвольная цепочка из L такая, что $\theta_2(z) = \eta_{12}(\theta_1(z)) = Z$, имеем $X \underset{\theta_1(L)}{\Leftrightarrow} \theta_1(z)$, откуда $X \in \theta_1(L)$.

Теорема ПII.3. *Если ρ_2 — L-регулярное укрупнение ρ_1, то:*

а) *$X \underset{\theta_1(L)}{\Rightarrow} Y$ тогда и только тогда, когда $\eta_{12}(X) \underset{\theta_2(L)}{\Rightarrow} \eta_{12}(Y)$;*

б) *X является конфигурацией ранга r языка $\theta_1(L)$ с результирующим A тогда и только тогда, когда $\eta_{12}(X)$ является конфигурацией ранга r языка $\theta_2(L)$ с результирующим $\eta_{12}(A)$.*

Доказательство. а) Пусть $X \underset{\theta_1(L)}{\Rightarrow} Y$, и пусть Z_1, Z_2 — произвольные цепочки из $(V/\rho_2)^*$ такие, что $Z_1\eta_{12}(X)\,Z_2 \in \theta_2(L)$. Если T_1, T_2 — произвольные η_{12}-прообразы цепочек Z_1 и Z_2 соответственно, то $\eta_{12}(T_1XT_2) = Z_1\eta_{12}(X)\,Z_2$, откуда по лемме ПII.3 $T_1XT_2 \in \theta_1(L)$; отсюда следует $T_1YT_2 \in \theta_1(L)$ и—снова по лемме ПII.3— $\eta_{12}(T_1YT_2) = Z_1\eta_{12}(Y)Z_2 \in \theta_2(L)$. Аналогично доказывается, что $\eta_{12}(X) \underset{\theta_2(L)}{\Rightarrow} \eta_{12}(Y)$ влечет $X \underset{\theta_1(L)}{\Rightarrow} Y$.

б) При $r = 1$ утверждение справедливо в силу а). Пусть оно доказано для всех $i < r$, и пусть X — конфигурация ранга r языка $\theta_1(L)$ с результирующим A. Пусть Z_1, Z_2 — цепочки из $(V/\rho_2)^*$, T_1, T_2 — произвольные их η_{12}-прообразы, и пусть при этом цепочка $Z_1\eta_{12}(X)\,Z_2$ принадлежит $\theta_2(L)$ и не содержит вхождений конфигураций рангов, меньших r, языка $\theta_2(L)$, перекрывающихся с выделенным вхождением $\eta_{12}(X)$, но не содержащихся в нем целиком. Тогда в силу индуктивного предположения цепочка T_1XT_2 не будет содержать вхождений конфигураций рангов, меньших r, языка $\theta_1(L)$, перекрывающихся с выделенным вхождением X, но не содержащихся в нем целиком; следовательно, поскольку $T_1XT_2 \in \theta_1(L)$, имеем $T_1AT_2 \in \theta_1(L)$, откуда $Z_1\eta_{12}(A)\,Z_2 \in \theta_2(L)$. Кроме того, в силу а) будет $\eta_{12}(A) \underset{\theta_2(L)}{\Rightarrow} \eta_{12}(X)$. Итак, $\eta_{12}(X)$ есть конфигурация ранга r языка $\theta_2(L)$ с результирующим $\eta_{12}(A)$. Обратное доказывается аналогично.

Лемма ПII.4. *Если* ρ_2 *— L-регулярное укрупнение* ρ_1 *и* ρ_3 *— L-регулярное укрупнение* ρ_2, *то* ρ_3 *есть L-регулярное укрупнение* ρ_1.

Доказательство. Пусть A, $A' \in V/\rho_1$, $C \in V/\rho_3$, A, $A' \subseteq C$. Положим $\eta_{12}(A) = B$, $\eta_{12}(A') = B'$. Поскольку $A \subseteq B$ и $A' \subseteq B'$, пересечения $B \cap C$ и $B' \cap C$ не пусты и, следовательно, $B \subseteq C$, $B' \subseteq C$; поэтому $B \underset{\theta_2(L)}{\Leftrightarrow} B'$, откуда по теореме ПII.3 $A \underset{\theta_1(L)}{\Leftrightarrow} A'$.

Из леммы ПII. 4 немедленно вытекает

Теорема ПII. 4. *Если* ρ' *— L-производная эквивалентность для* ρ *и* ρ'' *— L-производная эквивалентность для* ρ', *то* ρ'' *совпадает с* ρ'.

Рассмотрим теперь эквивалентность, L-производную для равенства. Эта эквивалентность есть, очевидно, не

что иное, как взаимозамещаемость относительно L, ограниченная цепочками длины 1. Для этой эквивалентности мы будем, наряду с записью $\underset{V, L}{\Leftrightarrow}$, пользоваться также обозначением S_L; соответствующие классы эквивалентности будем называть семействами. Разбиение на семейства, т. е. на классы «синтаксически эквивалентных» ЛЗ-сегментов, представляет собой, видимо, единственную содержательно интересную классификацию слов, точнее, ЛЗ-сегментов, которую можно получить, исходя лишь из языка L, т. е. пользуясь только понятием грамматической правильности и не принимая во внимание лексических значений, как мы до сих пор фактически поступали *). Чтобы получить другие классификации, в частности такие, которые могли бы служить в какой-то степени моделями понятий части речи, грамматического рода и т. п., нужно учитывать и лексические значения. А поскольку элементарный символ, т. е. ЛЗ-сегмент, является для нас неразложимой единицей, лексические значения тем более неразложимы, и поэтому их учет не может означать ничего другого, кроме включения в число исходных данных сведений о том, какие ЛЗ-сегменты отвечают одинаковым лексическим значениям и какие — разным. Иначе говоря, мы должны считать заданными множества ЛЗ-сегментов, имеющих одинаковые лексические значения, т. е. являющихся «формами одного слова». Эти множества мы назовем окрестностями. Примеры окрестностей для русского языка: {*физик, физика, физику, физиком, физике, физики, физиков, физикам, физиками, физиках*}, {*пальто*}, {*над*}**).

*) Это означает, что элементарные символы во всех предыдущих рассмотрениях можно было бы интерпретировать и как сегменты (ср. стр. 315). Конкретные результаты зависят, разумеется, от выбора интерпретации. Например, сегменту *физику* отвечают два ЛЗ-сегмента, скажем a_1 — с лексическим значением, включающим компонент «наука», — и a_2 — с лексическим значением, включающим компонент «специалист»; в то же время для сегментов *химию* и *химику* имеется по одному ЛЗ-сегменту — пусть это будут b и c. Трактуя предложения как цепочки ЛЗ-сегментов, получим $a_1 \Leftrightarrow b$, $a_2 \Leftrightarrow c$, а при отказе от учета лексических значений сегменты *физику*, *химию* и *химику* распределятся по трем разным семействам.

**) Здесь для простоты вместо ЛЗ-сегментов выписаны сегменты; необходимое уточнение очевидно — например, вместо сегмента *физику* следовало бы взять ЛЗ-сегмент a_2 из предыдущей сноски.

Ясно, что каждый ЛЗ-сегмент принадлежит одной и только одной окрестности. Таким образом, в число исходных данных наряду со словарем и языком должно быть включено некоторое отношение эквивалентности на словаре, интерпретируемое как совпадение лексических значений. Это приводит нас к рассмотрению в качестве исходного объекта упорядоченной тройки (V, L, Γ), где V — конечное множество (словарь), L — язык в словаре V и Γ — эквивалентность на V. Такую тройку мы назовем лексически размеченным языком; классы эквивалентности по Γ будем называть окрестностями.

Для произвольной эквивалентности ρ на V и произвольного $a \in V$ мы будем обозначать через $\rho(a)$ тот ρ-класс, который содержит a.

Пусть (V, L, Γ) — лексически размеченный язык. Эквивалентности S_L и Γ индуцируют две системы подмножеств словаря V: систему семейств и систему окрестностей.

Определим теперь с помощью тех же эквивалентностей еще две системы подмножеств V: систему классов и систему типов.

1) Для произвольного семейства σ, соответственно для произвольной окрестности γ, будем называть классом, порожденным семейством σ (окрестностью γ), объединение всех окрестностей (соответственно семейств), пересекающихся с σ (с γ).

2) Классы эквивалентности, L-производной для Γ (эта эквивалентность будет обозначаться $T_{L,\Gamma}$), мы будем называть типами.

Классы, в отличие от типов, не обязаны быть попарно непересекающимися множествами. Это, однако, имеет место в одном интересном частном случае, который мы сейчас рассмотрим.

Лексически размеченный язык (V, L, Γ) называется однородным, если, каковы бы ни были две окрестности, пересекающиеся с одним и тем же семейством, любое семейство, пересекающееся с одной из этих окрестностей, пересекается и с другой, или, другими словами, если из aS_Lb следует, что для любого $a' \in \Gamma(a)$ найдется $b' \in \Gamma(b)$ такое, что $a'S_Lb'$ (содержательно — из взаимозамещаемости хотя бы одной формы слова*) A с некоторой формой слова B следует, что и любая форма слова A взаимозамещаема с какой-либо формой слова B).

Замечание. Легко понять, что приведенное определение равносильно следующему: лексически размеченный язык однороден, если, каковы бы ни были два семейства, пересекающиеся с одной и той же окрестностью, любая окрестность, пересекающаяся с одним из этих семейств, пересекается и с другим. Таким образом, семейства и окрестности в определении однородности равноправны.

Пример 1. Рассмотрим шесть предложений: *Есть новый дом*, *Он боится нового дома*, *Он идет к новому дому*, *Он любит новый дом*, *Он знакомится с новым домом*, *Он говорит о новом доме*. Обозначим через L множество, состоящее из этих предложений и всех тех, которые можно получить из них, подставляя вместо форм слова *дом* соответствующие формы слов *стол*, *лампа*, *грядка* и изменяя, если нужно, прилагательные так, чтобы предложение осталось правильным. Через V обозначим множество всех ЛЗ-сегментов**), встречающихся в цепочках из L. Эквивалентность Γ определим естественным образом (например, *дом* Γ *дому*, ..., *новый* Γ *новым* Γ *новую* ... и т. п.).

Легко убедиться, что S_L (*дом*) = {*дом*, *стол*}, S_L(*лампе*) = {*лампе*, *грядке*} и т. д. Непосредственно проверяется, что лексически размеченный язык (V, L, Γ) однороден. Имеем, например, *дом* S_L *стол*; для ЛЗ-сегмента *дому* взаимозамещаемой с ним формой слова *стол* будет *столу*, для *домом* — *столом* и т. п. (рис. 17, *а*)).

Добавим теперь к V всевозможные ЛЗ-сегменты, являющиеся формами единственного числа слов *ведро*, *дерево*, и к L — предложения, полученные из шести исходных заменой форм слова *дом* формами слов *ведро*, *дерево* так же, как раньше. Эквивалентность Γ доопределим естественным образом. Новый лексически

*) Точнее — лексемы. (Лексема — это «слово как единица словаря» [Зализняк 1967, стр. 20]; разумеется, речь здесь идет об обычном словаре, а не о словаре в том смысле, в котором это слово постоянно употребляется в настоящей книге.)

**) Фактически мы выписываем сегменты (ср. сноску *) на стр. 327).

размеченный язык — обозначим его (V', L', Γ') — уже не будет однородным. Действительно, легко проверить, что $S_{L'}$ (*дом*) = {*дом*, *стол*} и $S_{L'}$ (*дому*) = {*дому*, *столу*, *ведру*, *дереву*}. Поэтому $S_{L'}$ (*ведру*) $\cap$ Γ' (*дом*) = {*дому*} $\neq$ $\neq \varnothing$, но Γ' (*ведру*) $\cap$ $S_{L'}$ (*дом*) $= \varnothing$ (рис. 17, *б*)).

Если, наконец, добавить к каждому предложению из L' придаточное *который* (*-ая*, *-ое*) *всем понравился*

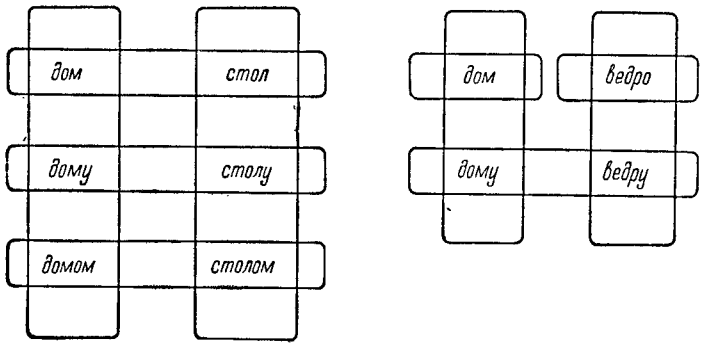


Рис. 17.

(*-ась*, *-ось*) (с соблюдением согласования) и доопределить соответствующим образом V и Γ', то полученный лексически размеченный язык (V'', L'', Γ'') снова будет однородным, так как в нем каждое семейство существительных содержит формы слов только одного рода; например, *ведру* нельзя заменить на *дому* в предложении *Он идет к новому ведру, которое всем понравилось.*

Множества Γ'' (*дом*) $\cup$ Γ'' (*стол*), Γ'' (*лампа*) $\cup$ Γ'' (*грядка*) и Γ'' (*ведро*) $\cup$ Γ'' (*дерево*), т. е. множества всех форм существительных одного рода, будут, очевидно, в (V'', L'', Γ'') классами. Окрестность Γ'' (*новый*) образует отдельный класс. Обозначая через θ алфавитный гомоморфизм $(V'')^*$ на $(V''/\Gamma'')^*$, видим, что все окрестности существительных взаимозамещаемы относительно $\theta(L'')$ (поскольку для каждого из слов, согласующихся с существительными, — *новый*, *который*, *понравился*, — формы всех трех родов входят в одну окрестность). Поэтому существительные образуют тип. Остальные классы и типы,

как легко убедиться, совпадают с окрестностями. В частности, особый тип образуют все формы слова *новый*; легко понять, что если бы мы добавили к V'' еще какие-нибудь прилагательные, соответствующим образом расширив L'' и доопределив Γ'', то все прилагательные составили бы один класс и один тип (а прочие классы и типы остались бы без изменения).

Таким образом, классы в (V'', L'', Γ'') попарно не пересекаются, так что отношение «принадлежать одному классу» есть эквивалентность на V''. Сразу видно, что эта эквивалентность является L''-регулярным укрупнением Γ'' и ее L''-производная эквивалентность совпадает с $T_{L'', \Gamma''}$. Следующие рассмотрения показывают, что такая же ситуация имеет место для всякого однородного лексически размеченного языка.

Теорема III. 5. *В однородном лексически размеченном языке всякий класс порождается любым семейством* (*любой окрестностью*), *пересекающимся* (*-ейся*) *с ним.*

Доказательство ввиду замечания, сделанного после определения однородности, достаточно провести для случая класса, порожденного семейством.

Пусть (V, L, Γ) — однородный лексически размеченный язык и $\varkappa$ — класс, порожденный семейством σ. Рассмотрим произвольное семейство σ', пересекающееся с $\varkappa$, и порожденный им класс $\varkappa'$. Условие $\sigma' \cap \varkappa \neq \varnothing$ означает, что σ' пересекается с некоторой окрестностью, пересекающейся с σ; поэтому, по вышеупомянутому замечанию, любая окрестность, пересекающаяся с σ, пересекается и с σ', и обратно, что дает $\varkappa = \varkappa'$.

Пусть теперь γ — произвольная окрестность, пересекающаяся с $\varkappa$, и $\varkappa''$ — порожденный ею класс. Окрестность γ пересекается с некоторой окрестностью, пересекающейся с σ, и, значит, совпадает с ней; таким образом, $\gamma \cap \sigma \neq \varnothing$. Если $a \in \varkappa$, то $\Gamma(a)$ пересекается с σ и, стало быть (по определению однородности), $S_L(a)$ пересекается с γ, что означает $S_L(a) \subseteq \varkappa''$, так что $a \in \varkappa''$. Следовательно, $\varkappa \subseteq \varkappa''$. Если $b \in \varkappa''$, то $S_L(b)$ пересекается с γ; поэтому, по использовавшемуся выше замечанию, $\Gamma(b)$ пересекается с σ, т. е. $\Gamma(b) \subseteq \varkappa$ и $b \in \varkappa$; итак, $\varkappa'' \subseteq \varkappa$.

Замечание. Обращение этой теоремы также справедливо, и притом в усиленной форме (см. упражнение ПII. 10).

Следствие. *Если* (V, L, Γ) *— однородный лексически размеченный язык, то любые два его класса либо совпадают, либо не пересекаются. Иначе говоря, существует эквивалентность* $K_{L,\Gamma}$ *на* V *такая, что* $aK_{L,\Gamma}b$ *тогда и только тогда, когда* a *и* b *принадлежат одному классу.*

Ясно, что $K_{L,\Gamma}$ является укрупнением для S_L и для Γ. Но сверх того имеет место

Теорема ПII. 6. *Если* (V, L, Γ) *— однородный лексически размеченный язык, то* $K_{L,\Gamma}$ *является* L*-регулярным укрупнением* Γ.

Доказательство. Нам нужно показать, что в условиях теоремы любые две окрестности, содержащиеся в одном классе, взаимозамещаемы относительно языка $\theta_\Gamma(L)$, где θ_Γ — алфавитный гомоморфизм V^* на $(V/\Gamma)^*$. Пусть $\varkappa$ — класс, γ_1 и γ_2 — окрестности, содержащиеся в $\varkappa$, и ξ, η — цепочки окрестностей такие, что $\xi\gamma_1\eta \in \theta_\Gamma(L)$. Тогда найдутся такие цепочки x, $y \in V^*$ и такое $a \in \gamma_1$, что $\theta_\Gamma(x) = \xi$, $\theta_\Gamma(y) = \eta$ и $xay \in L$. Ввиду однородности (V, L, Γ) класс $\varkappa$ порождается некоторым семейством, а отсюда по той же причине следует, что в γ_2 найдется элемент b, взаимозамещаемый с a, так что $xby \in L$, и поэтому $\xi\gamma_2\eta \in \theta_\Gamma(L)$. Итак, $\gamma_1 \Rightarrow \gamma_2$; обратное доказывается так же.

Пусть теперь $K'_{L,\Gamma}$ — L-производная эквивалентность для $K_{L,\Gamma}$. По теореме ПII.6 и лемме ПII.4 $K'_{L,\Gamma}$ есть L-регулярное укрупнение Γ; поэтому $T_{L,\Gamma}$ является укрупнением $K'_{L,\Gamma}$. В то же время по теореме ПII.3 из взаимозамещаемости двух окрестностей относительно $\theta_\Gamma(L)$ следует взаимозамещаемость содержащих эти окрестности классов относительно $\theta_{K_{L,\Gamma}}(L)$ ($\theta_{K_{L,\Gamma}}$ — алфавитный гомоморфизм V^* на $(V/K_{L,\Gamma})^*$), а это означает, что $K'_{L,\Gamma}$ есть укрупнение $T_{L,\Gamma}$. Итак, $K'_{L,\Gamma} = T_{L,\Gamma}$.

В русском языке однородность имеет место лишь для «малых» фрагментов, наподобие рассмотренного в примере 1; как мы сейчас увидим, при существенном расширении этого фрагмента однородность не сохраняется. Но для языков с более простой системой именного словоизменения (идеальный случай — эсперанто) можно, видимо, обеспечить однородность и в довольно представительных моделях.

Понятие класса может, впрочем, служить основой содержательно интересных «категоризаций» и при отсутствии однородности; что же касается понятия типа, то на его лингвистическую интерпретацию условие однородности, видимо, существенным образом не влияет. Проиллюстрируем это на примере, которым мы и завершим настоящее приложение.

Пример 2. Расширим построенный в примере 1 лексически размеченный язык (V'', L'', Γ''), добавляя к L'' последовательно следующие предложения:

1) все предложения, получаемые из прежних заменой форм единственного числа существительных и прилагательных соответствующими формами множественного числа *) и одновременной заменой придаточного *который* (*-ая*, *-ое*) *всем понравился* (*-ась*, *-ось*) другим придаточным: *один* (*одна*, *одно*) *из которых всем понравился* (*-ась*, *-ось*), причем должна сохраняться грамматическая правильность **);

2) все предложения, получаемые из имеющихся после предыдущего шага заменой форм существительных и прилагательных формами любых других существительных, соответственно прилагательных, с сохранением грамматической правильности, а также выбрасыванием прилагательных ***);

3) все предложения, получаемые из имеющихся после шага 2) заменой форм глаголов, кроме *есть* и *понравился* (*-ась*, *-ось*), соответствующими формами любых других глаголов, для которых это возможно без нарушения грамматической правильности;

*) При этом следовало бы перейти от орфографической записи к акцентуированной орфографической [Зализняк 1967, стр. 9], т. е. обозначать ударение, иначе такие формы, как *окна́* и *о́кна*, будут совпадать, что для наших целей нежелательно. Для простоты мы будем, однако, ставить знак ударения лишь там, где без него возможна путаница.

**) Мы могли бы обойтись здесь без таких понятий, как существительное, прилагательное, число, падеж и т. п. (ср. стр. 314), но тогда пришлось бы задавать предложения списком. Поскольку наш пример носит чисто иллюстративный характер, мы ради удобства изложения разрешаем себе использование указанных понятий.

***) При этом мы считаем тождественными ЛЗ-сегменты *о* и *об* (фонетически обусловленные варианты предлога).

4) все предложения, получаемые из имеющихся после шага 3) заменой местоимения *он* на *я*, *ты*, *мы*, *вы*, *они* соответственно, с одновременным изменением форм глаголов, обеспечивающим сохранение грамматической правильности.

Одновременно расширим словарь, добавляя к нему все слова, встречающиеся в новых предложениях; окрестности доопределим естественным образом, причем такие ЛЗ-сегменты, как *стол* и *столы*, *новый* и *новые*, *идет* и *идут*, будем относить каждый раз к одной окрестности *), в одну окрестность объединим также все личные местоимения.

Мы построили, таким образом, еще один лексически размеченный язык — пусть это будет (V_0, L_0, Γ_0). Посмотрим, как выглядят в нем семейства, классы и типы.

ЛЗ-сегменты, входящие в V_0, можно разделить на «типовые» (формы существительных, прилагательных и глаголов, кроме *есть* и *понравился*, *-ась*, *-ось*) и «изолированные» (все прочие). Для каждого из «изолированных» ЛЗ-сегментов непосредственно ясно, что он не взаимозамещаем ни с каким другим, т. е. образует одноэлементное семейство. Ясно также, что форма существительного, прилагательного или глагола может быть замещаема только на форму существительного же, прилагательного и глагола соответственно. Учитывая это замечание, найдем семейства «типовых» ЛЗ-сегментов.

Пусть S — некоторая окрестность существительного. Назовем «парадигмой» окрестности S упорядоченную систему двенадцати ЛЗ-сегментов из S: $\langle s_1, \ldots, s_{12}\rangle$, в которой на первом месте стоит форма именительного падежа единственного числа данного существительного, на втором — форма родительного падежа единственного числа, ..., на двенадцатом — форма предложного падежа множественного числа. Если данное существительное не имеет каких-либо падежно-числовых форм, на соответствующих местах парадигмы будем ставить 0. Аналогично определим парадигму окрестности прилагательного (состоящую из 72 родо-падежно-число-«одушевленностных» форм) и окрестности глагола (состоящую из 6 лично-числовых форм настоящего времени) *).

*) Для существительных это не единственный содержательно естественный способ выделения окрестностей — иногда такие формы, как *стол* и *столы*, относят к разным лексемам. См. об этом [Зализняк 1967, стр. 55—57].

Для краткости мы будем называть ЛЗ-сегменты, являющиеся формами существительных, прилагательных и глаголов, С-сегментами, П-сегментами и Г-сегментами соответственно. Будем говорить также о С-, П- и Г-семействах, С-, П- и Г-окрестностях и т. д.

Теперь нетрудно проверить (это предоставляется читателю), что С-сегменты s и s', относящиеся к лексемам S и S' соответственно, тогда и только тогда взаимозамещаемы, когда выполняются следующие два условия: а) s занимает в своей парадигме все те и только те места, которые s' занимает в своей; б) для каждой падежно-числовой формы лексемы S, реализуемой ЛЗ-сегментом s, множество согласующихся с этой формой П-сегментов то же самое, что и для соответствующей формы лексемы S'.

Например, С-сегменты *окно* и *здание* взаимозамещаемы. С-сегменты *окно* и *поле* не взаимозамещаемы, так как не выполнено условие а): С-сегмент *поле* занимает в своей парадигме, в частности, шестое место, тогда как *окно* в своей парадигме этого места не занимает (так что *поле* нельзя заменить на *окно* в предложении *Он говорит о новом поле, которое всем понравилось*). Для С-сегментов *портье* и *кофе* условие а) выполнено (оба они занимают в своих парадигмах все места), но не выполнено условие б): множества П-сегментов, согласующихся с формами винительного падежа единственного **) числа соответствующих лексем, различны (и поэтому *портье* нельзя заменить на *кофе* в *Он любит нового портье, который всем понравился*).

Среди С-семейств будут, например, такие, как {*дом, гвоздь*, ...}, {*до́ма, гвоздя*, ...}, {*дома́, гвозди*, ...}, {*тигр, конь*, ...}, {*тигра, коня*, ...}, {*тигры, кони*, ...}, {*дому, гвоздю, тигру, коню*, ...}, {*окна́, ведра́*, ...}, {*здания, решения*, ...}, {*окно, ведро, здание, решение*, ...}.

*) Для простоты мы считаем, что, например, для существительного каждая падежно-числовая форма реализуется единственным образом, хотя на самом деле это не всегда так (ср. *лампой* и *лампою*); если возможны разные (свободные) варианты, будем предполагать, что выбран «более регулярный» (*лампой*, а не *лампою*).

**) Равно как и множественного.

Легко заметить, что однородность для (V_0, L_0, Γ_0) не имеет места: например, окрестности С-сегментов *дом* и *тигр* пересекаются с семейством {*дому*, *тигру*, ...}, но семейство {*дом*, ...}, пересекающееся с первой из этих окрестностей, не пересекается со второй (т. е. С-сегмент *дом* не взаимозамещаем ни с одной формой лексемы *тигр*); аналогично ведут себя окрестности С-сегментов *окно* и *поле* *).

Легко видеть, далее, что П-сегменты s и s' взаимозамещаемы тогда и только тогда, когда s занимает в своей парадигме в точности те места, которые s' занимает в своей. Так, *новый* и *синий*, *новых* и *больших* взаимозамещаемы, а *новый* и *синяя*, *новый* и *большой*—нет.

Чтобы описать Г-семейства, введем следующее понятие. Скажем, что два глагола (Г-лексемы) имеют о д и н а к о в ы е с х е м ы у п р а в л е н и я, если среди тех пяти падежных и предложно-падежных конструкций, которые встречаются в наших предложениях ((1) род. пад. без предлога; (2) *к* + дат. пад.; (3) вин. пад. без предлога; (4) *с* + тв. пад.; (5) *о* + предл. пад.), эти глаголы могут управлять в точности одними и теми же **). Одинаковые схемы управления имеют, например, глаголы *знакомиться* и *соглашаться* (управляют конструкцией (4)), *вспоминать* и *забывать* (управляют конструкциями (3) и (5)). Легко убедиться теперь, что два Г-сегмента тогда и только тогда взаимозамещаемы, когда они занимают в своих парадигмах одинаковые места *) и соответствующие лексемы имеют одинаковые схемы управления.

*) Несколько расширив модель, можно устранить взаимозамещаемость С-сегментов *дому* и *тигру* и вообще форм неодушевленных существительных с формами одушевленных; достаточно добавить предложения типа *Он идет к дому, который все видели* (... *к тигру, которого* ..., ... *к домам, один из которых* ..., ... *к тиграм, одного из которых* ...). В то же время *окну* и *полю* останутся, видимо, взаимозамещаемыми при добавлении любых правильных предложений. (Аналогично *мышь* и *мышка* при отсутствии взаимозамещаемости *мыши* и *мышки* и т. п.) Поэтому действительным источником неоднородности можно считать наличие существительных, одинаково согласующихся (т. е. не различающихся родом и одушевленностью), но по-разному склоняющихся (т. е. имеющих разные схемы склонения — см. ниже), а также по-разному склоняющихся прилагательных (как *новый* и *большой* — см. ниже).

**) Говоря, что глагол может управлять той или иной конструкцией, мы подразумеваем, что он может ею управлять при отсутствии других дополнений (так как в L_0 нет предложений более чем с одним дополнением). Например, глагол *направлять* мы не считаем управляющим конструкцией (2), поскольку нельзя сказать * *Он направляет к берегу*, хотя и можно *Он направляет лодку к берегу*.

Переходя к классам, сразу видим, что С-семейства и С-окрестности порождают, вообще говоря, разные классы. Так, класс $\varkappa$, порождаемый окрестностью γ = {*дом*, *до́ма*, ...}, не порождается никаким семейством. (В самом деле, семейство, порождающее $\varkappa$, пересекалось бы с γ; непосредственным перебором пересекающихся с γ семейств — их имеется 10 — нетрудно убедиться, что ни одно из них не может порождать $\varkappa$. Например, класс, порождаемый семейством σ_1 = {*дом*, *гвоздь*, ...}, не пересекается с окрестностью γ' = {*тигр*, *тигра*, ...}, тогда как класс $\varkappa$ с ней пересекается, поскольку он содержит семейство σ_3 = {*дому*, *гвоздю*, *тигру*, ...}; в свою очередь, класс, порождаемый семейством σ_3, содержит окрестность γ', но $\varkappa$ ее не содержит, так как *тигр* $\notin \varkappa$. Для остальных семейств аналогично.) Нас будут интересовать С-классы, порождаемые семействами. Каждый из вышеприведенных примеров С-семейств сразу дает пример такого класса, нужно только собрать вместе все формы тех существительных, для которых одна какая-нибудь форма входит в данное семейство. Эти примеры показывают, в частности, что класс, порожденный С-семейством, может быть собственной частью другого такого класса. Возможно и частичное пересечение (например, классы, порождаемые семействами {*поле*, *чудовище*, ...} и {*о́кна*, *поля́*, ...}). Определим на V_0 следующее отношение P: sPs' тогда и только тогда, когда любой класс, порождаемый семейством, либо одновременно содержит, либо одновременно не содержит s и s'. Это отношение является, очевидно, эквивалентностью; подмножества (классы эквивалентности), на которые оно разбивает V_0, мы назовем п р и в е д е н н ы м и к л а с с а м и. Ясно, что всякий приведенный класс есть объединение окрестностей. Приведенные С-классы имеют весьма простой содержательный смысл; чтобы выяснить его, введем два новых понятия.

*) Г-сегмент может занимать в парадигме только одно место.

Будем говорить, что два существительных (С-лексемы) с парадигмами $\langle s_1, \ldots, s_{12}\rangle$ и $\langle s'_1, \ldots, s'_{12}\rangle$ имеют одинаковые схемы склонения, если: а) тогда и только тогда $s_i = 0$, когда $s'_i = 0$; б) тогда и только тогда $s_i = s_j$, когда $s'_i = s'_j$. Например, существительные *дом, гвоздь, ведро, дерево* имеют одинаковые схемы склонения. Другая схема склонения представлена словами *тигр, конь*. Еще примеры схем склонения: {*поле, море, ...*}, {*лампа, грядка, ...*}, {*вещь, соль, путь, ...*}, {*лань, дочь, ...*}, {*пальто, кофе, какаду, мадам, ...*}, {*сани, очки, ...*} *).

Скажем далее, что два существительных имеют одинаковые схемы согласования, если с одинаковыми падежно-числовыми формами этих существительных согласуются в точности одни и те же формы прилагательных. Одинаковые схемы согласования имеют, например, существительные *дом, гвоздь, путь, кофе*; *тигр, конь, какаду* дают другую схему согласования и т. д.**).

Нетрудно убедиться теперь, что окрестности двух существительных тогда и только тогда содержатся в одном приведенном классе, когда эти существительные имеют одинаковые схемы склонения и одинаковые схемы согласования.

Для прилагательных классы, порождаемые семействами и окрестностями, также не совпадают. Приведенные классы и здесь имеют простую интерпретацию; именно, определив для прилагательных схемы склонения так же, как для существительных, без труда убеждаемся, что две П-окрестности содержатся в одном приведенном классе тогда и только тогда, когда соответствующие лексемы имеют одинаковые схемы склонения ***).

*) В последнем примере на первых шести местах парадигмы стоят нули (при традиционной трактовке Pluralia tantum; о возможности другой трактовки см. [Зализняк 1967, стр. 57—61]).

**) Схема согласования определяется, очевидно, родом и одушевленностью (неодушевленностью), так что существительные, имеющие все падежно-числовые формы, дают шесть схем согласования (при традиционной трехродовой системе — ср. [Зализняк 1967, стр. 78—80]).

***) Приведенных П-классов, кажется, только два: {*новый, синий, ...*} и {*большой, прямой, ...*}.

Г-классы устроены проще. Здесь имеет место «локальная однородность»: если в формулировке условия однородности заменить слова «семейство» и «окрестность» на «Г-семейство» и «Г-окрестность» соответственно, то наша модель будет удовлетворять такому условию. Поэтому классы, порождаемые Г-семействами и Г-окрестностями, совпадают; два Г-сегмента принадлежат одному классу (или, безразлично, приведенному классу) тогда и только тогда, когда соответствующие лексемы имеют одинаковые схемы управления.

Обратимся, наконец, к типам. Все существительные образуют один тип — это устанавливается точно так же, как для лексически размеченного языка (V'', L'', Γ'') в примере 1. Аналогичным образом легко видеть, что и все прилагательные принадлежат одному типу. Наконец, тип глагола зависит только от того, какими предлогами он может управлять; таким образом, отличие Г-типов от Г-классов сводится к следующему: если два Г-класса различаются только тем, что либо а) глаголы одного из этих классов могут управлять конструкцией (1), а глаголы другого класса не могут, но могут управлять конструкцией (3), либо б) то же с переменой местами конструкций (1) и (3), либо в) глаголы одного класса могут управлять каждой из конструкций (1) и (3), а другого — только одной определенной, то такие классы содержатся в одном типе (так, *избегать*, *любить* и *ждать* попадут в один тип); остальные Г-типы совпадают с Г-классами.

Впрочем, в отношении Г-типов наш пример не показателен, так как он охватывает слишком мало случаев употребления глаголов. Все приведенные рассмотрения останутся, однако, в силе, если расширить L_0, включив туда предложения с другими падежными и предложно-падежными конструкциями (в любом количестве, ограниченном только реальным наличием таких конструкций в языке) и, в частности, предложения с несколькими дополнениями (среди них и такие, где разные дополнения выражены конструкциями с одним и тем же предлогом, ср. *знакомится с делами с прошлого года*), а также без дополнений; будем считать, что вместе с каждым предложением наш язык включает и все те, которые получаются из него перестановкой дополнений. При этом

принципы выделения семейств, классов и типов не изменятся; для существительных и прилагательных не изменятся и сами семейства, классы и типы. Для глаголов реальные семейства, классы и типы станут другими — не только потому, что добавятся новые глаголы, но и потому, что для многих старых глаголов изменятся схемы управления. Ограничимся описанием Г-типов; ему можно придать следующий вид. Перенумеруем как-нибудь все предлоги: $p_1, \ldots, p_s$. Назовем шаблоном управления произвольную упорядоченную систему $s+1$ целых неотрицательных чисел и скажем, что шаблон управления $(i_0, i_1, \ldots, i_s)$ допускается некоторым глаголом, если хотя бы в одном предложении нашего языка, содержащем этот глагол, имеется точно i_0 беспредложных дополнений, i_1 дополнений с предлогом p_1, i_2 с предлогом p_2 и т. д. Нетрудно понять, что окрестности двух глаголов содержатся в одном типе тогда и только тогда, когда совпадают множества допускаемых этими глаголами шаблонов управления.

Упражнения

ПII. 1. а) Указать пример бесконечного линейного языка, не имеющего нетривиальной замещаемости (определение см. в упражнении 8.6).

б) Может ли бесконечный ОА-язык не иметь нетривиальной замещаемости?

ПII. 2. Найти полные и приведенные конфигурационные характеристики следующих языков (в их «наименьших словарях»):

а) $\{abcde, afe, bbcd, bf, gde, abc, g\}$;

б) $\{bab\} \cup a^+$;

в) $\{fg\} \cup [c\,(d^+ - \{d\})\,eg]$.

ПII. 3. Показать, что язык

$$a\{ee, ccdd\}^* \cup \{ff, ccdd\}^* b$$

является конечно характеризуемым в словаре $\{a, b, c, d, e, f\}$ и не является таковым в словаре $\{a, b, c, d, e, f, g\}$.

ПII. 4. Назовем А-грамматикой без омонимии такую А-грамматику, в диаграмме которой никакие две дуги, помеченные одним и тем же символом, не могут исходить из разных узлов.

а) Показать, что язык, порождаемый А-грамматикой без омонимии, конечно характеризуем.

б) Указать алгоритм, позволяющий для всякой А-грамматики без омонимии Г построить приведенную конфигурационную характеристику языка $L(\Gamma)$.

[Гладкий 1963(а)].

ПII. 5. Указать алгоритм, позволяющий для любой А-грамматики Г с основным словарем V и любых цепочек $x, y \in V^*$ распознать, имеет ли место $x \underset{L(\Gamma)}{\Rightarrow} y$.

ПII. 6. Показать, что для любого А-языка L в словаре V, любого $b \in V$ и любого $r = 1, 2, \ldots$ множество $K_r(b, L)$ конфигураций ранга r языка L с результирующим b также является А-языком, и при этом по А-грамматике Г, символу b и числу r можно построить А-грамматику, порождающую $K_r(b, L(\Gamma))$ [Гладкий 1964 (а)].

ПII. 7. Показать, что в линейном языке

$$\{fg, ae, ad, ag\} \cup \{d^m a^n eg \mid n = 0, 1, \ldots;\ m = 1, \ldots, n+1\}$$

каждая цепочка вида

$$d^{k+1-j} a^k e \quad (k = 0, 1, \ldots;\ j = 0, 1, \ldots, k)$$

является конфигурацией ранга $j+1$ с результирующим f [Лучкин 1966].

ПII. 8. Указать пример лексически размеченного языка (V, L, Γ), в котором $T_{L,\Gamma}$ не является укрупнением S_L [Успенский 1957].

ПII. 9. Пусть (V, L, Γ) — лексически размеченный язык. Определим на V отношение эквивалентности $R_{L,\Gamma}$ следующим образом: $xR_{L,\Gamma}y$, если существуют $x_0, \ldots, x_k \in V$ такие, что $x_0 = x$, $x_k = y$ и для каждого $i = 1, \ldots, k$ либо $x_{i-1} S_L x_i$, либо $x_{i-1} \Gamma x_i$. (Классы эквивалентности $R_{L,\Gamma}$ называются разделами [Успенский 1957].)

а) Всегда ли $R_{L,\Gamma}$ является L-регулярным укрупнением $K_{L,\Gamma}$?

б) Показать, что если $T_{L,\Gamma}$ является укрупнением S_L (ср. упражнение ПII. 8), то L-производная эквивалентность для $R_{L,\Gamma}$ совпадает с $T_{L,\Gamma}$.

ПII. 10. Показать, что каждое из следующих условий достаточно для однородности лексически размеченного языка:

а) всякий класс, порожденный семейством, порождается любым семейством, с ним пересекающимся;

б) всякий класс, порожденный семейством, порождается любой окрестностью, с ним пересекающейся;

в) всякий класс, порожденный окрестностью, порождается любым семейством, с ним пересекающимся;

г) всякий класс, порожденный окрестностью, порождается любой окрестностью, с ним пересекающейся.

ПII. 11. Является ли достаточным условием однородности лексически размеченного языка попарное непересечение различных классов, порожденных семействами (или порожденных окрестностями, или любых)?

ПII. 12. Пусть (V, L, Γ) — лексически размеченный язык. Показать, что:

а) символы $x, y \in V$ тогда и только тогда принадлежат одному приведенному классу, когда любой элемент $\Gamma(x)$ взаимозамещаем

с некоторым элементом $\Gamma(y)$, и обратно (определение И. И. Ревзина [Ревзин 1967, стр. 159]);

б) непустое подмножество V тогда и только тогда является приведенным классом, когда оно может быть получено из классов, порожденных семействами, с помощью операций пересечения и вычитания, и никакое непустое множество, представимое таким образом, не является его собственной частью.

ПП. 13. Показать, что для любого лексически размеченного языка (V, L, Γ):

а) L-производная эквивалентность для отношения «принадлежать одному приведенному классу» совпадает с $T_{L,\Gamma}$ [Ревзин 1967, стр. 159]);

б) это отношение является укрупнением S_L тогда и только тогда, когда (V, L, Γ) однороден [Ревзин 1967, стр. 160].

БИБЛИОГРАФИЧЕСКИЕ ЗАМЕЧАНИЯ

К главе 1.

Понятие порождающей грамматики введено Н. Хомским [Chomsky 1956, 1957, 1959(а) *), 1963]. (Родственное понятие «полутуэвской системы» рассматривалось в теории алгоритмов и раньше — см., например, [Kleene 1952] **).) Им же введены понятия НС-грамматики, Б-грамматики и А-грамматики (см. указанные выше работы; Н. Хомский пользуется терминами «грамматики класса 1» для НС-грамматик, «грамматики класса 2» для Б-грамматик и «грамматики класса 3» для А-грамматик). Регулярные выражения введены С. Клини [Kleene 1956].

К главе 2.

Сигнализирующие функции детерминированных вычислений введены в 50-х гг. Г. С. Цейтиным (не опубликовано; см. [Яновская 1959, стр. 44, Трахтенброт 1967, стр. 9]) и независимо Б. А. Трахтенбротом [Трахтенброт 1956]. Общая теория сигнализирующих функций детерминированных вычислений — перенос ее идей на случай грамматик составляет основную часть содержания § 2.1 — построена М. Блюмом [Blum 1967]. Временна́я сложность и емкость для грамматик были введены — под влиянием идей Б. А. Трахтенброта — в [Гладкий 1964(в), 1966]. Понятия левой и правой глубины восходят к работе В. Ингве [Yngve 1960], понятие активной емкости — к работам Мэри Кэтрин Интемы [Yntema 1967] и Б. Брейнерда [Brainerd 1967]; в общем виде последнее понятие сформулировано Э. Д. Стоцким [Стоцкий 1969]. Понятие разброса введено в [Гладкий — Диковский 1970]. Лемма о сцеплении доказана для НС-грамматик в [Гладкий 1964(в)] и распространена на общий случай в [Book 1969]. Теорема 2.1 доказана для частного случая в [Гладкий 1963(в)]. Теорема 2.2 принадлежит Р. Буку [Book 1969]. Теоремы 2.4 и 2.5 — это известные теоремы Г. С. Цейтина (см. [Яновская 1959, стр. 45, Трахтенброт 1967, стр. 152]) и М. Рабина [Rabin 1959 ***), Трахтенброт 1967, стр. 194] из теории сложности вычислений, пере-

*) Эта работа содержит неточности в определениях, впоследствии замеченные и исправленные К. Чуликом [Culik 1965].

**) В русском переводе этой книги — «полусистема Туэ» (стр. 340).

***) В этой работе нет доказательства.

несенные на случай грамматик. Приводимые нами доказательства этих теорем по существу заимствованы из книги [Трахтенброт 1967]. Изучению временной сложности грамматик посвящена книга [Book 1969].

К главе 3.

Понятие дерева вывода восходит к основополагающим работам Н. Хомского [Chomsky 1956, 1957, 1959(а), 1963]. Ему же принадлежат понятие неукорачивающей грамматики и теорема 3.1 [Chomsky 1959(а)]. Теорема 3.2, в несколько иной форме, доказана в [Kuroda 1964]. Другое доказательство существования для всякой грамматики без растяжения эквивалентной ей НС-грамматики (положенное в основу доказательства теоремы 3.5) было дано в [Гладкий 1963(в)]. Теорема 3.3 отмечена в [Гладкий 1966]. (Эффективная) замкнутость класса НС-языков относительно пересечения доказана независимо в трех работах [Landweber 1963, Гладкий 1963(в), Kuroda 1964]. Теорема 3.7 доказана в [Гладкий 1964(в)]. Использованная в ее доказательстве техника следов независимо разработана — применительно к детерминированным машинам Тьюринга — Б. А. Трахтенбротом [Трахтенброт 1964], Я. М. Барздинем [Барздинь 1965], М. Рабином [Rabin 1963], Ф. Хенни [Hennie 1965]. Лемма 3.2 принадлежит фактически Б. А. Трахтенброту [Трахтенброт 1964]. Пример, аналогичный примеру 1 из § 3.4, был впервые построен Р. Парикхом [Parikh 1961]. Примеры односторонних НС-грамматик, выводящих за пределы класса Б-языков, были указаны независимо Л. Хейнсом [Haines 1965]*), Л. Г. Самойленко [Самойленко 1968] и И. Гавелом [Havel 1969]. Теорема 3.8 доказана Л. Хейнсом [Haines 1969, 1970].

К главе 4.

Основные факты теории Б-языков были впервые систематизированы — и многие из них впервые получены — в статье [Bar-Hillel — Perles — Shamir 1961]. В ней содержится практически все изложенное в §§ 4.1, 4.2, а также теорема 4.5 (и некоторые другие результаты, которые будут отмечены ниже). Большую роль в развитии теории Б-языков сыграла работа Р. Парикха [Parikh 1961]**), в которой введено понятие однозначности языка и указан первый пример неоднозначного бесконтекстного языка (пример 3 из § 4.4); там же доказана теорема 4.6. Теорема 4.7 доказана в [Гладкий 1965(б)]. Теорема 4.8 впервые опубликована, видимо, в [Scheinberg 1960]; она имеется также в [Bar-Hillel — Perles — Shamir 1961]. Неоднозначность Б-языков изучалась в [Ginsburg — Ullian 1966], где для важного в приложениях случая ограниченных языков (см. упражнение 5.32) получено необходимое и достаточное условие неоднозначности (соответствующие результаты изложены также в [Ginsburg 1966, гл. 6]). Понятие МП-машины введено Н. Хомским [Chomsky

*) О существовании примера, построенного в диссертации [Haines 1965], автору стало известно из рукописи [Haines 1970], любезно присланной ему Л. Хейнсом в конце 1970 г.

**) Препринт, впоследствии перепечатанный в [Parikh 1966].

1962]*); им же доказана теорема 4.9 [Chomsky 1962, 1963]. Впоследствии эта теорема несколько раз передоказывалась, в частности Р. Эви [Evey 1963] и Г. С. Цейтиным (не опубликовано). Конструкции, используемые для ее доказательства в настоящей книге (леммы 4.12—4.14), навеяны идеями Ю. А. Шрейдера, В. Б. Борщева и М. В. Хомякова [Шрейдер 1969, Борщев — Хомяков 1970] и Л. С. Модиной [Модина 1970]. Детерминированные МП-машины впервые рассмотрел М. Шютценберже [Schützenberger 1963]; они изучались также С. Гинзбургом и Ш. Грейбах [Ginsburg — Greibach 1966(б)] и А. Я. Диковским [Диковский 1969]. Теория Б-языков посвящена книга [Ginsburg 1966].

К главе 5.

Конечные автоматы в явной форме подвергаются систематическому изучению с 50-х гг. (их называют также «последовательностными машинами», «автоматами Мура-Мили»); однако в неявном виде они рассматривались и раньше в теории алгоритмов (головка машины Тьюринга по существу представляет собой конечный автомат; см. [Turing 1936—1937]) и в теории информации (см., например, [Shannon 1948]**)). Теорема 5.3 в неявном виде установлена Ю. Т. Медведевым [Медведев 1956]; в явной форме впервые опубликована в [Rabin — Scott 1959]. Регулярные языки введены в [Kleene 1956]; там же отмечена (эффективная) замкнутость класса регулярных языков относительно основных операций и доказана теорема, по существу совпадающая с теоремой 5.6. Теорема 5.7 впервые появилась, по-видимому, в [Myhill 1957] и [Nerode 1958]. Изучению ОА-грамматик посвящена работа [Chomsky — Miller 1958]. Линейные грамматики введены в [Schützenberger 1961], металинейные — либо в той же работе***), либо в [Chomsky 1963] и [Chomsky — Schützenberger 1963]. МП-машины с ограниченным числом поворотов и ОАЕВ-грамматики изучались в [Ginsburg — Spanier 1966]; в этой работе доказаны теоремы 5.11 и 5.12.

К главе 6.

Категориальные грамматики появились значительно раньше всех других типов формальных грамматик; они были введены в работах С. Лесьневского [Leśniewski 1929] и К. Айдукевича [Ajdukiewicz 1935] и предназначались для описания структуры формальных языков математики. Использовать категориальные грамматики для моделирования естественных языков предложил И. Бар-Хиллел [Bar-Hillel 1953]; он же поставил вопрос о взаимоотношении между категориальными грамматиками и Б-грамматиками, который был ре-

*) Сходные конструкции использовались ранее, на неформальном уровне, в теории программирования — см., например, [Burks — Warren — Wright 1954, Samelson — Bauer 1960].

**) Стр. 313 русского перевода.

***) Автор не имел возможности с ней ознакомиться; приводимые здесь сведения о ней заимствованы из [Gross — Lentin 1967].

шен Х. Гайфманом [Bar-Hillel — Gaifman — Shamir 1960] *). Позднее другое доказательство теоремы Х. Гайфмана об эквивалентности категориальных грамматик и Б-грамматик было дано А. Я. Диковским и Л. С. Модиной [Диковский—Модина 1968] **). В настоящей книге излагается (с рядом модификаций) первоначальное доказательство Х. Гайфмана ***). Теорема о нормальной форме доказана Шейлой Грейбах ([Greibach 1965]; другое, более простое доказательство см. [Greibach 1967]). Простые доминационные грамматики (грамматики зависимостей) введены Д. Хейсом [Hays 1961]. Их связь с Б-грамматиками была изучена в работе Х. Гайфмана [Gaifman 1961] ****), где доказана теорема, близкая к теореме 6.5. В приводимой здесь форме эта теорема дана С. Я. Фитиаловым [Фитиалов 1968]; им же введено понятие степени грамматики (и степени системы составляющих). Доминационные грамматики общего вида введены М. И. Белецким [Белецкий 1967]. Задание Б-языков с помощью нормальных систем уравнений предложено С. Гинзбургом и Г. Райсом [Ginsburg — Rice 1962], с помощью формальных степенных рядов — М. Шютценберже [Schutzenberger 1959, Chomsky — Schützenberger 1963]. Теорема 6.7 впервые опубликована, кажется, в [Chomsky — Schützenberger 1963].

К главе 7.

Теоремы 7.1 и 7.2 в неявном виде фактически содержатся в [Yngve 1960]; в явной форме теорема 7.1 впервые изложена, видимо, в [Шрейдер 1966]. Теорема 7.4 принадлежит Э. Д. Стоцкому [Стоцкий 1969]. Теорема 7.5 доказана С. Гинзбургом и Э. Спеньером [Ginsburg — Spanier 1968]; в этой же работе впервые указаны примеры Б-языков сколь угодно большой активной емкости и Б-языка, не являющегося ОАЕ-языком (при этом существенным образом использованы результаты работы [Yntema 1967], где, в частности, построен пример Б-языка, не принадлежащего замыканию класса линейных языков относительно подстановки). Те же результаты получены независимо в [Диковский 1972 (а, б)] другим способом — с помощью введенного в этих же работах понятия густоты (там же доказана лемма 7.2). Приводимое нами доказательство теоремы 7.5 воспроизводит во всех основных чертах рассуждения из работ А. Я. Диковского; оттуда же взят (с некоторым видоизменением) пример 1 в § 7.2. Понятие самовставления и лемма 7.3 имеются уже в [Chomsky 1959 (а, б)]. Теорема 7.6 (доказанная автором в процессе работы над книгой) представляет собой ответ на вопрос, поставленный С. Я. Фитиаловым.

*) В этой же работе, видимо, впервые появилось определение категориальной грамматики в принятой сейчас форме, а также самый термин «категориальная грамматика».

**) В 1966 г. Х. Гайфман в устной беседе сообщил автору, что эта теорема была доказана также Н. Хомским. Однако в литературе упоминаний о доказательстве Н. Хомского обнаружить не удалось.

***) Доказательство А. Я. Диковского и Л. С. Модиной существенно проще, но опирается на теорему о нормальной форме. Автор счел более естественным обратный порядок изложения.

****) Препринт, перепечатанный впоследствии в [Gaifman 1965].

К главе 8.

Проблема распознавания самоприменимости машины Тьюринга явилась одной из первых алгоритмических проблем, для которых была доказана неразрешимость [Turing 1936—1937]. Теорема Райса также представляет собой один из классических результатов теории алгоритмов [Rice 1953]. Доказательства нераспознаваемости в классе НС-грамматик пустоты и конечности языка, а также неразрешимости проблемы эквивалентности НС-грамматик приведены в [Chomsky 1963] (со ссылкой на неопубликованную заметку С. Шайнберга). Нераспознаваемость бесконтекстности в классе НС-грамматик указана Э. Шамиром [Shamir 1962]. Теорема 8.3 доказана в [Гладкий 1964 (б)]. Нераспознаваемость в классе ОБ-грамматик свойств языка иметь пустое и конечное дополнение и быть ОА-языком, а также неразрешимость проблемы эквивалентности Б-грамматик установлены в [Bar-Hillel — Perles — Shamir 1961] с помощью теоремы Поста (см. упражнение 8.14). Техника протоколов (лемма 8.5) предложена в [Гладкий 1965 (а)] (и независимо в [Hartmanis 1967]). Нераспознаваемость неоднозначности Б-языков доказана независимо в [Гладкий 1965 (б)] и [Ginsburg — Ullian 1966] *). Теорема 8.5 установлена (в несколько иной форме) Ш. Грейбах [Greibach 1968]. Неразрешимость для бесконтекстных и линейных языков проблем распознавания замещаемости, взаимозамещаемости и конфигураций доказана в [Гладкий 1965 (а)].

К приложению I.

Понятие системы составляющих восходит к грамматической традиции английского языка и американскому структурализму («анализ по непосредственным составляющим»; см. [Bloomfield 1933], [Harris 1961] и в особенности [Wells 1947]). Понятия степени левого и правого ветвления идут от работы В. Ингве [Yngve 1960]; строго формально эти понятия впервые введены, видимо, в [Bar-Hillel—Kasher—Shamir 1963]. Понятие степени гнездования введено Н. Хомским [Chomsky 1961]; мы пользуемся этим понятием в видоизмененной форме, данной в [Bar-Hillel — Kasher — Shamir 1963] (эта форма не равносильна первоначальной). Понятие дерева синтаксического подчинения восходит к русской грамматической традиции; в явном виде оно впервые появилось (как часть некоторого более сложного понятия) у Л. Теньера [Tesniére 1959]. Идеи, приведшие к понятиям проективности и слабой проективности, впервые были — в неявной форме — высказаны Г. С. Цейтиным (см. [Цейтин — Засорина 1961]) и независимо К. Харпером и Д. Хейсом [Harper — Hays 1960]. В явном виде эти понятия ввели, независимо друг от друга, И. Лесерф [Lecerf 1960], М. И. Белецкий [Белецкий 1961] и С. Я. Фитиалов [Фитиалов 1962]. (Общепринятые сейчас формулировки, приводимые и в настоящей книге, принадлежат С. Я. Фитиалову.) Проективность изучалась с формальной и лингвистической точек зрения, в частности, в [Белецкий — Григорян — Заславский 1963],

*) Содержание этой статьи изложено также в книге [Ginsburg 1966] (гл. 6).

[Дрейзин 1963], [Иорданская 1967]. Описанный в § ПІ. 3 способ установления соответствия между системами составляющих и деревьями подчинения указан в [Падучева 1964] (в неформальном виде; формализованное изложение — в [Гладкий 1966]).

К приложению II.

Начало теории замещаемости было положено работой [Кулагина 1958]. В этой работе было, в частности, введено понятие конфигурации. Определение конфигурации, изложенное в настоящей книге, — основанное на тех же идеях, что и первоначальное определение О. С. Кулагиной, но не равносильное ему, — дано в [Гладкий 1963(а)]; там же введены понятия конфигурационных характеристик и конечно характеризуемого языка и получены теоремы ПІІ. 1 и ПІІ. 2 *). Еще одно видоизменение понятия конфигурации было предложено М. Новотным [Novotný 1965(а, б, в)]. Лингвистическая критика конфигурационной модели имеется в [Падучева 1965], стимулированное в значительной степени этой критикой распространение понятия конфигурации на «языки деревьев» — в [Щербакова 1971]. Все основные понятия § ПІІ. 3 введены в [Кулагина 1958]; там же доказаны теоремы ПІІ. 3—ПІІ. 6. Алгебраическое изложение этих понятий и результатов было дано Д. Н. Ленским [Ленской 1962] **) и М. Новотным [Новотный 1965]. Приведенные классы были введены (под именем «подклассов») И. И. Ревзиным [Ревзин 1962, 1967]. Анализирующим моделям посвящена книга [Marcus 1967], их лингвистическим приложениям — [Ревзин 1967].

*) При понимании конфигураций в смысле первоначального определения эти теоремы не имеют места.

**) На языке алгебры бинарных отношений (см. также [Шрейдер 1970]).

ЛИТЕРАТУРА

Барздинь Я. М. Сложность распознавания симметрии на машинах Тьюринга. — Сб. «Проблемы кибернетики», вып. 15, «Наука», 1965, 245—248.

Белецкий М. И. Модель языка алгоритма грамматического анализа. — Сб. «Тезисы Конференции по обработке информации, машинному переводу и автоматическому чтению текста», ВИНИТИ, 1961, 12.

Белецкий М. И. Бесконтекстные и доминационные грамматики и связанные с ними алгоритмические проблемы. — Кибернетика, 1967, № 4, 90—97.

Белецкий М. И., Григорян В. М., Заславский И. Д. Аксиоматическое описание порядка и управления слов в некоторых типах предложений. — Сб. «Математические вопросы кибернетики и вычислительной техники», № 1, Ереван, 1963, 71—85.

Борщев В. Б., Хомяков М. В. Окрестностные грамматики и модели перевода. Часть I. Окрестностные грамматики. — Научно-техническая информация, сер. 2, 1970, № 3, 39—44.

Гладкий А. В. Конфигурационные характеристики языков. — Сб. «Проблемы кибернетики», вып. 10, Физматгиз, 1963(а), 251—260.

Гладкий А. В. О распознавании замещаемости в рекурсивных языках. — Алгебра и логика, 1963 (б), **2**, № 3, 5—22.

Гладкий А. В. Грамматики с линейной памятью. — Алгебра и логика, 1963(в), **2**, № 5, 43—55.

Гладкий А. В. Алгоритм распознавания конфигураций для класса автоматных языков. — Сб. «Проблемы кибернетики», вып. 12, «Наука», 1964(а), 243—245.

Гладкий А. В. Алгоритмическая природа инвариантных свойств грамматик непосредственно составляющих. — Алгебра и логика, 1964(б), **3**, № 2, 17—32.

Гладкий А. В. О сложности вывода в грамматиках непосредственно составляющих. — Алгебра и логика, 1964(в), **3**, № 5—6, 29—44.

Гладкий А. В. Некоторые алгоритмические проблемы для КС-грамматик. — Алгебра и логика, 1965(а), **4**, № 1, 3—13.

Гладкий А. В. Алгоритмическая нераспознаваемость существенной неопределенности КС-языков. — Алгебра и логика, 1965(б), **4**, № 4, 53—64.

Гладкий А. В. Лекции по математической лингвистике для студентов НГУ, изд. Новосибирского гос. университета, 1966.

Гладкий А. В. Об описании синтаксической структуры предложения. — Computational linguistics, **VII**. Budapest, 1969, 21—44.

Гладкий А. В. Описание синтаксической структуры предложения с помощью систем синтаксических групп. 1. Формальный аппарат. — Научно-техническая информация, сер. 2, 1971, № 9, 35—38.

Гладкий А. В., Диковский А. Я. Теория формальных грамматик. — Труды 2-й Всесоюзной конференции по программированию, Приглашенные доклады (1-й вып.), Новосибирск, 1970, 43—70.

Гладкий А. В., Мельчук И. А. Элементы математической лингвистики, «Наука», 1969.

Гладкий А. В., Мельчук И. А. Грамматики деревьев. 1. Опыт формализации преобразований синтаксических структур естественного языка. — Сб. «Информационные вопросы семиотики, лингвистики и автоматического перевода», вып. 1, 1971, 16—41.

Диковский А. Я. О соотношении между классом всех контекстно-свободных языков и классом детерминированных контекстно-свободных языков. — Алгебра и логика, 1969, 8, № 1, 44—64.

Диковский А. Я. Замечание о детерминированных линейных языках. — Сб. «Проблемы кибернетики», вып. 23, «Наука», 1970, 281—286.

Диковский А. Я. Густота — мера сложности вывода в контекстно-свободной грамматике. — Проблемы передачи информации, 1972(а), 8, вып. 2, 92—105.

Диковский А. Я. Густота дерева вывода и активная емкость грамматики. — Проблемы передачи информации, 1972(б), 8, вып. 4.

Диковский А. Я., Модина Л. С. Минимизация одной функции сложности в классе МП-автоматов и категориальные грамматики сложности 3. — Алгебра и логика, 1968, 7, № 3, 23—37.

Дрейзин Ф. А. Некоторые свойства связей между словами в предложении. — Научные труды Ташкентского университета, вып. 228, 1963, 33—39.

Жолковский А. К., Мельчук И. А. О семантическом синтезе. — Сб. «Проблемы кибернетики», вып. 19, «Наука», 1967, 177—238.

Зализняк А. А. Русское именное словоизменение, «Наука», 1967.

Иорданская Л. Н. Автоматический синтаксический анализ, II, «Наука», 1967.

Кулагина О. С. Об одном способе определения грамматических понятий на базе теории множеств. — Сб. «Проблемы кибернетики», вып. 1, Физматгиз, 1958, 203—214.

Ленской Д. Н. Об одном применении алгебры бинарных отношений в математической лингвистике. — Уч. зап. Кабардино-Балкарского гос. ун-та, вып. 16, сер. физ.-мат., 1962, 73—80.

Лучкин В. Д. О рангах конфигураций контекстно-свободных языков. — Алгебра и логика, 1966, 5, № 3, 59—70.

Марков А. А. Теория алгорифмов. — Труды Математического ин-та АН СССР им. В. А. Стеклова, XLII, Изд-во АН СССР, 1954.

Мартыненко Г. Я. Статистическое исследование синтаксической сложности предложения (на материале болгарского языка). — Сб. «Информационные вопросы семиотики, лингвистики и автоматического перевода», вып. 1, 1971, 84—101.

Медведев Ю. Т. О классе событий, допускающих представление в конечном автомате. — Сб. «Автоматы», ИЛ, 1956, 385—401.

Модина Л. С. Свертки языков деревьев. — Сб. «Структурно-математические методы моделирования языка. Тезисы докладов и сообщений Всесоюзной научной конференции», ч. II, Киев, 1970, 91—92.

Новотный М. Об алгебраизации теоретико-множественной модели языка. — Сб. «Проблемы кибернетики», вып. 15, «Наука», 1965, 234—243.

Падучева Е. В. О способах представления синтаксической структуры предложения. — Вопросы языкознания, 1964, № 2, 99—113.

Падучева Е. В. О понятии конфигурации. — Вопросы языкознания, 1965, № 1, 56—68.

Падучева Е. В. О связях глубины по Ингве со структурой дерева подчинения. — Научно-техническая информация, 1966, № 6, 38—43.

Ревзин И. И. Модели языка, Изд-во АН СССР, 1962.

Ревзин И. И. Метод моделирования и типология славянских языков, «Наука», 1967.

Самойленко Л. Г. Об одном классе грамматик непосредственно составляющих. — Кибернетика, 1968, № 2, 102—103.

Стоцкий Э. Д., О некоторых ограничениях на способ вывода в грамматиках непосредственных составляющих. — Научно-техническая информация, сер. 2, 1967, № 7, 35—38.

Стоцкий Э. Д. Понятие индекса в обобщенных грамматиках. — Научно-техническая информация, сер. 2, 1969, № 5, 16—17.

Трахтенброт Б. А. Сигнализирующие функции и табличные операторы. — Уч. зап. Пензенского гос. пед. ин-та, IV, 1956, 75—87.

Трахтенброт Б. А. Тьюринговы вычисления с логарифмическим замедлением. — Алгебра и логика, 1964, 3, № 4, 33—48.

Трахтенброт Б. А. Сложность алгоритмов и вычислений, изд. Новосибирского гос. университета, 1967.

Успенский В. А. К определению части речи в теоретико-множественной системе языка. — Бюллетень Объединения по проблемам машинного перевода, 1957, № 5, 22—26.

Фитиалов С. Я. О моделировании синтаксиса в структурной лингвистике. — Сб. «Проблемы структурной лингвистики», Изд-во АН СССР, 1962, 100—114.

Фитиалов С. Я. Об эквивалентности грамматик НС и грамматик зависимостей. — Сб. «Проблемы структурной лингвистики, 1967», «Наука», 1968, 71—102.

Цейтин Г. С. К вопросу о построении математических моделей языка. — Сб. «Тезисы совещания по математической лингвистике», Л., 1959, 45.

Цейтин Г. С., Засорина Л. Н. О выделении конфигураций в русском предложении. — Доклады конференции по обработке информации, машинному переводу и автоматическому чтению текста, ВИНИТИ, 1961, вып. 2.

Шрейдер Ю. А. Характеристики сложности структуры текста. — Научно-техническая информация, 1966, № 7, 34—39.

Шрейдер Ю. А. Окрестностная модель языка. — Tartu Ülikooli toimetised, Уч. зап. Тартуского ун-та, вып. 226, 1969, 129—142.

Шрейдер Ю. А. Алгебра бинарных отношений. — В кн. С. Маркус, Теоретико-множественные модели языков, «Наука», 1970, 300—330.

Щербакова Н. Г. Конфигурации в языках деревьев. — Научно-техническая информация, сер. 2, 1971, № 6, 28—32.

Яновская С. А. Математическая логика и основания математики. — Сб. «Математика в СССР за 40 лет», т. I, Физматгиз, 1959, 13—120.

Ajdukiewicz K. Die syntaktische Konnexität. — Studia Philosophica, 1935, № 1, 1—27.

Bar-Hillel Y. A. quasi-arithmetical notation for syntactic description. — Language, 1953, 29, № 1, 47—58 *).

Bar-Hillel Y., Gaifman C., Shamir E. On categorial and phrase structure grammars. — Bull. Res. Council Israel, 1960, 9F, 155—166 *).

Bar-Hillel Y., Kasher A., Shamir E. Measures of syntactic complexity. — The Hebrew University of Jerusalem, Applied Logic Branch, Technical Rept, 1963, № 13, August 1963. (Русский перевод: Бар-Хиллел И., Кашер А., Шамир Е., Меры синтаксической сложности, Кибернетический сборник, нов. серия, вып. 4, «Мир», 1967, 219—227.)

Bar-Hillel Y., Perles M., Shamir E. On formal properties of simple phrase structure grammars. — Z. Phonetik, Sprachwiss. Kommunikationsforsch., 1961, 14, № 2, 143—172 *).

Bloomfield L., Language, N. Y., 1933. (Русский перевод: Блумфилд Л., Язык, «Прогресс», 1968.)

Blum M. A machine-independent theory of the complexity of recursive functions. — J. Assoc. Computing Machinery, 1967, 14, № 2, 322—337. (Русский перевод: Блюм М., Машинно-независимая теория сложности рекурсивных функций, Сб. «Проблемы математической логики», «Мир», 1970, 401—422.)

Book R. V. Grammars with time functions. — Mathematical Linguistics and Automatic Translation, Harvard Univ. Computation Lab., Rept. NSF-23, February, 1969.

Brainerd B. An analog of a theorem about context-free languages. — Information and Control, 1967, 11, № 5—6, 561—567.

Büchi J. Regular canonical systems. — Arch. math. Logik Grundlagenforsch., 1964, 6, 91—111.

Burks A. W., Warren D. W., Wright J. B. An analysis of a logical machine using parenthesis-free notation. — Math. Tables and Other Aids of Computation, 1954, 8, 53—57.

Chomsky N. Three models for the description of language. — IRE Trans. of Inform. Theory, 1956, IT-2, 113—124. (Русский перевод: Хомский Н., Три модели для описания языка, Кибернетический сборник, вып. 2, ИЛ, 1961, 237—266.)

*) Работы Bar-Hillel 1953, Bar-Hillel — Gaifman — Shamir 1960, Bar-Hillel — Perles — Shamir 1961 перепечатаны в книге Bar-Hillel Y., Language and Information, Reading (Mass.), 1964.

Chomsky N. Syntactic structures. — s'Gravenhage, 1957. (Русский перевод: Хомский Н., Синтаксические структуры, сб. «Новое в лингвистике» вып. II, ИЛ, 1962, 412—527.)

Chomsky N. On the certain formal properties of grammars. — Information and Control, 1959 (a), 2, № 2, 137—167. (Русский перевод: Хомский Н., О некоторых формальных свойствах грамматик, Кибернетический сборник, вып. 5, ИЛ, 1962, 279—311.)

Chomsky N. A note on phrase structure grammars. — Information and Control, 1959 (б), 2, № 4, 393—395. (Русский перевод: Хомский Н., Заметка о грамматиках непосредственно составляющих, Кибернетический сборник, вып. 5, ИЛ, 1962, 312—316.)

Chomsky N. On the notion «rule of grammar». — In «Structure of language and its mathematical aspects», Providence (Rhode Island), 1961, 6—24. (Русский перевод: Хомский Н., О понятии «правило грамматики», сб. «Новое в лингвистике», вып. IV, «Прогресс», 1965, 34—65.)

Chomsky N. Context-free grammars and pushdown storage. — M. I. T. Res. Lab. Electron. Quart. Progr. Rept., 1962, № 65.

Chomsky N. Formal properties of grammars. — In: R. R. Bush, E. H. Galanter, R. D. Luce (eds.), Handbook of Math. Psychology, 2, N. Y., 1963, 323—418. (Русский перевод: Хомский Н., Формальные свойства грамматик, Кибернетический сборник, новая серия, вып. 2, «Мир», 1966, 121—230.)

Chomsky N., Miller G. A. Finite state languages. — Information and Control, 1958, 1, № 2, 91—112. (Русский перевод: Хомский Н., Миллер Дж., Языки с конечным числом состояний, Кибернетический сборник, вып. 4, ИЛ, 1962, 233—255.)

Chomsky N., Schützenberger M. P. The algebraic theory of context-free languages. — In: P. Braffort, D. Hirschberg (eds.), Computer programming and formal systems, Amsterdam, 1963, 118—161. (Русский перевод: Хомский Н., Шютценберже М. П., Алгебраическая теория контекстно-свободных языков, Кибернетический сборник, новая серия, вып. 3, «Мир», 1966, 195—242.)

Culik K. Axiomatic system for phrase structure grammars, I. — Information and Control, 1965, 8, № 5, 493—502.

Evey R. J. The theory and applications of pushdown store machines. — Mathematical Linguistics and Automatic Translation, Harvard Univ. Computation Lab., Rept. NSF-10, May 1963.

Friš I., On stop-conditions of constructive languages. — Z. Math. Logik Grundl. Math., 1965, 11, № 1, 61—73.

Gaifman H. Dependency systems and phrase structure systems. — Santa Monica (Calif.), RAND Corp., 1961 (P-2315).

Gaifman H. Dependency systems and phrase structure systems. — Information and Control, 1965, 8, № 3, 304—337.

Ginsburg S., The mathematical theory of context-free languages, — N. Y. — San Francisco — Toronto — London, 1966. (Русский перевод: Гинзбург С., Математическая теория контекстно-свободных языков, «Мир», 1970.)

Ginsburg S., Greibach S. A. Mappings which preserve context-sensitive languages. — Information and Control, 1966 (a), 9, № 6, 563—583. (Русский перевод: Гинзбург С., Грейбах Ш., Об инвариантности класса НС-языков относительно некоторых

отображений, Кибернетический сборник, новая серия, вып. 5, «Мир», 1968, 167—188.)

Ginsburg S., Greibach S. A. Deterministic context-free languages. — Information and Control, 1966 (б), **9**, № 6, 620—648.

Ginsburg S., Rice H. G. Two families of languages related to ALGOL. — J. Assoc. Computing Machinery, 1962, **9**, № 3, 350—371. (Русский перевод: Гинзбург С., Райс Х., Два класса языков типа АЛГОЛ, Кибернетический сборник, новая серия, вып. 6, «Мир», 1969.)

Ginsburg S., Spanier E. H., Finite-turn pushdown automata. — J. SIAM Control, 1966, **4**, № 3, 429—453.

Ginsburg S., Spanier E. H. Derivation-bounded languages. — J. of Computer and System Sciences, 1968, **2**, № 3, 228—250.

Ginsburg S., Ullian J. S. Ambiguity in context-free languages. — J. Assoc. Computing Machinery, 1966, **13**, № 3, 364—368.

Greibach S. A. Undecidability of the ambiguity problem for minimal linear grammars. — Information and Control, 1963, **6**, № 2, 119—125.

Greibach S. A. A new normal-form theorem for context-free phrase structure grammars. — J. Assoc. Computing Machinery, 1965, **12**, № 1, 42—52.

Greibach S. A. The unsolvability of the recognition of linear context-free languages. — J. Assoc. Computing Machinery, 1966, **13**, № 4, 582—587.

Greibach S. A. A simple proof of the standard-form theorem for context-free grammars. — In: «Mathematical Linguistics and Automatic Translation», Harvard Univ. Computation Lab., Rept. NSF-18, August, 1967, II-1—II-4.

Greibach S. A. A note on undecidable properties of formal languages. — Math. Systems Theory, 1968, 2, № 1, 1—6.

Gross M., Lentin A. Notions sur les grammaires formelles. — Paris, 1967. (Русский перевод: Гросс М., Лантен А., Теория формальных грамматик, «Мир», 1971.)

Gruska J. On a classification of context-free languages. — Kybernetika, 1967, **3**, № 1, 22—29.

Haines L. H. Generation and recognition of formal languages. — Ph. D. Thesis, Massachusets Institute of Technology, 1965.

Haines L. H. Representation theorems for context-sensitive languages. — Notices Amer. Math. Soc. 1969, **16**, № 3, 527.

Haines L. H. Representation theorems for context-sensitive languages. — Univ. of California, Berkly, Calif., 1970.

Harper K. E., Hays D. G. The use of machines in the construction of a grammar and computer program for structural analysis. — In «IFIP. Information processing 1959, Proceedings...», Paris, 1960, 188—194. (Русский перевод: Харпер К. Е., Хейс Д. Г., Использование машин при построении грамматики и программа для структурного анализа, сб. «Автоматизация в лингвистике», «Наука», 1966, 132—143.)

Harris Z.S. Structural linguistics. — Chicago, 1961.

Hays D. G. Grouping and dependency theories. — In «Proceedings of the National Symposium on Machine Translation», Englewood Cliffs (N. J.), 1961, 258—266.

Hartmanis J. Context-free languages and Turing machine computations. — Proc. Symposium in Applied Mathematics of the Amer. Math. Soc., 1967, 42—51.

Havel I. A note on one-sided context-sensitive grammars. — Kybernetika, 1969, **5**, № 3, 186—189.

Hennie F. C. One-tape, off-line Turing machine computations. — Information and Control, 1965, 8, № 6, 553—578. (Русский перевод: Хенни Ф. К., Вычисления на одноленточной машине Тьюринга с записью на ленте, сб. «Проблемы математической логики», «Мир», 1970, 223—248.)

Hopcroft J. E., Ullman J. D. Formal languages and their relations to automata. — Menlo Park, Calif., 1969.

Kleene S. C. Introduction to Metamathematics. — Princeton, N. J., 1952. (Русский перевод: Клини С. К., Введение в метаматематику, ИЛ, 1957.)

Kleene S. C. Representation of events in nerve sets and finite automata. — In: C. E. Shannon, J. McGarthy (eds.), Automata Studies, Princeton, N. J., 1956, 3—41. (Русский перевод: Клини С. К., Представление событий в нервных сетях и конечных автоматах, сб. «Автоматы», ИЛ, 1956, 15—67.)

Kuroda S.-Y. Classes of languages and linear-bounded automata. — Information and Control, 1964, **7**, № 2, 207—223.

Landweber P. S. Three theorems on pharse structure grammars of type 1. — Information and Control, 1963, **6**, № 2, 131—136.

Lecerf Y. Programme des conflits, modèle des conflits. — Traduction automatique, 1960, **1**, № 4, 11—18; № 5, 17—36.

Leśniewski S. Grundzüge eines neuen Systems der Grundlagen der Mathematik. — Fundam. Math., 1929, **14**, 1—81.

Marcus S. Algebraic linguistics; analytical models. — N. Y. — L., 1967. (Русский перевод: Маркус С., Теоретико-множественные модели языков, «Наука», 1970.)

Myhill J. Finite automata and the representation of events. — WADC Technical Report 57—624, 1957, 112—137.

Nerode A. Linear automaton transformations. — Proc. Amer. Math. Soc., 1958, **9**, № 4, 541—544.

Novotný M. Über endlich characterisierbare Sprachen. — Spisy připodověd. fak. univ. v Brně, 1965 (а), № 10, 495—502.

Novotný M. Bemerkung über ableitbare Sprachen. — Spisy přirodověd. fak. univ. v Brně, 1965 (б), № 10, 503—507.

Novotný M. Bemerkungen über Homorphismen von Sprachen. — Spisy přirodověd. fak. univ. v Brně, 1965 (в), № 10, 509—518.

Parikh R. J. Language generating devices. — M. I. T. Res. Lab. Electron. Quart. Prog. Rept, 1961, № 60, 199—212.

Parikh R. J. On context-free languages. — J. Assoc. Computing Machinery, 1966, **13**, № 4, 570—580.

Post E. L. A variant of a recursively unsolvable problem. — Bull. Amer. Math. Soc., 1946, **52**, № 4, 264—268.

Rabin M. O. Speed of computation of function and classification of recursive sets. — Bull. Res. Council Israel, 1959, **8F**, 69—70.

Rabin M. O. Real-time computation. — Israel Journal Math., 1963, **1**, № 4, 203—211.

Rabin M. O., Scott D. Finite automata and their decision problems. — IBM Journal Res. and Devel., 1959, **3**, № 2, 114—125. (Русский перевод: Рабин М., Скотт Д., Конечные автоматы и проблемы их разрешения, Кибернетический сборник, вып. 4, ИЛ, 1962, 56—91.)

Rice H. G. Classes of recursive enumerable sets and their decision problems. — Trans. Amer. Math. Soc., 1953, **74**, № 2, 358—366.

Ritchie R. W. Classes of predictible computable functions. — Trans. Amer. Math. Soc., 1963, **106**, № 1, 139—173. (Русский перевод: Ричи Р. В., Классы предсказуемо вычислимых функций, сб. «Проблемы математической логики», «Мир», 1970, 50—93.)

Rosenberg A. L. A machine realisation of the linear context-free languages. — Information and Control, 1967, **10**, № 2, 175—188.

Samelson K., Bauer F. L., Sequential formula translation. — Commun. Assoc. Computing Machinery, 1960, **3**, № 2, 76—83.

Scheinberg S. Note on the Boolean properties of context-free languages. — Information and Control, 1960, **3**, № 4, 372—375.

Schützenberger M. P. On context-free languages and pushdown automata. — Information and Control, 1963, **6**, № 3, 246—264.

Schützenberger M. P. Une problème de la théorie des automates. — Seminaire Dubreil — Pisot, Paris, 1959.

Schützenberger M. P. Some remarks on Chomsky's context-free languages. — M. I. T. Res. Lab. Electron. Quart. Prog. Rept., 1961, № 63, 155—170.

Shamir E. A remark of discovery algorithms for grammars. — Information and Control. 1962, **5**, № 3, 246—251.

Shannon K. E. A mathematical theory of communication. — Bell System Techn. Journal, 1948, **27**, № 3, 379—423; № 4, 623—656. (Русский перевод: Шеннон К., Математическая теория связи, сб. «Работы по теории информации и кибернетике», ИЛ, 1963, 249—332.)

Tesnière L. Éléments de syntaxe structurale. Paris, 1959.

Turing A. M. On computable numbers, with an application to the Entscheidungsproblem. — Proc. London Math. Soc., ser. 2, 1936—1937, **42**, 230—265. A correction, ibid., 1937, **43**, 544—546.

Ullian J. S. Failure of a conjecture about context-free languages. — Information and Control, 1966, **9**, № 1, 61—65.

Wells R. S. Immediate constituents. — Language, 1947, **23**, № 2, 81—117.

Yngve V. H. A model and a hypothesis for language structure. Proc. Amer. Phil. Soc., 1960, **104**, № 5, 444—466.

Yntema M. K. Inclusion relations among families of context-free languages. — Information and Control, 1967, **10**, № 6, 572—597.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

УКАЗАТЕЛЬ ОБОЗНАЧЕНИЙ