А.В. Гладкий

Лекции

по математической лингвистике для студентов НГУ

Новосибирск 1966.

Дорогому Абраму Ильичу
от автора
А. Гладкий

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

А.В.ГЛАДКИЙ

Л Е К Ц И И
по математической лингвистике
для студентов НГУ

Новосибирск
1966 г.

ОТ АВТОРА

Курс математической лингвистики читается в Новосибирском государственном университете (отделение математической лингвистики) в течение пятого и шестого семестров по два часа в неделю. Настоящий конспект содержит весь материал этого курса.

При составлении конспекта автор пользовался записями лекций, сделанными А.Я.Диковским и Л.С.Модиной. Они же прочли рукопись и сделали много ценных замечаний. Автор выражает им глубокую благодарность.

Автор благодарит также всех, кто помогал готовить конспект к печати.

ВВЕДЕНИЕ

Математическая лингвистика - это специальная математическая дисциплина, возникшая в связи с необходимостью выработки точных методов исследования языка. Она не является частью собственно лингвистики (языкознания); последняя изучает конкретные языки, в то время как содержание первой - выработка и изучение некоторых абстрактных понятий, используемых в собственно лингвистике в качестве моделей различных аспектов языка. Таким образом, соотношение между языкознанием и математической лингвистикой примерно такое же, как, скажем, между физикой и обслуживающими ее математическими дисциплинами. И точно так же, как любая другая математическая дисциплина, связанная с приложениями, математическая лингвистика развивается не только под воздействием потребностей этих приложений, но и под воздействием своих собственных внутренних потребностей, своей "внутренней логики". Таким образом, не все понятия и результаты математической лингвистики имеют непосредственные приложения в языкознании; однако развиваться и получать результаты, имеющие "выход" в собственно лингвистику, математическая лингвистика может лишь как целое, заботясь не только о сохранении связи с лингвистикой, но и о решении задач, возникших внутри нее самой.

Следует заметить, что математическая лингвистика не есть совокупность <u>всех</u> математических методов, применяемых в лингвистике. В математическую лингвистику входит только тот математический аппарат, который является <u>специфическим</u> для лингвистики в том смысле, что он относится к наиболее важным сторонам функционирования языка и разрабатывается специально в связи с приложениями к языку. Те математические методы, для которых лингвистика является лишь одним из возможных приложений, не оказывающим существенного влияния на их характер, к математической лингвистике не принадлежат. В частности, не относятся к математической лингвистике широко применяемые при изучении языка статистические методы.

Математическая лингвистика входит в тот комплекс математических дисциплин, который в настоящее время принято называть "дискретной математикой".

Основой ее аппарата являются методы математической логики и абстрактной алгебры, а также комбинаторные методы. Можно даже рассматривать математическую лингвистику как одно из ответвлений математической логики, тесно переплетающееся с другими ее ответвлениями, прежде всего с теорией алгоритмов и теорией автоматов.

Нужно иметь в виду, что словосочетание "математическая лингвистика" часто употребляется также в совсем другом смысле, а именно как общее обозначение для всех лингвистических работ, в которых используется какой-либо математический аппарат ж). Такое словоупотребление следует признать неудачным, поскольку исследования, в которых применяется математический аппарат, не образуют никакого определенного раздела или направления в науке о языке ; едва ли не в любой области теоретической лингвистики возможны как исследования с использованием математического аппарата, так и "нематематические" работы.

x x

x

Остановимся на предпосылках возникновения математической лингвистики.

Абстрактные модели используются при изучении языка очень давно - с тех пор, как существует грамматика. Ведь такие элементарные понятия "школьной" грамматики, как подлежащее, сказуемое, падеж, род и т.п., предотавляют собой весьма абстрактные конструкции, относящиеси к фактам языка именно как модели. Осознать их абстрактный характер трудно только из-за их привычности. По своему характеру эти понятия довольно близки к математическим жж).

ж) А иногда и вообще для любых нетрадиционных лингвистических работ (в особенности прикладных, например, связанных с автоматическим переводом), даже если в них никакой математический аппарат не используется.

жж) Заметим, что на базе понятий подлежащего и сказуемого было выработано центральное понятие современной математической логики - понятие предиката.

В сущности, создание абстрактных моделей всегда было главным средством теоретического изучения языка. Но значение этого факта было осознано лишь в начале нынешнего столетия, когда в трудах основоположников так называемой структурной лингвистики возникла концепция языка как абстрактной знаковой системы, в которой определяющая роль принадлежит не материальной природе знаков, а отношениям между ними. Эта система сама по себе не наблюдаема, наблюдению поддается только результат ее деятельности-речь; поэтому единственным средством познания законов языка является построение - на основе анализа речи - абстрактных моделей его структуры и изучение строения самих этих моделей.

Из указанной концепции естественно было сделать вывод , что язык следует изучать средствами, близкими к тем, которыми пользуется математика при анализе своих формальных систем ; ведь эти системы, как и язык, характеризуются тем, что в них важны лишь отношения между об"ектами, а не материальная природа последних ; в сущности, математические системы - это специальные разновидности языка, отличающиеся особо четкой структурой. Анализ математических формальных систем является предметом математической логики и отчасти абстрактной алгебры, и естественно поэтому ожидать, что методы изучения языка могут оказаться сходными с методами этих математических дисциплин. Иначе говоря, математика (точнее-некоторый ее раздел или разделы) может оказаться подходящим <u>метаязыком</u> для изучения естественных языков и тем самым сделаться универсальным языком лингвистики. Впоследствии, когда возникла математическая лингвистика, именно так и произошло.

На необходимость математизации лингвистических понятий указывали еще на рубеже XIX и XX столетий такие крупнейшие лингвисты, как Ф. де Соссюр и И.А.Бодуэн де Куртенэ ж). Но только через полвека эта математизация стала осуществляться на деле. Отчасти это может быть об"яснено тем, что лингвисты, как правило, не были знакомы с математикой, а математики - с лингвистикой.Но наиболее существенно, конечно, то, что в начале XX века не было еще теории алгоритмов, идеи которой сыграли важнейшую роль в

ж) См. предисловие Р.Якобсона к книге : Structure of language and its mathematical aspects. Providence, 1961.

формировании понятий математической лингвистики.

Как бы то ни было, к середине 50-х годов определились основные принципы математической трактовки лингвистических понятий, и математическая лингвистика начала бурно развиваться. Ускорению ее развития в значительной степени способствовало то обстоятельство, что как раз тогда в связи с появлением электронных вычислительных машин и быстрым ростом потока научной информации была поставлена в порядок дня задача автоматизации перевода и информационного поиска; эта задача привлекла к лингвистике внимание специалистов в области точных наук и пред"явила к ней требования, которые невозможно было удовлетворить без резкого повышения уровня строгости лингвистических понятий. [ж)]

x x

x

В настоящее время развитие математической лингвистики достигло уже такой стадии, когда можно систематизировать накопленный ею материал и охарактеризовать ее главные направления.

Различные направления в математической лингвистике связаны с различными подходами к изучению языка (не исключающими, а дополняющими друг друга). Среди этих направлений выделяются два основных, к настоящему времени наиболее полно разработанных. Одно из них - теория порождающих грамматик - исходит из трактовки грамматики языка как системы, порождающей речевые единицы - словоформы, предложения и т.п. Такой подход дает возможность просто и единообразно описывать указанные единицы, связывая структуру каждой из них с процессом порождения ее соответствующей грамматикой. Формальный аппарат теории порождающих грамматик

ж) Роль прикладных задач в математической лингвистике часто преувеличивается - вплоть до утверждения, что она имеет значение только для этих задач и не связана с теоретической лингвистикой. На самом деле математическая лингвистика занимается как раз разработкой аппарата лингвистической теории, и наличие тесных связей между математической лингвистикой и прикладными лингвистическими вопросами этому нисколько не противоречит, т.к. решение таких задач, как автоматизация перевода и информационного поиска, невозможно без глубокой теоретической базы (и именно лингвистической, а не какой-либо другой!).

носит математикологический характер ; по существу , эта теория есть составная часть так называемой теории исчислений, которая, в свою очередь, является одним из разделов (точнее, ответвлений) математической логики. Проблематика теории порождающих грамматик имеет много точек соприкосновения с проблематикой теории алгоритмов и теории автоматов.

Начало теории порождающих грамматик было положено работами американского лингвиста Н.Хомского. В настоящее время эта теория интенсивно разрабатывается как самим Н.Хомским и его сотрудниками, так и многими другими исследователями в различных странах.

Другое направление состоит в построении таких описаний языков, в которых исходным материалом служит некоторая совокупность речевых единиц и с помощью формальных операций над этой совокупностью получаются определенные "грамматические характеристики" языка. (Такой подход в известном смысле противоположен подходу теории порождающих грамматик). Методы этого направления - алгебраические и комбинаторные, но и здесь есть точки соприкосновения с теорией алгоритмов и теорией автоматов. Указанное направление - мы условно назовем его (по его центральному понятию) "теорией замещаемости " - было начато работами советского математика О.С.Кулагиной.

Среди прочих направлений можно отметить теорию так называемых категориальных грамматик Лесьневского-Айдукевича - Бар-Хиллела; эти грамматики, подобно порождающим, сопоставляют каждой речевой единице определенного вида описание ее структуры, но при этом представляют собой не порождающие, а распознающие системы. Важным, но пока еще мало разработанным разделом математической лингвистики является теория отношений подчинения между частями предложения.

Особенностью многих конструкций математической лингвистики является то, что они могут применяться к языковым явлениям различной конкретной природы, или, как часто говорят, к различным "уровням" языка. Так, с помощью порождающих грамматик можно одинаково успешно описывать как синтаксический уровень языка (правила образования предложений из слов), так и морфологический (правила образования слов из морфем). Это явление - приложимость одной и той же абстрактной системы к разным конкретным фактам - является в математике вполне обычным, но для лингвистики возможность исследования разных уровней языка одинаковыми средствами - совершенно новая ситуация.

x x

x

В настоящий курс включены следующие вопросы: способы описания синтаксической структуры предложения - системы составляющих и отношения подчинения (гл.I), теория замещаемости (гл.П), основы теории порождающих грамматик (гл.Ш), категориальные грамматики (гл.IУ) ; в гл.У излагаются некоторые алгоритмические вопросы, связанные с порождающими грамматиками. Ограниченный об"ем курса не позволил включить в него ряд важных вопросов - автоматы с магазинной памятью (pushdown store automata), одно - значность выводов в грамматиках, оценки сложности выводов и др. Из алгоритмических вопросов теории грамматик изложена лишь небольшая часть. В то же время для сохранения цельности изложения в конспект включены теорема о конгруентностях конечного индекса (теорема 3.IO) и доказательство замкнутости класса автоматных языков относительно операций об"единения, пересечения и т.д. (часть доказательства теоремы 3.II), хотя эти вопросы рассматриваются в курсе теории автоматов.

Лингвистические приложения методов математической лингвис - тики в курсе не излагаются; однако автор стремился по возможности останавливаться на лингвистической интерпретации основных понятий и иллюстрировать изложение лингвистическими примерами. В тех случаях, когда какое-либо понятие (или система понятий) допускает различные лингвистические интерпретации, рассматривается только одна из них. При этом оказывается возможным все время пользоваться одной и той же интерпретацией, относящейся к син - таксическому уровню языка и оперирующей словами (точнее, словоформами) как неразложимыми единицами. Эту интерпретацию мы будем называть главной.

Для изучения первых четырех глав формально требуется лишь очень небольшой об"ем математических знаний- некоторые первоначальные сведения из теории множеств , теории графов и общей алгебры. Нетрудно было бы построить изложение так, чтобы никаких предварительных знаний вообще не требовалось; в этом, однако , не было необходимости, т.к. курс математической лингвистики читается после курсов введения в дискретную математику и алгебры; к тому же чтение глав I-IУ требует некоторой привычки к абст - рактным математическим рассуждениям, при наличии которой не представляет труда и самостоятельное ознакомление с нужными понятиями. Для чтения главы У требуется знакомство с теорией алгоритмов в об"еме, например, первых пяти глав книги А.И.Мальцева "Алгоритмы и рекурсивные функции".

Никаких сведений из лингвистики, выходящих за рамки школьного курса грамматики русского языка, для чтения не требуется.

ГЛАВА I.

ПРЕДСТАВЛЕНИЕ СИНТАКСИЧЕСКОЙ СТРУКТУРЫ ПРЕДЛОЖЕНИЯ

§ I. Цепочки, подцепочки, вхождения.

Мы будем рассматривать непустое конечное множество $V=\{a_1,\ldots,a_n\}$, элементы которого будем называть **элементарными символами** или просто **символами**. В главной интерпретации элементарные символы будут представлять собой словоформы некоторого естественного языка. Само множество V будем называть **словарем**. Произвольные конечные последовательности символов будем называть **цепочками** ж) над словарем V . (Будет рассматриваться и пустая цепочка, которую мы всегда будем обозначать Λ). Непустая цепочка записывается в виде $a_{i_1}\ldots a_{i_k}$, где $i_1,\ldots,i_k$ принимают значения $1,\ldots,n$. Число k мы будем называть **длиной** цепочки $a_{i_1}\ldots a_{i_k}$ и обозначать $\ell(a_{i_1}\ldots a_{i_k})$. Длину пустой цепочки будем по определению считать равной нулю.

Множество всех цепочек над V будем обозначать $F(V)$. Введем на множестве $F(V)$ бинарную операцию **конкатенации** (приписывания) следующим образом: I) если $x=a_{i_1}\ldots a_{i_k}$, $y=a_{j_1}\ldots a_{j_m}$, то результатом конкатенации x и y будет цепочка $a_{i_1}\ldots a_{i_k}a_{j_1}\ldots a_{j_m}$; 2) для любой цепочки y результатом конкатенации как Λ и y , так и y и Λ является y . Результат конкатенации x и y обозначается $x\cdot y$ или xy . Множество $F(V)$ с такой операцией является, как известно, (см. например , [8] , стр. 163)

ж) В алгебре то, что мы называем цепочкой, обычно обозначают термином "слово";мы не пользуемся им во избежание смешения с лингвистическим термином "слово",имеющим другой смысл.

свободной полугруппой с системой свободных образующих V .

Среди цепочек над V выделяются некоторые "отмеченные " цепочки, считающиеся в главной интерпретации "грамматически правильными предложениями" соответствующего языка (способов выделения этих цепочек мы в данной главе не касаемся). Каждой такой цепочке естественно приписать какую-то "синтаксическую структуру" (т.е. произвести нечто вроде того, что в школьной грамматике называют анализом предложения). Существуют два основных способа представления этой структуры- с помощью **систем составляющих** и с помощью **отношений синтаксического подчинения**. Изучению этих способов и будет посвящена настоящая глава. Но предварительно нам необходимо ввести еще некоторые важные понятия.

Пусть x и y - цепочки над V . Мы будем называть y **подцепочкой** цепочки x , если существуют такие цепочки z и u , что $x=zyu$. (Например, цепочки a, aca, $baab$, abc являются подцепочками цепочки $acabaabc$; Λ является подцепочкой любой цепочки ; каждая цепочка является подцепочкой самой себя) .

Если $x=a_{i_1}\ldots a_{i_k}$, $y=a_{j_1}\ldots a_{j_m}$ - цепочки над V и y - подцепочка x , мы будем называть **вхождением** y в x всякую пару (y,s) , где s - натуральное число такое, что $i_s=j_1,\ i_{s+1}=j_2,\ldots,\ i_{s+m-1}=j_m$. Если y состоит из одного символа, будем говорить о **вхождении символа** в x. (Например, если $x=abbbcbc$, то вхождений bc в x имеется два: $(bc,4)$ и $(bc,6)$; вхождений b в x имеется четыре : $(b,2)$, $(b,3)$, $(b,4)$ и $(b,6)$; вхождение bcb в x имеется одно : $(bcb,4)$; вхождений bb в x имеется два : $(bb,2)$ и $(bb,3)$).

Пусть $x=a_{i_1}\ldots a_{i_p}$ - цепочка над V, $y=a_{j_1}\ldots a_{j_q}$ и $z=a_{k_1}\ldots a_{k_r}$ - подцепочки x и (y,s) , (z,t) - вхождения соответственно y и z в x . Если $s\le t\le s+q-1$ или $t\le s\le t+r-1$, будем говорить, что вхождения (y,s) и (z,t) **перекрываются**. Если же $s\le t$ и $t+r-1\le s+q-1$, будем говорить, что (y,s) **покрывает** (z,t) (или что (z,t) **вложено** в (y,s)) . Очевидно, если (y,s) покрывает (z,t), то (y,s) и (z,t) перекрываются. (Пусть,например,

$x = abbbcbc$; тогда : единственное вхождение в x цепочки ab перекрывается с первым вхождением bb, но не перекрывается со вторым вхождением bb и ни с одним вхождением bc; единственное вхождение $bcbc$ покрывает оба вхождения bc, но не покрывает ни одного вхождения bb) .

§ 2. Системы составляющих.

Пусть x - цепочка над V и B_x - множество всех вхождений подцепочек в x .
Множество $C \subseteq B_x$ называется системой составляющих цепочки x, если : I) C содержит $(x,1)$ и все вхождения символов (одноэлементных цепочек) ; 2) для любых $\alpha, \beta \in C$ либо α покрывает β, либо β покрывает α, либо α и β не перекрываются.

Элементы C называются составляющими . Составляющие, которые представляют собой вхождения символов, будем называть атомными. Во всех системах составляющих цепочки x множество атомных составляющих одно и то же.

В главной интерпретации составляющие - это словосочетания, т.е. подцепочки предложения, являющиеся в каком-то смысле "синтаксическими единствами" (См.ниже пример 3).

Пусть (y,s) и (z,t) - различные составляющие системы C такие, что (y,s) покрывает (z,t) и не существует составляющей $(w,r) \in C$, которая : I) покрывает (z,t) ; 2) вложена в (y,s) и 3) отлична от (y,s) и (z,t). Тогда мы будем говорить, что (y,s) непосредственно покрывает (z,t) (или что (z,t) непосредственно вложена в (y,s)) .

Примеры.

I) Пусть $x = aaabbbcbcbcb$
Множество $C_1 = \{(x,1), (aaa,1), (a,1), (a,2), (a,3), (bbb,4), (b,4), (b,5), (b,6), (cbcbcb,7), (cb,7), (c,7), (b,8), (c,9), (b,10), (cb,11), (c,11), (b,12)\}$

является системой составляющих для x .

2) Другой системой составляющих для того же x является множество $C_2 = \{(x,1), (aaabbb,1), (aaa,1), (a,1), (aa,2), (a,2), (a,3), (bbb,4), (b,4), (bb,5), (b,5), (b,6), (cbcbcb,7), (cb,7), (c,7), (b,8), (cbcb,9), (cb,9), (c,9), (b,10), (cb,11), (c,11), (b,12)\}$.

3) Пусть V - множество словоформ русского языка и x = свободные от занятий студенты ушли в кино. Поскольку каждый символ (словоформа) входит в x только один раз, мы можем отождествлять вхождения подцепочек в x с самими этими цепочками. Следующее множество есть система составляющих для x : $C = \{x$, свободные от занятий студенты, свободные от занятий, свободные , от занятий, от, занятий, студенты, ушли в кино, ушли, в кино, в, кино$\}$.
Здесь составляющие имеют простой лингвистический смысл (группа подлежащего, группа сказуемого, предложная группа и т.п.).

4) Для любой цепочки x множество, состоящее из $(x,1)$ и всех вхождений символов в x , является системой составляющих.

Систему составляющих цепочки x удобно представлять в виде графа следующим образом: вершинами служат составляющие, и из вершины α в вершину β идет дуга (стрелка) тогда и только тогда, когда α непосредственно покрывает β .

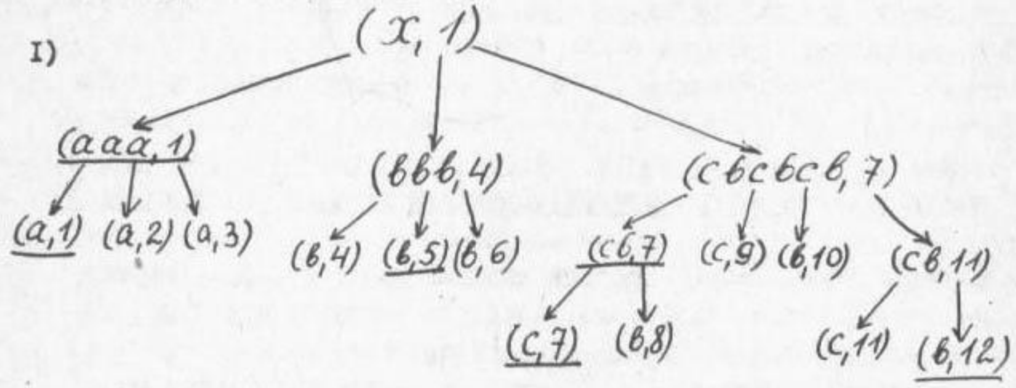
Мы будем называть граф деревом, если он не содержит контуров (замкнутых путей), в каждую его вершину входит не более одной дуги и существует единственная вершина (корень дерева) , в которую не входит никакая дуга. ж)

ж) В теории графов термин дерево часто употребляется в другом, более широком смысле (см. [I] , стр.I65). (Цифры в квадратных скобках относятся к списку литературы в конце конспекта). Деревья в нашем смысле названы в русском переводе книги [I] (стр.I73) , прадеревьями (в оригинале arborescences). Поскольку всякое прадерево является деревом в смысле [I] (см.там же теорему 7 на стр.I73) и никакие другие деревья нам не понадобятся, наше употребление термина дерево не может привести к недоразумениям.

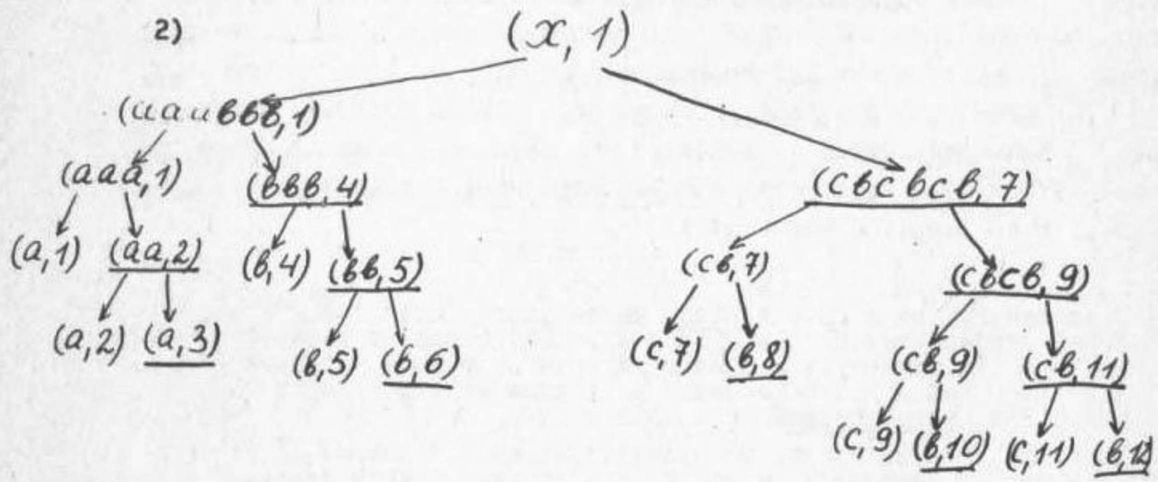
Ясно, что граф, изображающий систему составляющих, является де - ревом с корнем $(x,1)$. Действительно, в этом графе путь из составляющей α в составляющую β существует тогда и только тогда, когда α покрывает β и отлична от β ; поэтому замк - нутые пути невозможны. То, что в каждую вершину входит не более одной дуги, сразу следует из пункта 2) определения системы сос - тавляющих. Очевидно также, что в $(x,1)$ не входит никакая дуга и $(x,1)$ - единственная вершина, обладающая этим свойством.

Системам составляющих примеров I) - 4) соответствуют де - ревья, показанные на черт.I (значение подчеркивания некоторых составляющих будет об"яснено ниже).

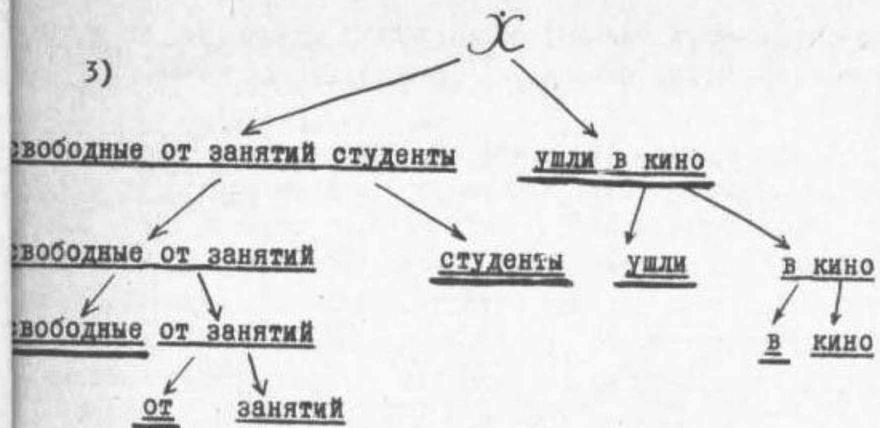
I)



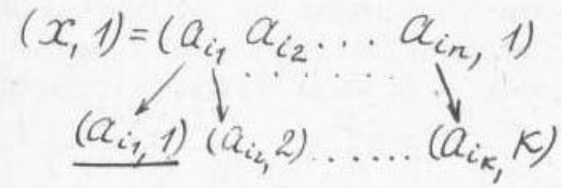
2)



3)



4)



Черт.I

Укажем еще один способ изображения систем составляющих,ме - ее громоздкий и достаточно наглядный (по крайней мере в неслож- ых случаях). Он состоит просто в том, что каждая составляющая аключается в скобки. Пункт 2) определения системы составляющих арантирует однозначность восстановления системы составляющих по акому изображению. Атомные составляющие и составляющую $(x,1)$ ожно в скобки не заключать.

Системы из примеров I) - 4) изображаются при этом способе ледующим образом:

I) $(aaa)(bbb)((cb)cb(cb))$

2) $((a(aa))(b(bb)))((cb)((cb)(cb)))$

3) ((свободные (от занятий)) студенты) (ушли (в кино)).

4) $a_{i_1} a_{i_2} \dots a_{i_k}$

Мы будем называть систему составляющих <u>бинарной</u>, если соответствующее ей дерево бинарно. *) Системы наших примеров 2) и 3) бинарны.

Пусть C - система составляющих цепочки $\mathcal{X}$. Выделим в C некоторые составляющие, которые будем называть <u>главными</u>, так, чтобы : I) составляющая $(\mathcal{X},1)$ не была главной ; 2) для каждой неатомной составляющей α среди составляющих, непосредственно вложенных в α, одна и только одна была главной. Если C - система составляющих и K - выделенное в ней множество главных составляющих, мы будем называть K <u>разметкой</u> системы C, а упорядоченную пару (C,K) - <u>размеченной системой составляющих</u> цепочки $\mathcal{X}$.

Системы составляющих из примеров I) - 4) можно разметить, в частности, так, как показано на черт.I, где главные составляющие подчеркнуты. Разумеется, разметка системы составляющих, вообще говоря, не единственна.

§ 3. <u>Отношения синтаксического подчинения</u>.

Пусть $\mathcal{X}$ - цепочка над словарем V и A_x - множество всех вхождений символов в $\mathcal{X}$. Пусть на A_x задано бинарное отношение R такое, что множество A_x с этим отношением, рассматриваемое как граф, является деревом. Всякое такое отношение мы будем называть <u>отношением синтаксического подчинения</u> (или просто <u>отношением подчинения</u>) для $\mathcal{X}$.

Рассмотрим несколько примеров отношений подчинения(Черт.2).

Для этих примеров возьмем те же цепочки, что и в примерах предыдущего параграфа. Договоримся при графическом изображении отношения подчинения представлять вхождения символов точками горизонтальной прямой так, чтобы при $p<q$ вхождение (a_{i_p},p) располагалось левее вхождения (a_{i_q},q). При таком способе изображения вхождений можно без потери информации писать вместо (a_{i_p},p) просто a_{i_p}, что мы и будем делать. Кроме того, все дуги дерева, соответствующего отношению подчинения,

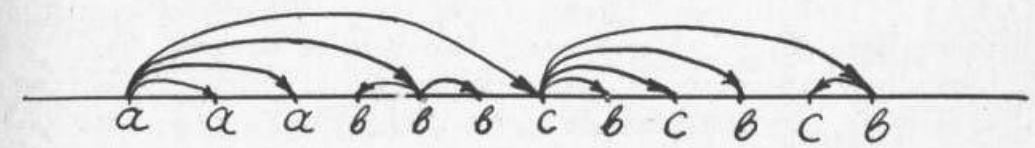
---

*) Дерево называется <u>бинарным</u>, если из каждой его вершины исходит либо два ребра, либо ни одного.

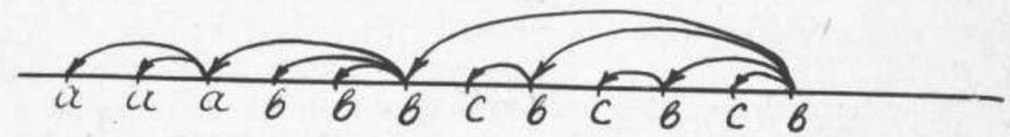
будем проводить выше прямой, на которой расположены вхождения символов. (Каждая стрелка в дереве направляется от подчиняющего вхождения к подчиняемому).

Описанных соглашений мы будем придерживаться не только в нижеследующих примерах, но и во всем дальнейшем изложении.

I)

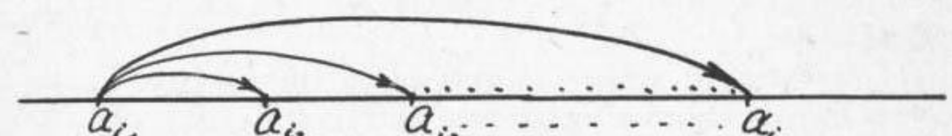


2)



3)

4)



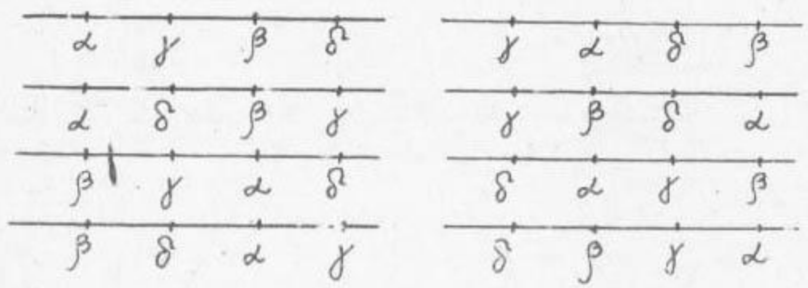
Черт.2.

Для дальнейшего нам понадобится несколько вспомогательных понятий. Вхождения символов в цепочку мы будем далее называть <u>точками</u> этой цепочки.

Пусть $\mathcal{X}$ - цепочка над V и $\alpha=(a_{i_p},p)$, $\beta=(a_{i_q},q)$, $\gamma=(a_{i_r},r)$ - точки $\mathcal{X}$. Если $p<q$, будем говорить, что α <u>предшествует</u> β и β <u>следует</u> за α. Если $p<q<r$ или $r<q<p$, будем говорить, что β <u>расположена между</u> α и γ.

Пусть $\alpha, \beta, \gamma, \delta$ - точки $\mathfrak{X}$ такие, что в точности одна из точек α и β расположена между γ и δ. В этом случае мы будем говорить, что пары (α, β) и (γ, δ) <u>разделяют друг друга</u>.

Всевозможные случаи разделения пар показаны на черт.3.



Черт.3.

Пусть, далее, для цепочки $\mathfrak{X}$ определено отношение подчинения. Последовательность $\alpha_0, \alpha_1, \ldots, \alpha_n$ точек $\mathfrak{X}$ будем называть <u>иерархической</u>, если для каждого $i = 1, \ldots, n$ α_{i-1} подчиняет α_i. Если α, β - точки $\mathfrak{X}$, для которых существует иерархическая последовательность $\alpha_0, \ldots, \alpha_n$ такая, что $\alpha_0 = \alpha$ и $\alpha_n = \beta$, будем говорить, что β <u>зависит</u> от α.

Заметим, что отношение зависимости транзитивно, антирефлексивно (т.е. никакое вхождение не зависит от самого себя) и антисимметрично (т.е. если β зависит от α, то α не зависит от β). (Транзитивность непосредственно следует из определения зависимости, антирефлексивность - из отсутствия контуров в графе подчинения, антисимметричность - из транзитивности и антирефлексивности).

Будем, далее, называть <u>кустом</u> точки α цепочки $\mathfrak{X}$ множество, состоящее из всех точек, зависящих от α, и самой α. Всякое множество, полученное из куста точки α выбрасыванием кустов некоторых (или всех) точек, подчиненных α, будем называть <u>квазикустом</u> точки α.

Сформулируем теперь следующие основные определения:

Отношение подчинения для цепочки $\mathfrak{X}$ называется <u>проективным</u>, если для любых четырех точек $\alpha, \beta, \gamma, \delta$ цепочки $\mathfrak{X}$ из того, что β подчинена α и δ подчинена γ, следует, что пары (α, β) и (γ, δ) не разделяют друг друга.

Отношение подчинения для цепочки $\mathfrak{X}$ называется <u>сильно проективным</u>, если для любых трех точек α, β, γ цепочки $\mathfrak{X}$ из того, что β подчинена α и γ расположена между α и β, следует, что γ зависит от α.

Для принятого нами графического способа изображения отношения подчинения проективность равносильна возможности провести все стрелки (при соблюдении наших соглашений) так, чтобы никакие две из них не пересекались [*]; сильная проективность равносильна требованию, чтобы каждая точка, находящаяся под некоторой стрелкой, зависела от начала этой стрелки.

В примерах I)- 4) отношения подчинения, как сразу видно из черт.2., проективны и сильно проективны.

Чтобы выяснить значение проективности и сильной проективности, мы введем теперь следующие новые понятия:

Будем говорить, что множество $\mathcal{M}$ точек цепочки $\mathfrak{X} = a_{i_1} a_{i_2} \ldots a_{i_k}$ <u>заполняет отрезок</u>, если существует такое вхождение (y, s) цепочки $y = a_{i_s} a_{i_{s+1}} \ldots a_{i_{s+p}}$ в $\mathfrak{X}$, что

$$\mathcal{M} = \{(a_{i_s}, s), (a_{i_{s+1}}, s+1), \ldots, (a_{i_{s+p}}, s+p)\}$$

В этом случае мы можем отождествить множество $\mathcal{M}$ с вхождением (y, s).

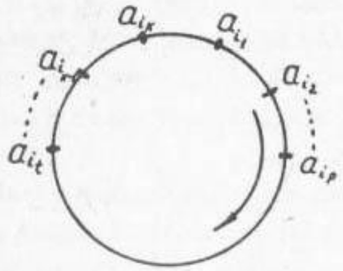
Будем говорить, что множество $\mathcal{M}$ точек цепочки $\mathfrak{X} = a_{i_1} a_{i_2} \ldots a_{i_k}$ <u>заполняет циклический отрезок</u>, если либо $\mathcal{M}$ заполняет отрезок, либо существуют такие вхождения $(y, 1)$ и (z, t) соответственно цепочек $y = a_{i_1} \ldots a_{i_p}$ и $z = a_{i_t} \ldots a_{i_k}$ в $\mathfrak{X}$ (где $p < t$), что

$$\mathcal{M} = \{(a_{i_1}, 1), \ldots, (a_{i_p}, p), (a_{i_t}, t), \ldots, (a_{i_k}, k)\}$$

---

[*] Доказательство этого геометрического факта выходит за рамки настоящего изложения.

(Иначе говоря - если записать $\mathfrak{X}$ по кругу (черт.4), то $\mathcal{M}$ заполнит целую дугу). Если $\mathcal{M}$ заполняет циклический отрезок, то мы можем отождествить $\mathcal{M}$ либо с одним вхождением цепочки, либо с парой вхождений цепочек. Очевидно, множество $\mathcal{M}$ точек цепочки $\mathfrak{X}$ тогда и только тогда не заполняет циклического отрезка, когда существуют точки $\alpha, \beta \in \mathcal{M}$ такие, что как внутри интервала (α, β), так и вне его имеются точки, не принадлежащие $\mathcal{M}$.



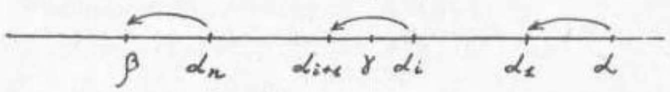
Черт.4.

Теорема I.I. Отношение подчинения для цепочки $\mathfrak{X}$ тогда и только тогда проективно (сильно проективно), когда для каждой точки α цепочки $\mathfrak{X}$ куст α заполняет циклический отрезок (соответственно- заполняет отрезок).

Доказательству этой теоремы предпошлем две простые леммы.

Лемма I.I. Пусть α, β, γ - точки цепочки $\mathfrak{X}$ такие, что β зависит от α , γ не зависит от α и γ лежит между α и β . Тогда существуют такие μ и ν , принадлежащие кусту α , что ν подчинено μ и γ лежит между μ и ν

Доказательство . Пусть $\alpha = \alpha_0, \alpha_1, \ldots, \alpha_n = \beta$ - иерархическая последовательность, и пусть для определенности β предшествует α (черт.5.). Если α_i - последняя точка последовательности, которая еще следует за γ , то γ лежит между α_i

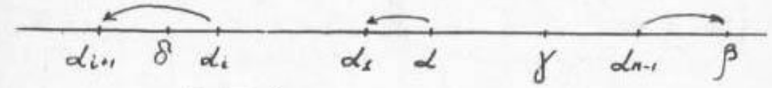


Черт.5.

и α_{i+1} .

Лемма I.2. Пусть $\alpha, \beta, \gamma, \delta$ - точки цепочки $\mathfrak{X}$ такие, что : I) β зависит от α, γ и δ не зависят от α ; 2) γ лежит между α и β ; 3) δ не лежит между α и β . Тогда существуют такие μ и ν , принадлежащие кусту α , что ν подчинено μ и пары (μ, ν) и (γ, δ) разделяют друг друга.

Доказательство. Пусть $\alpha = \alpha_0, \alpha_1, \ldots, \alpha_n = \beta$ - иерархическая последовательность, и пусть для определенности α лежит между γ и δ (черт.6). Если α_i -



Черт.6.

последняя точка последовательности, которая еще лежит между γ и δ , то пары (α_i, α_{i+1}) и (γ, δ) разделяют друг друга.

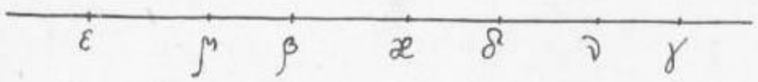
Доказательство теоремы.

Если подчинение не сильно проективно, существуют такие точки α, β, γ цепочки $\mathfrak{X}$, что β подчинено α , γ лежит между α и β и γ не зависит от α . Поскольку куст α содержит β и не содержит γ , он не может заполнять отрезок. Обратно, пусть куст некоторой точки α не заполняет отрезка, и пусть β и γ - соответственно нижняя и верхняя границы куста α . Между β и γ имеется точка δ , не зависящая от α . δ лежит либо между α и β , либо между α и γ ; в обоих случаях по лемме I.I. имеются точки μ и ν , принадлежащие кусту α и такие, что ν подчинено μ и δ лежит между μ и ν . Поскольку δ не зависит от α , δ не зависит и от μ , так что подчинение не является сильно проективным.

Пусть теперь подчинение не проективно, т.е. существуют такие точки $\alpha, \beta, \gamma, \delta$, что β подчинено α , δ подчинено γ и пары (α, β) и (γ, δ) разделяют друг друга. В силу антисимметричности отношения зависимости хотя бы одна из точек α и γ не зависит от другой ; пусть для определенности α не зависит от γ . Тогда и β не зависит от γ , так что и внутри интервала (δ, γ) , и вне его имеются точки, не зависящие от γ . Поэтому куст γ не заполняет циклического отрезка.

Обратно, пусть куст некоторой точки α - обозначим его K_α - не заполняет циклического отрезка. Пусть $\beta, \gamma \in K_\alpha$, $\delta, \varepsilon \,\bar{\in}\, K_\alpha$, δ лежит между β и γ , ε не лежит между β и γ (черт.7). Легко видеть, что где бы ни лежала точка α , либо пары (α, β) и (δ, ε) , либо пары (α, γ) и (δ, ε) разделяют друг друга.

Поэтому в силу леммы I.2 найдутся такие $\mu, \partial \in K_\alpha$, что ∂ подчинено μ и пары (μ, ∂) и (δ, ε) разделяют друг друга. Пусть для определенности δ лежит между μ и ∂ и ε не лежит между μ и ∂.



Черт.7.

Обозначим через $\varkappa$ корень дерева подчинения. Очевидно, $\varkappa \bar{\in} K_\alpha$ (иначе было бы $K_\alpha = \mathscr{X}$). Возможны два случая: 1) $\varkappa$ лежит между μ и ∂; 2) $\varkappa$ не лежит между μ и ∂. Рассмотрим случай 1). Поскольку ε зависит от $\varkappa$, имеется иерархическая последовательность $\varkappa = \varkappa_0, \varkappa_1, \ldots, \varkappa_n = \varepsilon$. Эта последовательность не может содержать ни μ, ни ∂ - иначе ε зависела бы от μ, а, значит, и от α.

Пусть $\varkappa_i$ - последняя точка последовательности, которая еще лежит между μ и ∂. Тогда пары (μ, ∂) и $(\varkappa_i, \varkappa_{i+1})$ разделяют друг друга, причем ∂ подчинена μ и $\varkappa_{i+1}$ подчинена $\varkappa_i$, так что подчинение не проективно. В случае 2) рассуждаем аналогично с заменой ε на δ.

Пусть на цепочке $\mathscr{X}$ задано некоторое отношение подчинения. Интервал, ограниченный двумя точками цепочки $\mathscr{X}$, будем называть <u>подстрелочным</u>, если один из его концов подчинен другому. Подстрелочный интервал назовем <u>неправильным</u>, если внутри него имеется точка, не зависящая ни от одного его конца. Сильная проективность означает отсутствие неправильных подстрелочных интервалов.

Сформулируем теперь три следствия из теоремы I.I.

<u>Следствие I</u>. Всякое сильно проективное отношение подчинения проективно.

<u>Следствие 2</u>. Если отношение подчинения проективно, то каждый неправильный подстрелочный интервал содержит корень дерева подчинения.

<u>Следствие 3</u>. Проективное отношение подчинения тогда и только тогда сильно проективно, когда никакой подстрелочный интервал не содержит корня дерева подчинения.

Следствие I очевидно. Следствие 2 доказывается так: если на цепочке $\mathscr{X}$ определено проективное отношение подчинения и (α, β) - неправильный подстрелочный интервал (для определенности считаем, что α подчиняет β), то корень дерева подчинения во всяком случае отличен от α (иначе интервал (α, β) не был бы неправильным) и тем более не входит в куст α; но, поскольку интервал (α, β) неправильный и куст α заполняет циклический отрезок, все точки, не входящие в куст α, в том числе и корень, должны лежать между α и β. Наконец, следствие 3 непосредственно вытекает из следствия 2.

Из лингвистических исследований известно (см.,например, [3]), что в естественных языках грамматически правильные предложения обычно проективны и даже сильно проективны (точнее, проективны, соответственно сильно проективны, отношения подчинения, естественным образом определяемые на грамматически правильных предложениях). Сравнительно много отклонений от проективности встречается лишь в художественной литературе (особенно в поэзии),где эти отклонения служат одним из средств придания языку особой выразительности. В языке научно-технической и деловой прозы отклонения от проективности редки и, по-видимому, встречаются в основном в тех предложениях, для которых само отношение подчинения не является достаточно естественным средством представления синтаксической структуры (например, предложения, содержащие составные союзы типа "не только,..., но и...", некоторые аналитические формы глаголов и т.п.) ж) Причина этого явления состоит, видимо, в простоте вида кустов, характерной для проективных и сильно проективных отношений подчинения ввиду теоремы I.I.

ж) Эта закономерность проверена, конечно, далеко не для всех языков; но, насколько известно автору, пока что в литературе указан лишь один пример достаточно представительного фрагмента естественного языка, где она не соблюдается (пример Постала [7] , относящийся к языку мохавк).

В заключение параграфа приведем пример проективного отношения подчинения, не являющегося сильно проективным, и два примера непроективных отношений подчинения (черт.8):

Черт.8.

§ 4. Связь между системами составляющих и отношениями подчинения.

Пусть C - система составляющих цепочки x и R - отношение подчинения для той же цепочки. Будем говорить, что R <u>согласовано</u> с C, если : I) все кусты точек цепочки x, соответствующие отношению R, являются составляющими системы C; 2) каждая составляющая системы C является кустом или квазикустом в смысле отношения R. *)

Ввиду теоремы I.I всякое отношение подчинения, согласованное с некоторой системой составляющих, сильно проективно.

---

*) Здесь множества точек отождествляются с вхождениями цепочек (см. выше) .

Мы укажем способы построения по данной системе составляющих согласованного с ней отношения подчинения и по данному сильно проективному отношению подчинения - системы составляющих, с которой данное отношение согласовано. При этом нам будет удобнее исходить из размеченной системы составляющих - это даст возможность строить отношение подчинения однозначно.

Прежде всего нам нужно теперь ввести некоторые новые понятия.

Пусть (C, K) - размеченная система составляющих цепочки x и $\varphi_0, \varphi_1, \dots, \varphi_t = \alpha$ - последовательность составляющих системы C такая, что для каждого $i=0,\dots,t-1$ φ_i непосредственно покрывает φ_{i+1} и $\varphi_t = \alpha$ есть атомная составляющая (точка). Если при этом φ_0 - не главная составляющая, а все составляющие $\varphi_1, \dots, \varphi_t$ - главные, мы будем называть последовательность $\varphi_0, \dots, \varphi_t$ <u>опорной</u> для точки α, а составляющую φ_0 - <u>опорной составляющей</u> этой точки.

<u>Лемма I.3.</u> а) Всякая составляющая входит в одну и только одну опорную последовательность. б) Всякая не главная составляющая является опорной для одной и только одной точки.

<u>Доказательство</u>. Достаточно доказать утверждение а), т.к. б) непосредственно следует из него. Пусть φ - произвольная составляющая, $\psi_1, \dots, \psi_p$ - все составляющие, покрывающие φ (упорядоченные так, что ψ_{i-1} непосредственно покрывает ψ_i), и $\theta_0, \theta_1, \dots, \theta_q$ - последовательность, построенная следующим образом : $\theta_0 = \varphi$; θ_1 - главная составляющая, непосредственно вложенная в φ (если такая существует, то она единственна), θ_2 - главная составляющая, непосредственно вложенная в θ_1, и т.д. Если φ - не главная составляющая, то последовательность $\theta_0, \theta_1, \dots, \theta_q$, очевидно, является опорной.

Пусть $\varphi = \theta_0$ - главная составляющая и s - наибольшее из чисел $1, \dots, p$, для которого составляющая ψ_s - не главная (такое s существует, т.к. ψ_1 есть вся цепочка и, следовательно, не является главной). Тогда, очевидно, последовательность $\psi_s, \psi_{s+1}, \dots, \psi_p, \theta_0, \theta_1, \dots, \theta_q$ - опорная.

Наконец, легко видеть, что две различные опорные последователь - ности не имеют общих членов. Действительно, если $\Phi_0, \Phi_1, \ldots, \Phi_n$ и $\Psi_0, \Psi_1, \ldots, \Psi_m$ - опорные последовательности, то из $\Phi_0 = \Psi_0$ следует совпадение последовательностей, а при $\Phi_0 \neq \Psi_0$ Φ_1 не перекрывается с Ψ_1 и тем более Φ_i не перекрывается с Ψ_j при любых $i=1,\ldots,n$, $j=1,\ldots,m$.

Определим теперь на множестве точек цепочки $\mathfrak{X}$ отношение R следующим образом: $\alpha R \beta$, если опорная составляющая точки β непосредственно вложена в один из членов опорной последовательности точки α. В силу леммы I.3 для всякой точки β существует не более одной точки α такой, что $\alpha R \beta$, и точно для одной точки β (той, для которой опорная составляющая есть $(\mathfrak{X},1)$) не существует точки α такой, что $\alpha R \beta$. Кроме того, из самого определения R вытекает, что граф этого отношения не содержит контуров. Итак, R есть отношение подчинения для $\mathfrak{X}$. Мы будем говорить, что это отношение <u>связано</u> с размеченной системой составляющих (C,K).

<u>Теорема I.2</u> . Отношение подчинения, связанное с размеченной системой составляющих (C,K), согласовано с системой составляющих C.

<u>Доказательство</u>. Пусть (C,K) - размеченная система составляющих, R - связанное с (C,K) отношение подчинения и $\Phi_0, \Phi_1, \ldots, \Phi_t = \alpha$ - произвольная опорная последовательность. Покажем, что Φ_0 совпадает с кустом точки α (в смысле R), а $\Phi_1, \ldots, \Phi_t$ - квазикусты (в смысле R). В силу леммы I.3 из этого утверждения будет непосредственно следовать теорема.

Назовем сложностью составляющей общее число составляющих, вложенных в нее. Докажем наше утверждение индукцией по сложности s составляющей Φ_0.

При $s=0$ утверждение очевидно, т.к. в этом случае последовательность состоит из одной составляющей Φ_0, совпадающей с α, и куст α состоит из одной точки α. Пусть утверждение доказано для опорных последовательностей, сложности начальных членов которых меньше s. Очевидно, $\bigcap_{i=0}^{t} \Phi_i = \alpha$, откуда

$$\Phi_0 = (\Phi_0 \setminus \Phi_1) \cup (\Phi_1 \setminus \Phi_2) \cup \ldots \cup (\Phi_{t-1} \setminus \Phi_t) \cup \alpha .$$

Но для каждого $i=1,\ldots,t$ $\Phi_{i-1} \setminus \Phi_i$ есть об"единение всех не главных составляющих, непосредственно вложенных в Φ_{i-1}; отсюда в силу определения отношения R следует, что $(\Phi_0 \setminus \Phi_1) \cup \ldots \cup (\Phi_{t-1} \setminus \Phi_t)$ есть об"единение опорных составляющих всех точек, подчиненных α ; а это - в силу индуктивного предположения - не что иное, как об"единение кустов всех точек, подчиненных α. Но об"единение кустов всех точек, подчиненных α, и самой α есть как раз куст точки α.

Далее, для любого $i=1,\ldots,t$ имеем $\Phi_i = \Phi_0 \setminus [(\Phi_0 \setminus \Phi_1) \cup \ldots \cup (\Phi_{i-1} \setminus \Phi_i)]$. Φ_0 есть куст α, а каждое из множеств $\Phi_0 \setminus \Phi_1, \ldots, \Phi_{i-1} \setminus \Phi_i$ есть ,как мы уже отмечали, об"единение кустов некоторых точек, подчиненных α. Поэтому Φ_i - квазикуст. Теорема доказана.

Справедливо и обратное предложение:

<u>Теорема I.3.</u> Всякое отношение подчинения, согласованное с системой составляющих C, связано с некоторой (и притом единственной) размеченной системой составляющих (C,K), построенной для данной C.

Предварительно докажем следующую лемму:

<u>Лемма I.4</u>. Пусть (C,K) - размеченная система составляющих и R - такое отношение подчинения, что для всякой точки α куст α в смысле R совпадает с опорной составляющей точки α в смысле (C,K). Тогда , если R' - отношение подчинения,связанное с (C,K), то R совпадает с R'.

<u>Доказательство леммы</u>. Пусть $\Phi_0, \Phi_1, \ldots, \Phi_t = \alpha$ опорная последовательность (в смысле (C,K)). Как в доказательстве теоремы I.2, имеем $\Phi_0 = (\Phi_0 \setminus \Phi_1) \cup \ldots \cup (\Phi_{t-1} \setminus \Phi_t) \cup \alpha$.

Для каждого $i=1,\ldots,t$ Φ_i содержит α и отлично от Φ_0 ; поэтому при $i \neq 0$ Φ_i не может быть кустом и, следовательно, является квазикустом точки α. А отсюда следует , что каждое из множеств $\Phi_{i-1} \setminus \Phi_i$ есть об"единение кустов некоторых точек, подчиненных α.

Пусть теперь $\alpha R \beta$ и Ψ – куст β. Тогда, по только что сказанному, существует такое i $(1 \le i \le t)$, что Ψ содержится в $\Phi_{i-1} \setminus \Phi_i$. Следовательно, Φ_{i-1} покрывает Ψ. Если бы существовала составляющая Θ, вложенная в Φ_{i-1} и покрывающая Ψ, то Θ содержала бы α; но среди составляющих, вложенных в Φ_{i-1}, только составляющие $\Phi_i, \Phi_{i-1}, \dots, \Phi_t$ содержат α, а ни одна из них не покрывает Θ. Итак, Ψ непосредственно вложена в Φ_{i-1}, так что $\alpha R' \beta$. Обратно, пусть $\alpha R' \beta$, т.е. куст Ψ точки β (совпадающий по условию с ее опорной составляющей), непосредственно вложен в составляющую Φ_i $(i = 0, \dots, t-1)$. Тогда во всяком случае $\beta \in \Phi_0$, так что β зависит от α (в смысле R).

Пусть γ – такая точка, что $\alpha R \gamma$ и β принадлежит кусту γ, который мы обозначим через Θ. Тогда $\Psi \subseteq \Theta$, так что Θ перекрывается с Φ_i, и значит, либо $\Theta \subseteq \Phi_i$, либо $\Phi_i \subseteq \Theta$. Но $\Phi_i \subseteq \Theta$ невозможно, поскольку Θ, являясь кустом γ, должна быть в силу условия леммы не главной составляющей; однако наименьшая не главная составляющая, покрывающая Φ_i, есть Φ_0, так что из $\Phi_i \subseteq \Theta$ следовало бы $\Phi_0 \subseteq \Theta$, что противоречит зависимости γ от α. Итак, $\Theta \subseteq \Phi_i$. Но отсюда следует, что $\Psi = \Theta$ (иначе составляющая Ψ не была бы <u>непосредственно</u> вложена в Φ_i), так что $\beta = \gamma$ и $\alpha R \beta$.
Мы показали, таким образом, что $\alpha R \beta$ тогда и только тогда, когда $\alpha R' \beta$. Лемма доказана.

<u>Доказательство теоремы.</u>

Пусть R – отношение подчинения, согласованное с C, и Φ – произвольная неатомная составляющая системы C. Φ является кустом или квазикустом некоторой точки α. Ту из составляющих, непосредственно вложенных в Φ, которая содержит α, будем считать главной. Полученную так разметку обозначим K. Покажем, что R связано с (C, K).

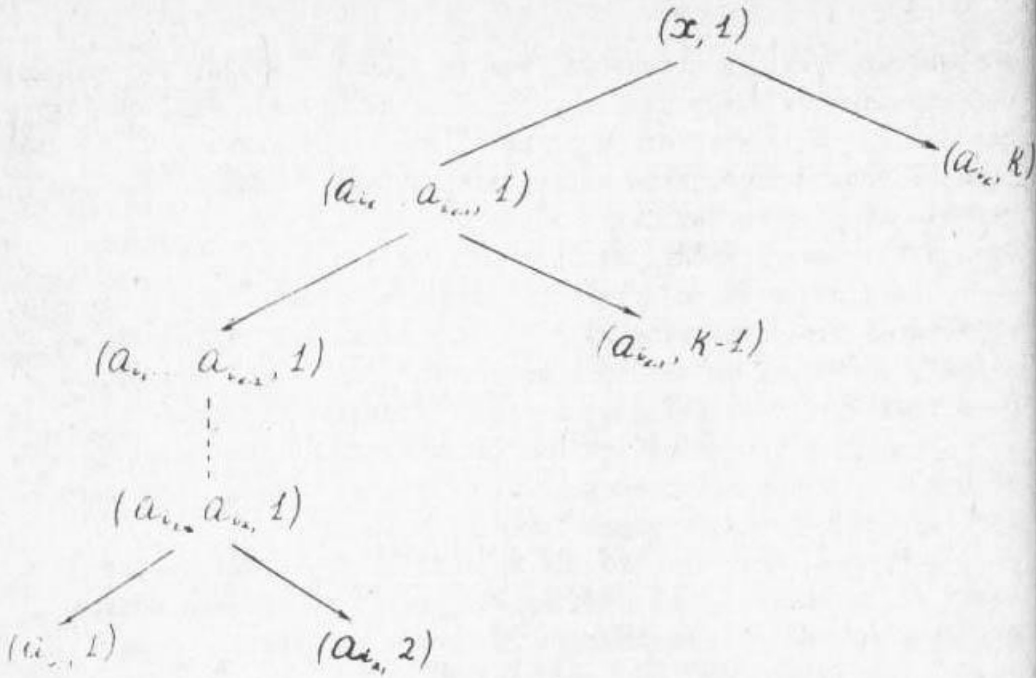
Пусть α – произвольная точка цепочки и Φ – ее куст. Рассмотрим последовательность составляющих $\Phi_0, \Phi_1, \dots, \Phi_t$ такую, что $\Phi_0 = \Phi$, $\Phi_t = \alpha$ и для каждого $i = 1, \dots, t$ Φ_{i-1} непосредственно покрывает Φ_i. Составляющая Φ_0 – не главная, т.к. из доказательства теоремы I.2 видно, что главная составляющая не может быть кустом. В то же время очевидной индукцией по i доказывается, что при $i > 0$ составляющая Φ_i – главная. Следовательно, наша последовательность – опорная для α, так что опорная составляющая точки α есть $\Phi_0 = \Phi$. Отсюда по лемме I.4 следует, что R связано с (C, K).

Единственность разметки K следует из того, что – как видно из доказательства теоремы I.2 – в размеченной системе составляющих, с которой связано отношение R, главными являются все те и только те составляющие, которые не являются кустами.

<u>Пример</u>. Легко видеть, что отношения подчинения, изображенные на черт.2., связаны с размеченными системами составляющих соответствующих цепочек, изображенными на черт.I.

Нетрудно заметить, что для каждого сильно проективного отношения подчинения R на цепочке $\mathfrak{X}$ существуют системы составляющих, с которыми R согласовано. В частности, такой системой является множество, состоящее из всех точек цепочки $\mathfrak{X}$ и всех кустов. Действительно, по теореме I.I все кусты сильно проективного отношения подчинения – отрезки и для любого отношения подчинения два куста могут пересекаться лишь тогда, когда один из них содержится в другом. Из этой системы составляющих – назовем ее минимальной – можно получить любую другую систему, с которой R согласовано, если в каждый куст вставить еще некоторую вложенную последовательность квазикустов– иначе говоря, если не сразу "разрывать" куст на кусты точек, подчиненных его корню, а несколько раз "отрывать" от него по несколько "веточек". В частности, если каждый раз "отрывать" по одной "веточке", получится бинарная система составляющих.

<u>Пример</u>. Система составляющих, изображенная на черт.I-4)- минимальная система, согласованная с отношением, изображенным на черт.2-4). Бинарная система, согласованная с тем же отношением, изображена на черт.9.



Черт.9.

Бинарная система составляющих, с которой согласовано данное отношение подчинения, также не всегда единственна. Простейший пример такого рода изображен на черт.IO.

Черт.IO.

Замечание. Во всех рассуждениях этого параграфа сильная проективность нужна только для того, чтобы все кусты были отрезками. Если изменить определение системы составляющих, допустив в качестве элементов системы множества точек, заполняющие циклические отрезки, то все наши рассуждения можно будет перенести на

проективные отношения подчинения. Если же допустить в качестве составляющих любые множества точек (конечно, при сохранении условий I) и 2) определения системы составляющих), то те же рассуждения будут годиться для произвольных отношений подчинения.

ГЛАВА П.

ЗАМЕЩАЕМОСТЬ

§ 5. Два способа описания языка.

Основной целью всех дальнейших рассмотрений будет установление связи между свойствами множества грамматически правильных предложений языка, с одной стороны, и свойствами синтаксической структуры этих предложений - с другой. Наблюдения над естественными языками показывают, что, зная свойства синтаксической структуры предложений некоторого языка, можно получить способ строить правильные предложения этого языка или отличать правильные предложения от неправильных; обратно, если множество правильных предложений языка каким - то способом задано, можно многое сказать о свойствах синтаксической структуры в этом языке. Поэтому при формальном описании строения языка [ж)] естественно либо исходить из понятия грамматически правильной фразы и получать понятия, относящиеся к синтаксической структуре, как производные, либо считать свойства синтаксической структуры заданными и получать в качестве производного понятие грамматически правильного предложения. К первому типу принадлежат способы описания языка, основанные на понятии замещаемости; этим способам посвящена настоящая глава. Ко второму типу относятся порождающие и распознающие грамматики, которые будут рассмотрены в двух следующих главах.

§ 6. Понятия замещаемости, взаимозамещаемости и конфигурации.

Пусть V - произвольный словарь. Любое множество цепочек над V мы будем называть языком над V. Рассматривая фиксированный словарь, будем обычно опускать слова " над V " .В этом

ж) Точнее - синтактического уровня языка. Однако те же соображения применимы и к другим уровням,например,морфологическому.

и следующем параграфах словарь все время фиксирован.

Будем говорить, что цепочка x замещаема на цепочку y в языке L (обозначение: $x \Rightarrow y\ (L)$ или, при фиксированном L, просто $x \Rightarrow y$), если для любых цепочек z_1 и z_2 из $z_1 x z_2 \in L$ следует $z_1 y z_2 \in L$. Если $x \Rightarrow y(L)$ и $y \Rightarrow x(L)$, будем говорить, что x и y взаимозамещаемы в языке L (обозначение: $x \Leftrightarrow y(L)$ или $x \Leftrightarrow y$).

Замечания. I) Взаимозамещаемость является, очевидно, эквивалентностью - и даже конгруентностью - на полугруппе $F(V)$.

2) Из $x \Rightarrow y(L)$ и $x \in L$ следует $y \in L$.

Если интерпретировать V как множество словоформ некото- рого конкретного языка и L - как множество грамматически правильных фраз ж) этого языка (указанная интерпретация счи- тается главной), то взаимозамещаемые цепочки будут представлять собой словосочетания, " синтаксически эквивалентные", т.е. выпол- няющие в точности одну и ту же синтаксическую функцию. Так, в русском языке цепочка высокому дому взаимозамещаема, по-видимо- му, с цепочкой умному человеку, цепочка стене - с доске, хоро- шими - с новыми.

Дадим теперь следующее определение. Если цепочка x, дли- на которой больше I, взаимозамещаема в языке L с элементарным символом a, то x называется конфигурацией I-го ранга языка L с результирующим a.

В главной интерпретации конфигурации I-го ранга - это "потенциально синтаксически связные" словосочетания, иначе - "потенциальные составляющие" (т.е. цепочки, которые могут - хотя и не обязаны ! - входить в лингвистически естественные системы составляющих грамматически правильных предложений). На- пример, словосочетание очень маленькая, по -видимому, является в русском языке конфигурацией I-го ранга с результирующим малень- кая.

ж) Грамматическую правильность не следует смешивать с осмысленностью.

Однако "потенциальные составляющие " заведомо не исчерпы - ваются конфигурациями I-го ранга. Рассмотрим, например, слово- сочетание маленькая птичка, которое естественно считать "потен- циальной составляющей". Ясно, что оно не может быть взаимозаме- щаемо ни с каким словом, отличным от существительного женского рода в именительном падеже единственного числа. В то же время все существительные женского рода в именительном падеже единст- венного числа взаимозамещаемы; поэтому, если цепочка маленькая птичка была бы конфигурацией I-го ранга, то слово птичка было бы ее результирующим. Это, однако, не имеет места - например, во фразе в окно влетела очень маленькая птичка нельзя без нару- шения грамматической правильности заменить маленькая птичка на птичка; такая замена возможна лишь в тех контекстах, где нет сочетаний вроде очень маленькая , или совсем маленькая, пере- крывающихся с заменяемым вхождением сочетания маленькая птичка. В то же время слово птичка (по-видимому) замещаемо на малень- кая птичка. Эти соображения приводят к следующему определению.

Пусть r - натуральное число >1 и для каждого нату - рального $i<r$ определено понятие конфигурации ранга i языка L. Тогда цепочка x длины >1 называется конфигурацией ранга r языка L с результирующим a (где a - элементар- ный символ), если выполняются следующие два условия : I) $a \Rightarrow x\ (L)$; 2) если $z_1 x z_2 \in L$ и цепочка $z_1 x z_2$ не содержит вхождений конфигураций рангов $< r$, перекрывающихся с выделен- ным вхождением x, но не содержащихся в нем целиком, то $z_1 a z_2 \in L$.

Например, в русском языке маленькая птичка есть, видимо, конфигурация второго ранга.

Из определения ясно, что всякая конфигурация ранга r явля- ется и конфигурацией любого ранга, большего r (с тем же резуль- тирующим).

Заметим еще, что результирующий конфигурации, вообще говоря, не единственен. Например, для конфигурации маленькая птичка результирующим будет любое существительное женского рода в имени- тельном падеже единственного числа.

§ 7. Конфигурационные характеристики.

Будем называть цепочку, принадлежащую языку L, неприводимой, если она не содержит вхождений конфигураций языка L. Множество неприводимых цепочек языка L будем обозначать $Б(L)$.

Конфигурацию ранга r языка L будем называть простой, если она не содержит вхождений конфигураций ранга r (или, что то же самое, рангов $\le r$) языка L, отличных от нее самой.

Пусть T - некоторое множество конфигураций языка L и T^+ - 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^+ будет обозначаться $K(L)$ (соответственно $П(L)$).

Упорядоченную пару $(Б(L), K(L))$ (соответственно $(Б(L), П(L))$) будем называть полной (соответственно приведенной) конфигурационной характеристикой языка L.

Лемма 2.I. Пусть L_1 и L_2 - языки над словарем 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 $l(x)$ *) покажем, что $x \in L_2$.

Базис. Если $l(x)=1$, то из $x \in L_1$ следует $x \in Б(L_1)$; но $Б(L_1) \subseteq Б(L_2) \subseteq L_2$.

Индукционный шаг. Допустим, что всякая цепочка y языка L_1 такая, что $l(y) \le n$, принадлежит L_2. Пусть $l(x) = n+1$ и $x \in L_1$. Если $x \in Б(L_1)$, рассуждаем, как в базисе. Если же $x \notin Б(L_1)$ и r - наименьший ранг конфигураций языка L, содержащихся в x, то x содержит и простую конфигурацию ранга r; иначе говоря, существуют такие z_1, z_2, w, b, что $x = z_1 w z_2$ и $(w, b) \in П(L_1)$. Поскольку x не содержит конфигураций рангов, меньших r, имеем $z_1 b z_2 \in L_1$. Но $l(z_1 b z_2) < n+1$, откуда по индуктивному предположению $z_1 b z_2 \in L_2$. А поскольку $(w, b) \in П(L_1) \subseteq K(L_2)$, получаем $x = z_1 w z_2 \in L_2$.

*) Напомним, что через $l(x)$ обозначается длина x, т.е. число вхождений символов в x.

Из леммы 2.I непосредственно следует

Теорема 2.I. Пусть L_1 и L_2 - языки над словарем V. Если $Б(L_1) = Б(L_2)$ и $K(L_1) = K(L_2)$ или соответственно $Б(L_1) = Б(L_2)$ и $П(L_1) = П(L_2)$, то $L_1 = L_2$.

Иначе говоря, язык над словарем V вполне определяется своей полной или приведенной конфигурационной характеристикой.

Особо интересным с содержательной лингвистической точки зрения представляется класс языков, у которых $Б$ и $П$ конечны. Такие языки мы будем называть конечно характеризуемыми. По-видимому, естественные языки конечно характеризуемы, т.к. в них все достаточно длинные грамматически правильные фразы и конфигурации содержат вхождения некоторых стандартных словосочетаний, которые, вероятно, при всяком разумном уточнении понятия грамматически правильной фразы соответствующего конкретного языка будут простыми конфигурациями. (Напомним, что словарь V конечен !). (В то же время множество всех конфигураций естественного языка, видимо, бесконечно. Например, в русском языке сочетание существительного с любым числом определяющих его прилагательных будет конфигурацией).

Пример. Пусть $V = \{a, b, c, d, e, f, g\}$,
$L = \{abc, abde, bdbc, bdbde, af, bdf, ge, dfe, abbb, bdbbb\}$. Найдем конфигурационные характеристики языка L. Сначала для каждого $\alpha \in V$ найдем множество

$$\Phi_\alpha = \{x \mid x \in F(V),\ \alpha \Rightarrow x(L),\ l(x) > 1\}$$

Непосредственно проверяется, что $bd \in \Phi_a$. Других цепочек Φ_a не содержит, т.к. если бы символ a был замещаем на какую-либо цепочку $x \ne bd$, то язык L содержал бы цепочку xbc, что не имеет места. Далее, множества Φ_b, Φ_d и Φ_e пусты, поскольку b, d и e входят в цепочку $bdbde \in L$, длина которой равна 5, и если бы один из этих символов был замещаем на цепочку x, где $l(x)>1$, то язык L содержал бы цепочку длины >5. Φ_f также пусто, поскольку L не содержит цепочек вида dxe, где $x \neq f$.

Φ_c содержит, как непосредственно видно, цепочку de и не содержит других цепочек, поскольку в L нет цепочек вида abx, где $l(x) > 1$ и $x \neq de$. Наконец, аналогично предыдущему легко проверить, что $\Phi_g = \{df,\ abd,\ bdbd\}$

Далее, непосредственно проверяется, что $de \Rightarrow c$, $abd \Rightarrow g$, $bdbd \Rightarrow g$. В то же время bd не замещаема на a и df не замещаема на g. Итак, язык L имеет три конфигурации I-го ранга – abd и $bdbd$ с результирующим g, de с результирующим c. Далее, bd есть конфигурация 2-го ранга с результирующим a; действительно, только в двух цепочках языка L, а именно в $bdbc$ и $bdbbb$, имеются вхождения bd, не перекрывающиеся с вхождениями конфигураций I-го ранга, и цепочки abc и $abbb$, получаемые соответственно из $bdbc$ и $bdbbb$ заменой bd на a, также принадлежат L. В то же время df не является конфигурацией 2-го ранга с результирующим g, поскольку цепочка bdf не содержит вхождений конфигураций I-го ранга и все же $bg \bar{\in} L$. Однако df есть конфигурация 3-го ранга с результирующим g, поскольку вхождение df, не перекрывающееся со вхождением конфигураций I-го и 2-го рангов, имеется только в цепочке dfe и $ge \in L$. Таким образом, язык L имеет пять конфигураций, из которых простыми, очевидно, являются три: $de,\ bd,\ df$.

Итак:

$$Б(L) = \{abc,\ df,\ ge,\ abbb\};$$

$$K(L) = \{(de, c),\ (abd, g),\ (bdbd, g),\ (bd, a),\ (d$$

$$\Pi(L) = \{(de, c),\ (bd, a),\ (df, g)\}.$$

§ 8. Укрупнения эквивалентностей.

Пусть $V = \{a_1, \ldots\ldots, a_n\}$ – словарь, $F(V)$ – свободная полугруппа с системой свободных образующих V и R – некоторая эквивалентность на V. Будем обозначать через $\bar{R}$ отношение на $F(V)$, определяемое следующим образом:

$x \bar{R} y$ тогда и только тогда, когда либо $x = y = \Lambda$, либо $x = a_{i_1} \ldots\ldots a_{i_k}$, $y = a_{j_1} \ldots\ldots a_{j_k}$,

$a_{i_1} R a_{j_1}, \ldots\ldots, a_{i_k} R a_{j_k}$

Ясно, что $\bar{R}$ – эквивалентность и, более того, конгруентность на $F(V)$.

<u>Пример</u>. Пусть для элементарных символов a_i и a_j $a_i S_L a_j$ означает " a_i взаимозамещаем с a_j в языке L " (Классы эквивалентности по отношению S_L называются <u>семействами</u> языка L). Тогда $x \bar{S}_L y$ влечет $x \Longleftrightarrow y(L)$; обратное, вообще говоря, неверно.

Положим теперь $F_R(V) = F(V)/\bar{R}$ ж)

Обозначим через V_R множество всех классов эквивалентности по отношению R и через $F(V_R)$ – свободную полугруппу с системой свободных образующих V_R. Естественный гомоморфизм $F(V)$ на $F_R(V)$ будем обозначать φ_R.

Пусть $x = a_{i_1} \ldots\ldots a_{i_k} \in F(V)$ и $X = \varphi_R(x)$. Обозначим через $\psi_R(X)$ цепочку $\varphi_R(a_{i_1}) \ldots\ldots \varphi_R(a_{i_k})$ над V_R. Кроме того, будем полагать $\psi_R(\Lambda) = \Lambda$. Ясно, что ψ_R есть отображение $F_R(V)$ <u>на</u> $F(V_R)$. Легко видеть, что это отображение является изоморфизмом. Гомоморфное отображение $F(V)$ на $F(V_R)$, являющееся произведением φ_R и ψ_R, обозначим Θ_R. Очевидно, $\tilde{x} = \Theta_R(x)$ тогда и только тогда, когда либо x и $\tilde{x}$ – пустые цепочки, либо $x = a_{i_1} \ldots\ldots a_{i_k}$, $\tilde{x} = \tilde{a}_{i_1} \ldots\ldots \tilde{a}_{i_k}$, где $\tilde{a}_{i_j}$ – класс эквивалентности по отношению R, содержащий a_{i_j}. Цепочка $\Theta_R(x)$ называется R – <u>структурой</u> цепочки x.

Пусть R_1 и R_2 – эквивалентности на V. Мы будем называть R_2 <u>укрупнением</u> R_1, если из $x R_1 y$ следует $x R_2 y$ – иначе говоря, если каждый элемент V_{R_1} является подмножеством некоторого элемента V_{R_2}. Если R_2 – укрупнение R_1, $Y \in F(V_{R_1})$, $Z \in F(V_{R_2})$, мы будем полагать $Z = \eta_{R_1 R_2}(Y)$, если либо $Y = A_1 \ldots A_k$, $Z = B_1 \ldots B_k$, где

ж) $F(V)/\bar{R}$ означает фактор-полугруппу полугруппы $F(V)$ по $\bar{R}$ (См. [8], стр.113).

$A_1, \ldots, A_k \in V_{R_1}$, $B_1, \ldots, B_k \in V_{R_2}$, $A_i \subseteq B_i$ $(i=1,\ldots,k)$ либо $Y = \Lambda$, $Z = \Lambda$. Очевидно, $\varrho_{R_1R_2}$ - гомоморфизм $F(V_{R_1})$ на $F(V_{R_2})$ и $\Theta_{R_2} = \Theta_{R_1} \cdot \varrho_{R_1R_2}$.

Пусть теперь L - язык над V. Будем полагать для всякой эквивалентности R на V $L_R = \Theta_R(L)$. L_R есть язык над V_R, состоящий из R - структур цепочек языка L. В главной интерпретации L_R есть множество "схем" или "скелетов" грамматически правильных предложений.

Укрупнение R_2 эквивалентности R_1 на V мы будем называть L - <u>регулярным</u>, если любые два R_1 - класса, содержащиеся в одном R_2 - классе, взаимозамещаемы в языке L_{R_1}.

<u>Пример</u>. Если E - тривиальная эквивалентность, все классы которой одноэлементны, то S_L (см.выше) есть L - регулярное укрупнение E. Более того, эквивалентность R тогда и только тогда является L - регулярным укрупнением E, когда S_L есть укрупнение R.

Если R и R' - эквивалентности на V и L - язык над V, то R' называется L - <u>производной эквивалентностью</u> для R, если R' есть L - регулярное укрупнение R и никакое собственное (т.е. отличное от R') укрупнение R' не является L - регулярным укрупнением для R (иначе говоря, R' есть максимальное L - регулярное укрупнение R).

<u>Пример</u>. S_L есть L - производная эквивалентность для E.

<u>Теорема 2.2.</u> Если R_2 - регулярное укрупнение R_1, то $Y \in L_{R_1}$ тогда и только тогда, когда $\varrho_{R_1R_2}(Y) \in L_{R_2}$.

<u>Доказательство.I</u>. В силу соотношения $\Theta_{R_2} = \Theta_{R_1} \cdot \varrho_{R_1R_2}$ имеем $L_{R_2} = \varrho_{R_1R_2}(L_{R_1})$; поэтому из $Y \in L_{R_1}$ вытекает $\varrho_{R_1R_2}(Y) \in L_{R_2}$.

II. Пусть $Z = \varrho_{R_1R_2}(Y) \in L_{R_2}$. Тогда существует цепочка $y \in F(V)$ такая, что $y \in L$ и $\Theta_{R_2}(y) = Z$. Положим $Y' = \Theta_{R_1}(y)$; тогда $Y' \in L_{R_1}$. В силу соотношения $\Theta_{R_2} = \Theta_{R_1} \cdot \varrho_{R_1R_2}$ имеем $\varrho_{R_1R_2}(Y') = Z$. Пусть теперь $Z = B_1 \ldots B_k$, где $B_1, \ldots, B_k \in V_{R_2}$. Тогда по определению отображения $\varrho_{R_1R_2}$ будем иметь $Y = A_1 \ldots A_k$,

$Y' = A'_1 \ldots A'_k$, где $A_1, \ldots A_k, A'_1, \ldots, A'_k \in V_{R_1}$ и $A_i, A'_i \subseteq B_i$ $(i=1,\ldots,k)$. Т.к. R_2 - регулярное укрупнение R_1, последнее включение влечет $A_i \Longleftrightarrow A'_i\ (L_{R_1})$. Отсюда - поскольку взаимозамещаемость является конгруентностью - следует, что $Y \Longleftrightarrow Y'(L_{R_1})$, а поэтому $Y \in L_{R_1}$.

<u>Теорема 2.3</u>. Если R_2 - регулярное укрупнение эквивалентности R_1, то : I) $X \Longrightarrow Y(L_{R_1})$ тогда и только тогда, когда $\varrho_{R_1R_2}(X) \Longrightarrow \varrho_{R_1R_2}(Y)(L_{R_2})$; 2) X является конфигурацией ранга r языка L_{R_1} с результирующим A тогда и только тогда, когда $\varrho_{R_1R_2}(X)$ является конфигурацией ранга r языка L_{R_2} с результирующим $\varrho_{R_1R_2}(A)$.

<u>Доказательство</u>. I)Пусть $X \Longrightarrow Y(L_{R_1})$,

$S = \varrho_{R_1R_2}(X)$, $T = \varrho_{R_1R_2}(Y)$

и U_1, U_2 - произвольные цепочки над V_{R_2}. Если Z_1, Z_2 - цепочки над V_{R_1} такие, что $U_1 = \varrho_{R_1R_2}(Z_1)$, $U_2 = \varrho_{R_1R_2}(Z_2)$, то, поскольку $\varrho_{R_1R_2}$ - гомоморфизм, имеем

$$U_1 S U_2 = \varrho_{R_1R_2}(Z_1 X Z_2),$$

$$U_1 T U_2 = \varrho_{R_1R_2}(Z_1 Y Z_2)$$

. По теореме 2.2. из $U_1 S U_2 \in L_{R_2}$ следует $Z_1 X Z_2 \in L_{R_1}$; отсюда $Z_1 Y Z_2 \in L_{R_1}$ и - снова по теореме 2.2 - $U_1 T U_2 \in L_{R_2}$.

Т.о., $S \Longrightarrow T(L_{R_2})$. Аналогично доказывается, что $S \Longrightarrow T(L_{R_2})$ влечет $X \Longrightarrow Y(L_{R_1})$.

2) докажем индукцией по r. При $r=1$ утверждение сразу следует из I). Пусть оно доказано для всех $i<r$ и пусть X - конфигурация ранга r языка L_{R_1} с результирующим A. Положим $\varrho_{R_1R_2}(X) = S$, $\varrho_{R_1R_2}(A) = B$. Пусть U_1, U_2 - произвольные цепочки над V_{R_2} такие, что $U_1 S U_2 \in L_{R_2}$ и Z_1, Z_2 - такие цепочки над V_{R_1}, что $\varrho_{R_1R_2}(Z_1) = U_1$, $\varrho_{R_1R_2}(Z_2) = U_2$. Тогда $\varrho_{R_1R_2}(Z_1 A Z_2) = U_1 B U_2$, $\varrho_{R_1R_2}(Z_1 X Z_2) = U_1 S U_2$. Если при этом цепочка $U_1 S U_2$ не содержит вхождений конфигураций рангов $< r$ языка L_{R_2}, перекрывающихся с выделенным

вхождением S, но не содержащихся в нем целиком, то в силу индуктивного предположения цепочка $Z_1 X Z_2$ не содержит вхождений конфигураций рангов $< r$ языка L_{R_1}, перекрывающихся с выделенным вхождением X, но не содержащихся в нем целиком ; поэтому $Z_1 A Z_2 \in L_{R_1}$, откуда по теореме 2.2 $U_1 B U_2 \in L_{R_2}$. Кроме того, в силу I) $B \Rightarrow S(L_{R_2})$. Итак, S есть конфигурация ранга r языка L_{R_2} с результирующим B. Обратное утверждение доказывается аналогично.

Лемма 2.2. Если $R_2 - L$ - регулярное укрупнение R_1 и $R_3 - L$ - регулярное укрупнение R_2, то R_3 есть L -регулярное укрупнение R_1.

Доказательство. Пусть $A, A' \in V_{R_1}$, $C \in V_{R_3}$ и $A, A' \subseteq C$. Положим $\rho_{R_1R_2}(A) = B$, $\rho_{R_1R_2}(A') = B'$. Поскольку $A \subseteq B$ и $A' \subseteq B'$, пересечения $B \cap C$ и $B' \cap C$ непусты, а отсюда, поскольку R_3 - укрупнение R_2, следует, что $B \subseteq C$ и $B' \subseteq C$; поэтому $B \Longleftrightarrow B'(L_{R_2})$, откуда по теореме 2.3 $A \Longleftrightarrow A'(L_{R_1})$. Лемма доказана.

Из леммы 2.2. сразу следует

Теорема 2.4. Если $R' - L$ - производная эквивалентность для R и $R'' - L$ - производная эквивалентность для R', то R'' совпадает с R'.

§ 9. Окрестности, классы и типы.

Пусть V - словарь и L - язык над V. Введем на V некоторую специальную эквивалентность, классы которой будем называть окрестностями. В главной интерпретации окрестность будет представлять собой парадигму, т.е. множество всех форм одного слова (например, $\{$дом, дома, дому,..$\}$). Упорядоченную тройку (V, L, Γ) будем называть языком с парадигмами.

Для произвольного $a \in V$ и произвольной эквивалентности R на V будем обозначать через $R(a)$ тот класс эквивалентности R, которому принадлежит a.

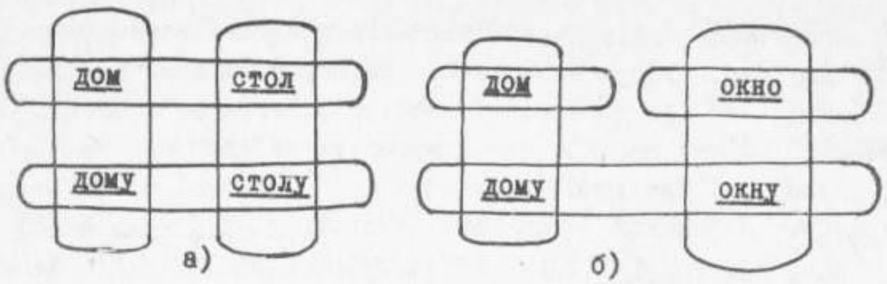
Язык с парадигмами (V, L, Γ) будем называть однородным, если для любых $a, b \in V$ из $S_L(a) \cap \Gamma(b) \neq \varnothing$ следует $\Gamma(a) \cap S_L(b) \neq \varnothing$. ($\varnothing$ означает пустое множество).

Содержательно однородность языка с парадигмами означает, что если два слова в одной какой-нибудь форме "синатаксически эквивалентны", то они должны быть "синтаксически эквивалентны" и во всех остальных формах. Чтобы лучше уяснить смысл этого понятия, рассмотрим следующий пример.

Пример. Рассмотрим шесть предложений: у него есть новый стол, у него нет нового стола, он подошел к новому столу, он увидел новый стол, он сидел за новым столом, он говорил о новом столе. Обозначим через L множество, состоящее из этих шести предложений и всех предложений, получающихся из них подстановкой вместо форм слова стол соответствующих форм слов дом, стена, машина и изменением прилагательных - если это нужно - так, чтобы сохранилось согласование, т.е. чтобы предложение осталось правильным. Через V обозначим множество всех словоформ, встречающихся в цепочках языка L ; окрестности определим естественным образом, так что, например, Γ (дом) = $\{$дом, дома, дому, домом, доме$\}$, Γ (новый) = $\{$новый, нового, новому, новым, новом, новая, новой, новую$\}$.

Легко убедиться, что S_L (стол) = $\{$стол, дом$\}$, S_L (стене) = $\{$стене, машине$\}$ и т.д. Непосредственно проверяется, что построенный нами язык с парадигмами (V, L, Γ) однороден. Например, S_L (дом) $\cap\ \Gamma$ (столу) = $\{$стол$\}$, Γ (дом) $\cap\ S_L$ (столу) = $\{$дому$\}$ (черт.IIа.).

Добавим теперь к V слова окно, дерево во всех формах единственного числа и к L - предложения, полученные из шести исходных заменой форм слова стол формами слов окно, дерево таким же образом, как раньше. Окрестности новых слов определим естественным образом. Полученный язык с парадигмами обозначим (V', L', Γ'). Он уже не будет однородным. Действительно, легко проверить, что $S_{L'}$(дом) = $\{$дом, стол$\}$, $S_{L'}$ (окно) = $\{$окно, дерево$\}$, но $S_{L'}$ (дому) = $\{$дому, столу, окну, дереву$\}$. Поэтому $S_{L'}$ (окну) $\cap\ \Gamma$(дом) = $\{$дому$\}$ и в то же время Γ (окну) $\cap\ S_{L'}$ (дом) = $\varnothing$ (черт.IIб).



Черт.II.

Далее, преобразуем каждое предложение из L', добавив к нему придаточное который (- ую,-ое) он купил (с соблюдением согласования) . Например, предложение у него есть новая машина перейдет в у него есть новая машина, которую он купил. Полученный язык обозначим L''. Доопределим соответствующим образом словарь V' до нового словаря V'' и эквивалентность Γ' до новой эквивалентности Γ''. Язык с парадигмами (V'', L'', Γ'') снова будет однородным, т.к. в нем каждое семейство существительных содержит только слова одного рода; например, окну не взаимозамещаемо в L'' с дому,т.к. окну нельзя заменить на дому в предложении он подошел к новому окну, которое он купил.

Пусть теперь (V, L, Γ) - язык с парадигмами, γ - некоторая окрестность и σ - семейство. Будем называть классом, порожденным окрестностью γ (соответственно- семейством σ), об"единение всех семейств (соответственно-окрестностей), пересекающихся с γ (с σ).

Теорема 2.5. В однородном языке с парадигмами всякий класс порождается любой окрестностью и любым семейством, пересекающимися с ним.

Доказательство проведем для случая класса, порожденного семейством; для класса, порожденного окрестностью, рассуждение будет аналогичным.

Пусть (V, L, Γ) - язык с парадигмами , σ - некоторое семейство языка L и $\mathfrak{X}$ - класс, порожденный семейством σ . Пусть , далее, γ - окрестность, пересекающаяся с $\mathfrak{X}$, и $\mathfrak{X}_1$ - класс, ею порожденный.

Покажем, что $\mathfrak{X} = \mathfrak{X}_1$. Пусть сначала $a \in \mathfrak{X}_1$. Нам нужно показать, что $a \in \mathfrak{X}$, т.е. что $\Gamma(a) \cap \sigma \neq \emptyset$. Для этого заметим прежде всего, что, поскольку γ пересекается с $\mathfrak{X}$, γ имеет непустое пересечение с некоторой окрестностью, пересекающейся с σ , а,значит, совпадает с этой окрестностью. Следовательно, существует $b \in V$ такой, что $\sigma = S_L(b)$, $\gamma = \Gamma(b)$. Далее, $a \in \mathfrak{X}_1$ означает, что

$$S_L(a) \cap \gamma = S_L(a) \cap \Gamma(b) \neq \emptyset.$$

Отсюда в силу однородности языка $\Gamma(a) \cap S_L(b) = \Gamma(a) \cap \sigma \neq \emptyset$. Итак, $\mathfrak{X}_1 \subseteq \mathfrak{X}$. Обратное включение доказывается аналогично, поскольку в нашем рассуждении семейства и окрестности равноправны.

Пусть теперь σ' - семейство, пересекающееся с $\mathfrak{X}$, и $\mathfrak{X}_2$ - класс, им порожденный. Покажем, что $\mathfrak{X}_1 = \mathfrak{X}_2$. Пусть $a \in \mathfrak{X}_2$. Непустота пересечения $\mathfrak{X} \cap \sigma'$ означает, что существует окрестность γ , для которой $\sigma \cap \gamma \neq \emptyset$ и $\sigma' \cap \gamma \neq \emptyset$. В то же время $a \in \mathfrak{X}_2$ означает, что $\sigma' \cap \Gamma(a) \neq \emptyset$. Пусть $b \in \sigma \cap \gamma$, $c \in \sigma' \cap \gamma$, $d \in \sigma' \cap \Gamma(a)$. Тогда $\sigma = S_L(b)$, $\sigma' = S_L(c) = S_L(d)$, $\gamma = \Gamma(b) = \Gamma(c)$
В силу однородности языка из $\sigma' \cap \Gamma(a) = S_L(c) \cap \Gamma(a) \neq \emptyset$ следует $\Gamma(c) \cap S_L(a) \neq \emptyset$. Но $\Gamma(c) = \Gamma(b)$, а $\Gamma(b) \cap S_L(a) \neq \emptyset$ влечет $S_L(b) \cap \Gamma(a) = \sigma \cap \Gamma(a) \neq \emptyset$, т.е. $a \in \mathfrak{X}$. Итак, мы доказали,что $\mathfrak{X}_2 \subseteq \mathfrak{X}$. Обратное включение доказывается аналогично.

Из доказанной теоремы непосредственно следует, что для однородного языка с парадигмами классы образуют разбиение словаря, т.е. что существует эквивалентность $K_{L,\Gamma}$ на V такая , что $a K_{L,\Gamma} b$ тогда и только тогда, когда a и b входят в один класс. Очевидно, $K_{L,\Gamma}$ является укрупнением эквивалентностей S_L и Γ .

Теорема 2.6. Если (V, L, Γ) - однородный язык с парадигмами, то $K_{L,\Gamma}$ является регулярным укрупнением Γ .

Доказательство. Пусть γ, γ' - окрестности, $\mathfrak{X}$ - класс и $\gamma, \gamma' \subseteq \mathfrak{X}$. Достаточно, очевидно, доказать, что $\gamma \Rightarrow \gamma'(L_\Gamma)$. Пусть ξ_1, ξ_2 — такие цепочки окрестностей , что $\xi_1 \gamma \xi_2 \in L_\Gamma$. Тогда существуют цепочки

x_1, x_2 над V и символ $a \in V$, для которых $x_1 a x_2 \in \mathcal{L}$ и при этом $\xi_1 = \Theta_\Gamma(x_1)$, $\xi_2 = \Theta_\Gamma(x_2)$, $\gamma = \Theta_\Gamma(a)$ (т.е. $a \in \gamma$) . По теореме 2.5. семейство $S_\mathcal{L}(a)$ порождает класс $\mathcal{X}$; поэтому пересечение $S_\mathcal{L}(a) \cap \gamma'$ непусто. Если $a' \in S'_\mathcal{L}(a) \cap \gamma'$,то a' взаимозамещаемо с a , так что $x_1 a' x_2 \in \mathcal{L}$; в то же время $\gamma' = \Theta_\Gamma(a')$, так что $\xi_1 \gamma' \xi_2 = \Theta_\Gamma(x_1 a' x_2)$, откуда по теореме 2.2. (поскольку $\Theta_\Gamma = \rho_{E\Gamma}$ и $\mathcal{L} = \mathcal{L}_E$, где E - тривиальная эквивалентность) $\xi_1 \gamma' \xi_2 \in \mathcal{L}_\Gamma$.

Будем обозначать через $T_{\mathcal{L},\Gamma}$ эквивалентность, $\mathcal{L}$ - производную для Γ . В силу леммы 2.2 и теоремы 2.6 в однородном языке с парадигмами $T_{\mathcal{L},\Gamma}$ совпадает с эквивалентностью, $\mathcal{L}$ - производной для $K_{\mathcal{L},\Gamma}$. Классы эквивалентности по отношению $T_{\mathcal{L},\Gamma}$ мы будем называть <u>типами</u>.

§ 10. <u>Пример</u>.

Следующий пример поясняет лингвистический смысл понятий класса и типа.

Пусть $\mathcal{L}$ состоит из грамматически правильных русских предложений, построенных по одной и той же простой схеме, а именно

$$A\, S\, Ad\ V\, A_1 S_1,\ K_1 V_1,\ A_2 S_2,\ K_2\, \underline{\text{я}}\, V_2$$

Здесь S - существительное или личное местоимение в именительном падеже единственного числа, S_1, S_2 - существительные в единственном числе, A, A_1, A_2 - прилагательные в единственном числе, V - глагол в настоящем времени и единственном числе, V_1 - глагол в 3-ем лице единственного числа настоящего времени, K_1 - слово <u>который</u> (-<u>ая</u>, - <u>ое</u>) в именительном падеже единственного числа, V_2 - переходный глагол в 1-м лице единственного числа настоящего времени, K_2 - слово <u>который</u> (<u>ая</u>, -<u>ое</u>) в винительном падеже единственного числа (т.е. одна из словоформ <u>который</u>, <u>которого</u>, <u>которую</u>, <u>которое</u>), Ad-наречие образа действия.

Некоторые из указанных в схеме членов могут отсутствовать, но при условии, что предложение остается правильным. Конечно, правильность может пониматься по-разному ; но, во всяком случае, мы будем считать правильными только такие предложения, в которых соблюдены все согласования в роде, одушевленности, падеже, числе и лице, а местоимения не имеют при себе прилагательных. Кроме того, эллиптические предложения мы не будем считать правильными. При таких условиях различные определения правильности могут возникать, видимо, только в связи с колебаниями в вопросе о возможности или необходимости для того или иного глагола управлять тем или иным падежом ; для наших целей эти различия несущественны.

Языку $\mathcal{L}$ будут принадлежать, например, предложения: <u>молодой лесоруб быстро рубит высокое дерево острым топором</u> ; <u>ты смеешься</u> ; <u>оно помогает человеку, который работает</u>;<u>дифференцируемый утюг несет восторгу, который заседает, рощу, которую я теряю</u> ; <u>я даю книгу новому студенту</u> . В то же время предложения <u>я даю</u>, <u>я даю книгу</u> , <u>я даю книгу новому</u> , <u>даю книгу новому студенту</u> не войдут в $\mathcal{L}$ как эллиптические.

Обратим внимание на то, что личные местоимения входят в предложения из $\mathcal{L}$ только в качестве подлежащих. Это сделано исключительно ради упрощения (как и исключение множественного числа).

Словарь V будем считать состоящим из всех словоформ, встречающихся в предложениях языка $\mathcal{L}$, и запятой (которая тоже встречается в предложениях языка $\mathcal{L}$). При этом ради простоты мы будем считать, что омонимичные формы одного слова представляют собой разные элементы V , если эти формы являются омонимичными только у отдельных слов; например, мы будем считать, что слово <u>кофе</u> в дательном и творительном падежах - это разные элементы V , скажем, <u>кофе</u> $_{\text{д.}}$ и <u>кофе</u>$_{\text{тв.}}$ В то же время формы, омонимия которых является общим правилом, не будут "расщепляться"— например, прилагательные мужского и среднего рода в творительном падеже, прилагательные женского рода в родительном, дательном , творительном и предложном падежах. Не будут считаться различными, в частности, одушевленные существительные мужского рода в родительном и винительном падежах и неодушевленные в именительном и винительном.

Разбиение словаря V на семейства, соответствующие языку L, получается очевидным образом. Оно имеет следующий вид.

I) Существительные распадаются на следующие семейства :

$S_1 = \{$человек, конь,... $\}$; $S_5 = \{$дом, стол, ... $\}$;
$S_2 = \{$человека, коня,... $\}$; $S_6 = \{$дома, стола,... $\}$;
$S_3 = \{$человеку, коню,... $\}$; $S_7 = \{$дому, столу,... $\}$;
$S_4 = \{$человеком, конем,... $\}$; $S_8 = \{$домом, столом,... $\}$;

$S_9 = \{$жена, лампа,... $\}$; $S_{14} = \{$окно, лицо,... $\}$;
$S_{10} = \{$жены, лампы,... $\}$; $S_{15} = \{$окна, лица,... $\}$;
$S_{11} = \{$жене, лампе,... $\}$; $S_{16} = \{$окну, лицу,... $\}$;
$S_{12} = \{$жену, лампу,... $\}$; $S_{17} = \{$окном, лицом,... $\}$;
$S_{13} = \{$женой, лампой,... $\}$;

(Поясним, что, например, конем и домом потому относятся к разным семействам, что в предложении типа я восхищаюсь конем, которого я вижу нельзя заменить конем на домом).

2) Прилагательные распадаются на такие семейства:

$S_{18} = \{$новый, красный ,... $\}$;
$S_{19} = \{$нового, красного,... $\}$;
$S_{20} = \{$новому, красному,... $\}$;
$S_{21} = \{$новым, красным,... $\}$;
$S_{22} = \{$новое, красное,... $\}$;
$S_{23} = \{$новая, красная,... $\}$;
$S_{24} = \{$новой, красной,... $\}$;
$S_{25} = \{$новую, красную,... $\}$;

3) Местоимения распадаются на три семейства:
$S_{26} = \{$я$\}$; $S_{27} = \{$ты$\}$; $S_{28} = \{$он , она, оно$\}$.

4) Наречия составляют одно семейство $S_{29} = \{$быстро, спокойно,...$\}$

5) Для каждой глагольной формы a семейство $S_L(a)$ состоит из всех форм глаголов того же лица, имеющих одинаковое управление с a . Например, завидуешь и уступаешь войдут в одно семейство, т.к. оба они требуют дополнения в дательном падеже и не допускают двух дополнений ; другое семейство образуют глаголы восхищаешься, истекаешь, требующие дополнения в творительном падеже и тоже не допускающие двух дополнений ; в третье семейство войдут глаголы спишь , краснеешь, не допускающие (беспредложных) дополнений вообще, и т.д. Выписать все семейства глаголов не представляло бы особого труда, хотя это и было бы утомительно; мы этого делать не будем.

6) Каждая форма слова который образует отдельное семейство.

7) Запятая, разумеется, также образует отдельное семейство.

Снабдим теперь язык L окрестностями таким образом, что окрестность каждого существительного, прилагательного и глагола, а также слова который, будет состоять из всех форм этого слова, а окрестности местоимений, наречий и запятой будут одноэлементными. Полученную эквивалентность обозначим Γ .

Легко видеть, что язык с парадигмами (V, L, Γ) однороден.

Выясним, что представляют собой классы этого языка.

I) Каждое семейство, состоящее из одушевленных существительных мужского рода, пересекается, очевидно, со всеми окрестностями одушевленных существительных мужского рода и не пересекается ни с какими другими окрестностями. Поэтому класс, порожденный этим семейством, состоит в точности из всех одушевленных существительных мужского рода (во всех падежах), т.е. является об"единением семейств S_1 , S_2 , S_3 , S_4 . Мы будем обозначать этот класс M' . Совершенно аналогично получаются классы неодушевленных существительных мужского рода, существительных женского рода и существительных среднего рода, которые мы обозначим соответственно M'' , Ж , С .

2) Каждое семейство, состоящее из прилагательных, пересекается со всеми окрестностями прилагательных и не пересекается ни с какими другими окрестностями. Таким образом, прилагательные образуют

один класс. Мы обозначим его через Π .

3) Классы местоимений совпадают с семействами, поскольку окрестности местоимений одноэлементны.

4) По аналогичной причине наречия образуют один класс.

5) Аналогично пункту I) легко установить, что глаголы распадаются на классы, каждый из которых есть об"единение трех семейств, соответствующих разным лицам одних и тех же глаголов. Например, словоформы завидую, завидуешь, уступаю, уступаешь войдут в один класс, восхищаюсь, истекаю, восхищаешься, истекаешь - в другой, и т.п. Таким образом, классы глаголов - это "классы управления". Отдельный класс образуют безличные глаголы.

6) Все формы слова который, очевидно, образуют один класс K.

7) Наконец, отдельный класс $З$ образует запятая.

Чтобы получить типы языка с парадигмами (V, L, Γ) , нужно - ввиду его однородности - об"единить классы, взаимозамещаемые в языке $L_{K_{L,\Gamma}}$. Посмотрим, какие классы взаимозамещаемы.

I) Все четыре класса существительных взаимозамещаемы. Покажем, например, что $М' \Rightarrow Ж\,(L_{K_{L,\Gamma}})$. Для остальных пар классов существительных доказательство будет таким же.

Пусть X, Y - цепочки над $V_{K_{L,\Gamma}}$ такие, что $XМ'Y \in L_{K_{L,\Gamma}}$. Это означает существование словоформы $a \in М'$ и цепочек x, y над V , для которых $\Theta_{K_{L,\Gamma}}(x) = X$, $\Theta_{K_{L,\Gamma}}(y) = Y$ и $xay \in L$. Пусть b - существительное женского рода, имеющее тот же падеж, число и a . Если цепочка xay не содержит слов, согласованных с a в роде, т.е. если x не оканчивается прилагательным, а y не начинается запятой и словом который (-ого), то $xby \in L$,откуда $XЖY \in L_{K_{L,\Gamma}}$. Пусть xay содержит слова, согласованные с a в роде. Рассмотрим один из наиболее сложных случаев, когда $x = x'c$, где c - прилагательное, и $y =$, которого y'. Тогда $X = X'\Pi$, $Y = ЗKY'$, где $X' = \Theta_{K_{L,\Gamma}}(x')$, $Y' = \Theta_{K_{L,\Gamma}}(y')$. Если теперь d - прилагательное женского рода в том же падеже , что и c (т.е. в том же, что и a и b), то цепочка

$x'db$, которую $y' \in L$; но $d \in \Pi$ и которую $\in K$, так что
$$XЖY = X'\Pi ЖЗKY' = \Theta_{K_{L,\Gamma}}(x'db, \text{которую } y'),$$
откуда $XЖY \in L_{K_{L,\Gamma}}$. Например, если $x =$ я восхищаюсь породистым, $a =$ конем и $y =$, которого я вижу, то можно взять $b =$ собакой, $d =$ породистой.
В остальных случаях - аналогично.

Итак, все существительные принадлежат одному типу. В то же время очевидно, что классы существительных не взаимозамещаемы ни с какими другими классами, так что существительные в точности образуют тип.

2) Ясно, что класс прилагательных не взаимозамещаем ни с каким другим классом и поэтому является типом.

3) Рассуждая аналогично пункту I), легко установить, что местоимения образуют тип. (Согласуемыми словами являются в этом случае глаголы).

4) Класс наречий, очевидно, является типом.

5) Легко видеть, что два глагольных класса взаимозамещаемы тогда и только тогда, когда они состоят из глаголов, требующих одинакового числа дополнений и допускающих также одинаковое число дополнений. Так, $K_{L,\Gamma}$(завидую) $\Longleftrightarrow$ $K_{L,\Gamma}$(восхищаюсь) , поскольку оба эти класса состоят из глаголов, требующих одного дополнения и не допускающих более одного ; это и дает возможность в любой цепочке языка $L_{K_{L,\Gamma}}$, содержащей символ $K_{L,\Gamma}$(завидую), заменять его на $K_{L,\Gamma}$(восхищаюсь) и обратно.(Например, цепочка $K_{L,\Gamma}$(он) $K_{L,\Gamma}$(завидую) $K_{L,\Gamma}$(брату) принадлежит $L_{K_{L,\Gamma}}$, т.к. он завидует брату $\in L$; но $K_{L,\Gamma}$(брату) $= K_{L,\Gamma}$(братом) и он восхищается братом $\in L$, так что $K_{L,\Gamma}$(он) $K_{L,\Gamma}$(восхищаюсь) $K_{L,\Gamma}$(брату) $\in L_{K_{L,\Gamma}}$).
В то же время класс $K_{L,\Gamma}$(пишу) не замещаем на $K_{L,\Gamma}$(даю), т.к. пишу может употребляться с одним дополнением или без дополнения, а даю требует двух дополнений - например, в цепочке $K_{L,\Gamma}$(я) $K_{L,\Gamma}$(пишу) $\in L_{K_{L,\Gamma}}$ нельзя заменить $K_{L,\Gamma}$(пишу) на $K_{L,\Gamma}$(даю). (Однако $K_{L,\Gamma}$(даю) $\Longrightarrow$ $K_{L,\Gamma}$(пишу)).
Класс безличных глаголов, очевидно, не взаимозамещаем ни с каким другим классом.

Ясно, наконец, что взаимозамещаемость между глагольным и неглагольным классами быть не может.

Итак, глаголы распадаются на следующие типы: а) тип глаголов, не допускающих дополнений *): разговариваю, сплю,... ; б) тип глаголов, которые могут употребляться без дополнения или с одним дополнением, но не с двумя: работаю, радуюсь,... ; в) тип глаголов, которые могут употребляться как без дополнения, так и с одним или двумя дополнениями: пишу, стираю ,... ; г) тип глаголов, требующих одного дополнения, но не двух : уважаю, перехожу,...; д) тип глаголов, требующих одного или двух дополнений: наполняю, продаю,...; е) тип глаголов, требующих двух дополнений: даю, преподношу,... ; ж) тип безличных глаголов: светает, вечереет,... .

Замечания: а) A priori мог бы существовать еще тип глаголов, употребляющихся либо без дополнений, либо с двумя дополнениями ; но таких глаголов, кажется, нет. б) Разумеется, словарь V включает не все глаголы; в него не входят, например, глаголы, сильно управляющие предлогами.

6) Класс форм слова который, очевидно, является типом.

7) Отдельный тип образует запятая.

Рассмотренный пример позволяет сделать вывод, что типы - это множества слов, выполняющих в предложениях приблизительно одинаковые синтаксические функции. Иначе говоря, тип является аналогом части речи ; разбиение словаря на типы оказывается близким к традиционному разбиению на части речи, хотя и не совпадает с ним полностью. Что же касается классов, то это- множества слов, которые выполняют одинаковую синтаксическую функцию и при этом одинаковым образом согласуются с другими словами (с прилагательными в случае существительных, с глаголами в случае местоимений, с управляемыми существительными в случае глаголов).

Разумеется, выделение семейств, классов и типов во всем русском языке приведет к значительно более сложной картине, чем та, которая получилась в нашей упрощенной ситуации; однако общий лингвистической смысл понятий семейства, класса и типа будет приблизительно такой же. (См. по этому вопросу книгу И.И. Ревзина [II]).

* Напомним, что речь идет только о беспредложных дополнениях.

Глава III.

ПОРОЖДАЮЩИЕ ГРАММАТИКИ

§ II. Основные определения.

Понятие порождающей грамматики является частным случаем понятия исчисления. Исчислением, или формальной системой, в математической логике называют разрешение производить некоторые четко определенные операции над некоторыми четко определенными объектами /в отличие от алгоритма, представляющего собой предписание производить некоторые операции/.*) Описания операций исчисления называются его правилами вывода; если некоторый объект получается из какой-то совокупности объектов применением правил вывода /любое число раз/, то говорят, что указанный объект выводим из этой совокупности.

В частности, порождающая грамматика - это исчисление, объектами которого служат цепочки над некоторым словарем, а правила вывода интерпретируются как грамматические правила некоторого языка, т.е. как описание свойств синтаксической структуры предложений этого языка /ср. § 5/; множество цепочек, выводимых в исчислении из некоторой фиксированной цепочки, понимается как совокупность грамматически правильных предложений данного языка.

Перейдем к формальным определениям соответствующих понятий.

Порождающей грамматикой называется упорядоченная четверка

$\Gamma = \langle V, V_1, Пр, S \rangle$, где: 1-2) V и V_1 - непересекающиеся конечные множества, называемые соответственно основным и вспомогательным словарями /их элементы называются соответственно основными и вспомогательными символами/; 3) Пр.- элемент V_1 ,

*) См. например: А.А.Марков, Теория алгорифмов, М.-Л., 1954, стр. 205, А.И.Мальцев, Алгоритмы и рекурсивные функции, М.,1965, стр.306.

называемый <u>начальным символом</u>; 4) S - конечное множество цепочек вида $\varphi \to \psi$, где φ, ψ - цепочки над $V \cup V_1$ и $\to$ - символ, не принадлежащий $V \cup V_1$. Эти цепочки называются <u>правилами</u>, а само S - <u>схемой</u>.

В настоящей главе мы будем называть порождающие грамматики просто <u>грамматиками</u>. Говоря о цепочках, будем, если не оговорено противное, подразумевать цепочки над $V \cup V_1$.

Цепочка η <u>непосредственно выводима</u> из цепочки ξ в грамматике Γ (обозначение $\xi \vDash \eta(\Gamma)$ или просто $\xi \vDash \eta$), если $\xi = \omega_1 \varphi \omega_2$, $\eta = \omega_1 \psi \omega_2$ и $\varphi \to \psi$ - одно из правил грамматики Γ.

Последовательность цепочек $(\tau_0, \tau_1, \ldots, \tau_n)$ называется <u>выводом</u> τ_n из τ_0 в грамматике Γ, если для каждого $i = 1, 2, \ldots n$ $\tau_{i-1} \vDash \tau_i(\Gamma)$.

Цепочка η <u>выводима</u> из цепочки ξ в грамматике Γ /обозначение: $\xi \vdash \eta(\Gamma)$ или просто $\xi \vdash \eta$/, если существует вывод η из ξ в Γ.

<u>Языком, порождаемым грамматикой Γ</u> (обозначение $L(\Gamma)$), называется множество цепочек над V, выводимых в Γ из $Пр$.

В главной интерпретации V есть множество словоформ некоторого конкретного языка, V_1 - множество грамматических категорий /таких, как "существительное", "группа существительного", "группа существительного в именительном падеже", "глагол", "группа глагола" и т.п./. В частности, Пр. - это категория "/грамматически правильное/ предложение". Применение каждого правила вывода - это замена некоторой цепочки категорий и/или словоформ другой цепочкой, которая может выполнять такие же синтаксические функции, что и первоначальная цепочка /например, если одно из правил имеет вид $S' \to AS$, где S' - группа существительного в некотором определенном падеже, числе и роде, и S, A - соответственно существительное и прилагательное в том же падеже, числе и роде, то правая часть этого правила может выполнять те же синтаксические функции, что и левая, но имеет более конкретную форму/. В частности, цепочки, выводимые из символа $Пр$, выполняют те же синтаксические функции, что и сам этот символ, т.е. являются грамматически правильными предложениями. Таким образом, $L(\Gamma)$ - это множество грамматически правильных предложений соответствующего языка.

<u>Пример</u>. Пусть $\Gamma = \langle V; V_1, Пр, S \rangle$, где $V = \{a, b, c\}$; $V_1 = \{Пр, A, B, C, Д\}$; $S = \{Пр \to ДBAC, Д \to BA, BABA \to a, aC \to cbb, A \to a, B \to b, C \to c, ДBA \to c, A \to bb, cc \to \lambda\}$.

Тогда, например, $ДBAC \vDash cC(\Gamma)$, $bccA \vDash bA(\Gamma)$.

Последовательность (Пр., ДВАС, ВАВАС, аС, ас) является выводом в Г. В данном примере число всех выводов, начинающихся с Пр., конечно, так что и язык $L(\Gamma)$ конечен; именно, $L(\Gamma) = \{$вавас, вввввас, вавввс, вввввввс, вавсвв, вввсвв, сс, свв, ас, $\lambda\}$.

Грамматика, каждое правило которой имеет вид $\eta_1 A \eta_2 \to \eta_1 \xi \eta_2$, где η_1, η_2 - произвольные цепочки, А - вспомогательный символ и ξ - непустая цепочка, называется <u>грамматикой непосредственно составляющих</u> /сокращенно <u>нс-грамматикой</u>/.

Для любой грамматики Γ и любой цепочки $\xi \in L(\Gamma)$ вывод ξ из Пр. в Г/"история порождения предложения"/ может интерпретироваться как некоторая информация о синтаксической структуре предложения ξ.*) Такая интерпретация особенно наглядна в случае нс-грамматик, т.к. с каждым выводом цепочки из начального символа в нс-грамматике естественным образом ассоциируется некоторая система составляющих этой цепочки в смысле § 2. Построение указанной системы основывается на следующих вспомогательных понятиях.

Пусть $\Gamma = \langle V, V_1, Пр, S \rangle$ - нс-грамматика и $\alpha = (Пр = \omega_0, \omega_1, \ldots \omega_n)$ - вывод в Г.

Для каждого $i = 0, 1, \ldots n-1$ цепочки ω_i и ω_{i+1} можно представить соответственно в виде

$$\omega_i = \varphi_i \psi_i A_i \eta_i \zeta_i, \quad \omega_{i+1} = \varphi_i \psi_i \xi_i \eta_i \zeta_i,$$

где A_i - вспомогательный символ и $\psi_i A_i \eta_i \to \psi_i \xi_i \eta_i$ - правило грамматики Г./Это правило мы будем для краткости обозначать $П_i$ /. Положим $\ell(\varphi_i \psi_i) = t_i$, $\ell(\xi_i) = u_i$.

Последовательность $((t_0, П_0), (t_1, П_1), \ldots, (t_{n-1}, П_{n-1}))$ назовем <u>способом проведения</u> вывода α. Вообще говоря, вывод может иметь более одного способа проведения. Если способ проведения вывода α фиксирован и для любого $i = 0, 1, \ldots n$

---

*) Разумеется, возможны предложения, имеющие более одного вывода в данной грамматике /"синтаксически омонимичные"/.

$\omega_i = a_{i1} a_{i2} \ldots a_{il_i}$, где a_{ij} - элементы $V \cup V_1$, то, очевидно, при любом $i = 0, 1, \ldots, n-1$ и при любом $j > t_i + u_i$ имеет место равенство $a_{ij} = a_{i+1, j+u_i-1}$. Мы будем теперь называть вхождение символа в ω_{i+1} /или, по терминологии § 3, точку цепочки ω_{i+1} / $(a_{i+1,q}, q)$ непосредственным потомком точки $(a_{i,p}, p)$ цепочки ω_i, если выполняется одно из трех условий: I) $p \leq t_i$, $q = p$; 2) $p = t_i + 1$, $t_i + 1 \leq q \leq t_i + u_i$; 3) $p > t_i + 1$, $q = p + u_i - 1$. (В случаях I) и 3), очевидно, $a_{ip} = a_{i+1,q}$). Иначе говоря, непосредственные потомки заменяемой точки - это все точки вставляемого вместо неё отрезка, а остальные точки имеют в качестве непосредственных потомков свои точные копии. Далее, при $0 \leq i_1 < i_2 \leq n$ мы будем говорить, что точка Δ_2 цепочки ω_{i_2} является потомком точки Δ_1 цепочки ω_{i_1} /или что Δ_1 является предком Δ_2 /, если существует такая последовательность $\Xi_{i_1}, \Xi_{i_1+1}, \ldots, \Xi_{i_2}$, что каждая Ξ_l есть точка цепочки ω_l, для каждого $i = i_1, i_1+1, \ldots, i_2$ Ξ_{i+1} является непосредственным потомком Ξ_i и при этом $\Xi_{i_1} = \Delta_1$, $\Xi_{i_2} = \Delta_2$.

Пусть теперь $\Delta = (\alpha_{ij}, j)$ - произвольная точка цепочки ω_i. Если $i < n$, обозначим через Φ_Δ множество всех точек цепочки ω_n, являющихся потомками точки Δ; при $i = n$ положим $\Phi_\Delta = \Delta$. Легко видеть, что Φ_Δ заполняет отрезок; поэтому можно отождествить Φ_Δ с вхождением некоторой цепочки в ω_n. Множество всех таких Φ_Δ обозначим через C. Ясно, что если Ξ - потомок Δ, то Φ_Δ покрывает Φ_Ξ. Далее, если $\Delta \neq \Xi$, Δ не есть потомок Ξ и Ξ не есть потомок Δ, то Φ_Δ и Φ_Ξ не перекрываются. Действительно, пусть Δ - точка цепочки ω_g и Ξ - точка цепочки ω_h, где $g \leq h$. Тогда существует такая точка Δ' цепочки ω_g, что Ξ либо есть потомок Δ', либо совпадает с Δ'. Очевидно, $\Delta' \neq \Delta$. Ясно, что Φ_Δ не перекрывается с $\Phi_{\Delta'}$ и тем более с Φ_Ξ. Итак, С удовлетворяет условию 2) определения системы составляющих. Кроме того, С содержит все точки цепочки ω_n и вхождение $(\omega_n, 1) = \Phi_{\Delta_0}$, где Δ_0 - единственная точка цепочки $\omega_0 = Пр$; таким образом, условие I) тоже выполняется.

Мы будем говорить, что полученная система составляющих С ассоциирована с данным выводом $\mathfrak{a}$ и данным способом его проведения.

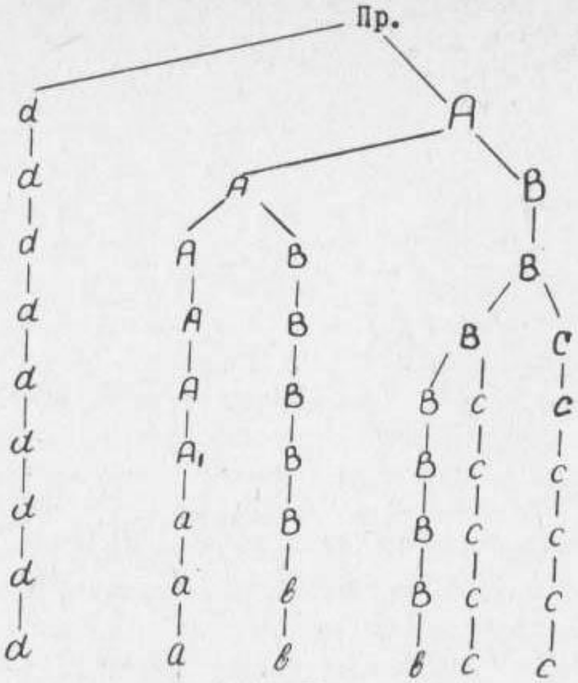
Пример. Пусть $Г = \langle V, V_1, Пр, S \rangle$, где $V = \{a, b, c, d, e\}$; $V_1 = \{Пр, A, A_1, B\}$ и S состоит из правил:

(I) Пр. $\to dA$
(П) Пр. $\to eA$
(Ш) $dA \to dAB$
(IУ) $eA \to eABB$
(У) $B \to Bc$
(УI) $A \to A_1$
(УП) $A_1 \to a$
(УШ) $B \to b$

Последовательность (Пр., dA, dAB, $dABB$, $dABBc$, $dABBcc$, dA_1BBcc, $daBBcc$, $dabBcc$, $dabbcc$} является, очевидно, выводом в Г. Этот вывод имеет единственный способ проведения, а именно: (/0,I/, /1,Ш/, /1,Ш/, /3,У/, /3,У/, /1,УI/, /1,УП/, /2,УШ/, /3,УШ/) (для краткости правила заменены номерами). Ассоциированная система составляющих имеет следующий вид /в скобочном представлении/: $(d((ab)((bc)c)))$

Заметим, что в системе составляющих, ассоциированной с выводом и способом его проведения, каждой неатомной составляющей /и, возможно, также некоторым атомным/ может быть приписан "синтаксический тип" - вспомогательный символ, от вхождения которого эта составляющая "происходит". Некоторые составляющие могут иметь более одного типа. Так, в нашем примере составляющая $(dabbcc, 1)$ имеет тип Пр.; (аввсс, 2) - тип А; (ав, 2) - тип А; (а,2) - типы А и A_I; (в,3), (вс,4) и (в,4) имеют тип В; (d,1),(с,5) и (с,6) не имеют типов.

Вывод в нс-грамматике, способ его проведения и ассоциированная с ними система составляющих могут быть представлены графической схемой, в которой цепочки, образующие вывод, записываются одна под другой и каждая точка соединяется линиями со своими потомками. Такая схема для рассмотренного нами примера приведена на черт.I2.



Черт.12

В заключение этого параграфа дадим еще два важных определения.

Грамматика называется <u>контекстно-свободной</u> /сокращенно кс-грамматикой/, если каждое её правило имеет вид $A \to \xi$, где A - вспомогательный символ и ξ - непустая цепочка.

Очевидно, всякая кс-грамматика является нс-грамматикой.

Грамматика называется <u>автоматной</u> /сокращенно <u>а-грамматикой</u>/, если каждое её правило имеет вид $A \to bB$ или $A \to b$, где $A, B \in V_1$ и $b \in V$.

Очевидно, всякая а-грамматика является кс-грамматикой.

В а-грамматике каждое предложение порождается слева направо.

Языки, порождаемые нс-, кс- и а-грамматиками, называются соответственно <u>нс</u>-, <u>кс</u>- и <u>а-языками</u>.

Примеры кс- и а-грамматик будут приведены в двух следующих параграфах.

§ 12. Абстрактные примеры.

В большинстве примеров этого параграфа будут явно выписываться лишь схемы грамматик. Основной словарь каждой грамматики будет состоять /кроме тех случаев, когда он явно выписывается/ из всех встречающихся в её правилах строчных латинских букв, а вспомогательный - из всех встречающихся в правилах прописных латинских букв и символа Пр., который всегда будет начальным. Для каждой грамматики указывается порождаемый ею язык, но доказательство того, что именно этот язык порождается данной грамматикой, либо только намечается, либо - в наиболее очевидных случаях - вовсе опускается.

<u>Пример 1.</u> а) Произвольный конечный язык $\{x_1, \ldots, x_k\}$, не содержащий пустой цепочки, порождается кс-грамматикой со схемой: $\{ Пр \to x_i, \; i = 1, \ldots, k$

б) Тот же язык может быть порожден а-грамматикой, которая строится следующим образом. Пусть для каждого $i = 1, 2, \ldots, k$ $x_i = a_{i1} a_{i2} \ldots a_{is_i}$, где a_{ij} - элементарные символы. Тогда схема нужной грамматики получается как объединение систем правил $S_1, S_2, \ldots, S_k$, где

$$S_i = \begin{cases} A_{i1} \to a_{i1} A_{i2} \\ A_{i2} \to a_{i2} A_{i3} \\ \ldots\ldots\ldots\ldots \\ A_{i,s_i-1} \to a_{is_i-1} A_{is_i} \\ A_{is_i} \to a_{is_i} \end{cases}$$

<u>Пример 2.</u> Назовем <u>полным языком</u> над словарем $V = \{a_1, \ldots, a_k\}$ (обозначение $F'(V)$) множество всех непустых цепочек над V.

Этот язык порождается грамматикой Γ_0 со схемой:

$$\begin{cases} Пр. \to a_i\, Пр, & i = 1, \ldots, k \\ Пр. \to a_i, & i = 1, \ldots, k \end{cases}$$

<u>Пример 3.</u> Язык $L_1 = \{a^{3n+1} / n = 0, 1, \ldots\}$ *) порождается а-грамматикой Γ_1 со схемой:

*) Для любой цепочки ω /в частности, состоящей из одного символа/ ω^k всегда будет означать $\underbrace{\omega\omega \ldots \omega}_{k \text{ раз}}$

$$\begin{cases} \text{Пр.} \longrightarrow aA \\ A \longrightarrow aB \\ B \longrightarrow a\ \text{Пр.} \\ \text{Пр.} \longrightarrow a \end{cases}$$

Пример. 4. Язык $L_2 = \{a^k b^m / k,m = 1,2,\ldots\}$ порождается а-грамматикой Γ_2 со схемой:

$$\begin{cases} \text{Пр} \longrightarrow aA \\ \text{Пр.} \longrightarrow aB \\ A \longrightarrow aA \\ A \longrightarrow aB \\ B \longrightarrow bB \\ B \longrightarrow b \end{cases}$$

Пример 5. Язык $L_3=\{ab^k c^l b^m a / k,m = 0,1,2,\ldots; l=1,2,\ldots\}$

порождается а-грамматикой Γ_3 со схемой:

$$\begin{cases} \text{Пр.} \longrightarrow aB_1 \\ B_1 \longrightarrow bB_1 \\ B_1 \longrightarrow cC \\ B_1 \longrightarrow cB_2 \\ C \longrightarrow cC \\ C \longrightarrow cB_2 \\ B_2 \longrightarrow bB_2 \\ B_2 \longrightarrow a \end{cases}$$

Пример 6. Язык $L_4=\{a^n b^n / n=1,2,\ldots\}$ порождается кс-грамматикой Γ_4 со схемой:

$$\begin{cases} \text{Пр.} \longrightarrow a\,\text{Пр}\,b \\ \text{Пр.} \longrightarrow ab \end{cases}$$

Пример 7. а) Язык $L_5=\{a^n b^n a^m / n,m=1,2,\ldots\}$ порождается кс-грамматикой Γ_5 со схемой:

$$\begin{cases} \text{Пр.} \longrightarrow \text{Пр}\,a \\ \text{Пр.} \longrightarrow Aa \\ A \longrightarrow aAb \\ A \longrightarrow ab \end{cases}$$

б) язык $L'_5=\{a^m b^n a^n / n,m=1,2,\ldots\}$ порождается кс-грамматикой Γ'_5 со схемой:

$$\begin{cases} \text{Пр.} \longrightarrow a\,\text{Пр.} \\ \text{Пр.} \longrightarrow aA \\ A \longrightarrow bAa \\ A \longrightarrow ba \end{cases}$$

Пример 8. Рассмотрим следующий вариант определения правильно построенной формулы /ппф/ алгебры логики: I) $p_1,\ldots,p_k$ /переменные высказывания/ суть ппф; 2) если α есть ппф, то $\neg(\alpha)$ есть ппф; 3) если $\alpha, \mathcal{L}$ - суть ппф и λ - один из символов $\&$, $\vee$, $\supset$, то $(\alpha)\lambda(\mathcal{L})$ есть ппф; 4) никаких других ппф нет. Множество L_6 всех ппф в смысле этого определения порождается кс-грамматикой Γ_6 над словарем $V=\{p_1,\ldots,p_k, \neg, \&, \vee, \supset,), (\}$ со схемой:

$$\begin{aligned} &\text{Пр.} \longrightarrow p_i, \quad i=1,\ldots,k. \\ &\text{Пр.} \longrightarrow \neg(\text{Пр.}) \\ &\text{Пр.} \longrightarrow (\text{Пр.})\lambda(\text{Пр.}), \lambda = \&, \vee, \supset \end{aligned}$$

Пример 9. Для произвольной непустой цепочки $x = a_{i_1}\ldots a_{i_s}$ положим $\hat{x}=a_{i_s}\ldots a_{i_1}$ /обращение x /. Тогда язык $L_7=\{x\hat{x}\}$, где x пробегает всевозможные непустые цепочки над словарем $V=\{a_1, a_2, \ldots a_k\}$, порождается кс-грамматикой Γ_7 со схемой:

$$\begin{cases} \text{Пр.} \longrightarrow a_i\,\text{Пр}\,a_i, & i=1,\ldots,k. \\ \text{Пр.} \longrightarrow a_i\,a_i, & i=1,\ldots,k. \end{cases}$$

Пример 10. Язык $L_8=F'(V)\setminus L_7$ порождается кс-грамматикой Γ_8 со схемой:

$$\begin{cases} \text{Пр.} \longrightarrow a_i\,\text{Пр}\,a_j, & i,j=1,\ldots,k. \\ \text{Пр.} \longrightarrow a_i, & i=1,\ldots,k. \\ \text{Пр.} \longrightarrow a_i A a_j, & i,j=1,\ldots,k;\ i\neq j \\ \text{Пр.} \longrightarrow a_i a_j, & i,j=1,\ldots,k;\ i\neq j \\ A \longrightarrow a_i A a_j, & i,j=1,\ldots,k \\ A \longrightarrow a_i a_j, & ij=1,\ldots,k. \end{cases}$$

Доказательство равенства $L(\Gamma_8)=L_8$: легко видеть, что при выводе в Γ_8 цепочки над V имеет место одна из следующих двух возможностей: либо сначала применяются n раз $(n=0,1,\ldots)$ правила I-й строки и затем /один раз/ правило 2-й строки; либо сначала применяются n раз $(n=0,1,\ldots)$ правила $1^{ой}$ строки, затем /один раз/ одно из правил 3-й или 4-й строки, далее - если предыдущее правило было из 3-й строки - m раз $(m=0,1,\ldots)$ правила 5-й строки, и, наконец, /один раз/ правило 6-й строки. Ясно, что в первом случае получаются все цепочки нечетной длины и только они, а во втором - все цепочки четной длины, не имеющие вида $x\hat{x}$, и только они.

Пример II. а) Пусть $L_9=\{a^n c a^{2n}/n=1,2,\ldots\}\cup$ $\cup\{b^{2n}cb^n/n=1,2\ldots\}\cup\{a^ncb^n/n=1,2,\ldots\}\cup\{b^nca^n/n=1,2\ldots\}$. L_9 порождается кс-грамматикой Γ_9 со схемой:

$$\begin{cases} \text{Пр.} \longrightarrow aAaa \\ \text{Пр.} \longrightarrow bbAb \\ \text{Пр.} \longrightarrow aAb \\ \text{Пр.} \longrightarrow bAa \\ aAa \longrightarrow aaAaaa \\ bAb \longrightarrow bbbAbb \\ aAb \longrightarrow aaAbb \\ bAa \longrightarrow bbAaa \\ A \longrightarrow c \end{cases}$$

б) Тот же язык порождается кс-грамматикой Γ_9' со схемой:

$$\begin{cases} \text{Пр.} \longrightarrow aAaa \\ \text{Пр.} \longrightarrow bbBb \\ \text{Пр.} \longrightarrow aCb \\ \text{Пр.} \longrightarrow bДa \\ A \longrightarrow aAaa \\ B \longrightarrow bbBb \\ C \longrightarrow aCb \\ Д \longrightarrow bДa \\ A \longrightarrow c \\ B \longrightarrow c \\ C \longrightarrow c \\ Д \longrightarrow c \end{cases}$$

Пример I2. Язык $L_{10}=\{a^nb^na^n/n=1,2,\ldots\}$ порождается грамматикой Γ_{10} со схемой /которую мы для наглядности разобьем на пять групп правил/:

$$/\text{гр.I}/\begin{cases} Пр \longrightarrow FA_0B_0C_0 \\ F \longrightarrow FABC \\ F \longrightarrow aBC \end{cases}$$

$$/\text{гр.II}/\begin{cases} BA \longrightarrow AB \\ CB \longrightarrow BC \\ CA \longrightarrow AC \\ BA_0 \longrightarrow A_0B \\ CA_0 \longrightarrow A_0C \\ CB_0 \longrightarrow B_0C \end{cases}$$

$$/\text{гр.III}/\begin{cases} CC_0 \longrightarrow C_0C_0 \\ BB_0 \longrightarrow B_0B_0 \\ AA_0 \longrightarrow A_0A_0 \end{cases}$$

$$/\text{гр.IV}/\begin{cases} aA_0 \longrightarrow aa \\ aB_0 \longrightarrow ab \\ bB_0 \longrightarrow bb \\ bC_0 \longrightarrow ba \\ aC_0 \longrightarrow aa \end{cases}$$

$$/\text{гр.V}/\ \{Пр \longrightarrow aba$$

Доказательство равенства $L(\Gamma_{10})=L_{10}$:
) Любая цепочка $a^nb^na^n$, $n=2,3,\ldots$ может быть ыведена в Γ_{10} из Пр. следующим образом: сначала с помощью прaил гр.I выводится $aBC(ABC)^{n-2}A_0B_0C_0$, затем-по равилам гр.II - $aA^{n-2}A_0B^{n-1}B_0C^{n-1}C_0$, далее - по равилам гр.III - $aA_0^{n-1}B_0^nC_0^n$ и, наконец - по правилам р.IУ - $a^nb^na^n$. Цепочка ава выводится по правилу гр.У.
) Пусть $x\neq aba$ - цепочка над $\{a,b\}$, выводимая з Пр. в Γ_{10} . Ясно, что в выводе x из Пр. обязательно применются правила гр.IУ. Перед началом применения этих правил це - очка может содержать только одно вхождение основного символа,а енно a, и притом в самом начале. Но чтобы к этой цепочке ожно было применить правила гр.IУ, она обязательно должна начиться подцепочкой aA_0 . А это, как видно из строения прaл гр. I-III, возможно лишь в случае, когда y имеет вид $aA_0^kB_0^lC_0^m$ действительно, приписывание символам А,В,С индекса 0 происходит лько справа налево и не может быть доведено до конца, если о дороге" встретится вхождение цепочки ВА,СВ или СА/. При этом но, что $m=l=k+1$. Итак, $y=aA_0^{m-1}B_0^mC_0^m$.
из такой цепочки по правилам гр.IУ можно получить только одну почку над $\{a,b\}$ - $a^mb^ma^m$.

Пример I3. Пусть $L_{11}=\{x b x\}$, где x – произвольная цепочка над $\{a_1, a_2, \dots, a_k\}$. L_{11} порождается грамматикой Γ_{11} со схемой:

$$\text{Пр.} \longrightarrow \text{Пр.}\, A_i a_i, \quad i=1,\dots,k$$
$$\text{Пр.} \longrightarrow b$$
$$a_i A_j \longrightarrow A_j a_i, \quad i,j=1,\dots,k$$
$$b A_i \longrightarrow a_i b, \quad i=1,\dots,k$$

Пример I4. Пусть $L_{12}=\{x_1 b x_2\}$, где x_1 и x_2 – произвольные различные непустые цепочки над $\{a_1, a_2, \dots, a_k\}$. L_{12} порождается кс-грамматикой Γ_{12} со схемой:

(I) Пр. $\longrightarrow$ Пр. a_i, $i=1,\dots,k$

(П) Пр. $\longrightarrow$ $A_i a_i$, $i=1,\dots,k$

(Ш) $A_i \longrightarrow a_j A_i a_l$, $i,j,l=1,\dots,k$

(IУ) $A_i \longrightarrow a_j B$, $i,j=1,\dots,k$, $i\neq j$

(У) $B \longrightarrow a_i B$, $i=1,\dots,k$

(УI) $B \longrightarrow b$

(УП) $A_i \longrightarrow b$, $i=1,\dots,k$

(УШ) Пр. $\longrightarrow C$

(IX) $C \longrightarrow a_i C a_j$, $i,j=1,\dots,k$

(X) $C \longrightarrow a_i Д$, $i=1,\dots,k$

(XI) $Д \longrightarrow a_i Д$, $i=1,\dots,k$

(XП) $Д \longrightarrow b$

Доказательство равенства $L(\Gamma_{12})=L_{12}$: Легко видеть, что при выводе цепочки над $\{a_1,\dots,a_k, b\}$ из Пр. в Γ_{12} последовательность применяемых правил всегда имеет одну из следующих трех форм:

1) I*, П, Ш*, IУ, У*, УI.

2) I*, П, Ш*, УП.

3) УШ, IX*, X, XI*, XП.

Здесь звездочкой помечены номера правил, которые могут применяться подряд сколько угодно раз, но могут и не применяться вовсе. Остальные правила применяются обязательно, но лишь по одному разу.

Нетрудно установить, что: в случае 1) получаются всевозможные цепочки вида $x' a_j y' b x'' a_i y''$, где x', x'', y', y'' – цепочки над $\{a_1,\dots,a_k\}$, $l(x')=l(x'')$ и $i\neq j$; в случае 2) получаются всевозможные цепочки вида $z' b z''$, где z', z'' – цепочки над $\{a_1,\dots,a_k\}$ такие, что $l(z')<l(z'')$; в случае 3) получаются всевозможные цепочки вида $z' b z''$, где z', z'' – цепочки над $\{a_1,\dots,a_k\}$ такие, что $l(z')>l(z'')$. Объединение этих трех множеств цепочек как раз и есть L_{12}.

Пример I5. Пусть $V=\{1\}$ и $L_{13}=\{\underbrace{11\dots1}_{n^2\ \text{раз}} \mid n=1,2,\dots\}$ /множество полных квадратов, записанных в "единичной системе счисления"/. L_{13} порождается грамматикой Γ_{13} со схемой:

(I) Пр. $\longrightarrow ABBДF$

(П) $BД \longrightarrow ДCB$

(Ш) $BC \longrightarrow CB$

(IУ) $AД \longrightarrow AAE$

(У) $EC \longrightarrow AE$

(УI) $EB \longrightarrow BE$

(УП) $EF \longrightarrow BBДF$

(УШ) $ДF \longrightarrow 1$

(IX) $B1 \longrightarrow 11$

(X) $A1 \longrightarrow 11$

(XI) Пр. $\longrightarrow 1$

Приведем пример вывода в грамматике Γ_{13}. /В кружках указаны номера применяемых правил/.

Пр

I $\Rightarrow AB^2ДF$

П $\Rightarrow ABДCBF$

П $\Rightarrow AД(CB)^2F$

Ш $\Rightarrow AДC^2B^2F$

IУ $\Rightarrow A^2EC^2B^2F$

У $\Rightarrow A^3ECB^2F$

У $\Rightarrow A^4EB^2F$

УI $\Rightarrow A^4BEBF$

УI $\Rightarrow A^4B^2EF$

см. след. стр.

VII → $A^4B^4ДF$

II

$A^4Д(CB)^4F$

III

$A^4ДC^4B^4F$

IV

$A^5EC^4B^4F$

V

A^9EB^4F

VI

A^9B^4EF

VII

$A^9B^6ДF$

VIII

A^9B^6I

IX

A^9I^7

X

I^{16}

Из приведенного примера видно , что вывод в $Г_{13}$ происхо- дит следующим образом: сначала из Пр. по правилу I выводится це- почка $AB^2ДF = A^{1^2}B^{2\cdot 1}ДF$, а затем выполняется про- извольное число циклов, на каждом из которых цепочка $A^{n^2}B^{2n}ДF$ преобразуется в $A^{(n+1)^2}B^{2(n+1)}ДF$. Цикл состоит в следующем: каждое вхождение B порождает "двойника" - вхождение C /правило II/, которое затем "переходит" влево /правило III/, после чего превращается в A /правило V/, причем до начала превраще- ний C в A появляется еще одно вхождение A /правило IV/; в резуль- тате число вхождений A становится равным $n^2+2n+1=(n+1)^2$ в конце цикла появляются еще два вхождения B /правило VII/. После окончания каждого очередного цикла можно вместо того, чтобы нача- новый цикл, применить правила VIII-X, превращающие $A^{n^2}B^{2n}ДF$ $I^{n^2+2n+1}=I^{(n+1)^2}$. Таким образом получается любая цепоч- ка вида $I^{(n+1)^2}$, и легко видеть, что только такие цепочки мо- гут быть выведены из Пр. по правилам I-X. Цепочка $I=I^{1^2}$ полу- чается по правилу XI. Итак, $L(Г_{13})=L_{13}$.

§ 13. Лингвистические примеры.

Пример I. Рассмотрим кс-грамматику, строящуюся следующим образом. Её основной словарь состоит из русских словоформ, а вспомогательный-из символов Пр., S^*_{xyzu}, S_{xyz}, S'_{xyz}, S''_{xyzu}, V^*_{xuz}, V_{xuz}, V'_{xuz}, Adv (x,y, z , u - переменные, множества значений которых указаны ниже). Вспомогательные символы интерпретируются так:

Пр. - предложение

S^*_{xyzu} - группа существительного рода x в падеже y , числе z и лице u /указание на лицо необходимо потому, что в число существительных мы включаем личные местоимения/.

S_{xyz} - неместоименное нарицательное существительное рода x в падеже y и числе z .

S'_{xyz} - неместоименное собственное существительное рода x в падеже y и числе z .

S_{xyzu} - местоименное существительное /личное местоимение/ рода x в падеже y , числе z и лице u .

V^*_{xuz} - группа глагола в прошедшем времени, роде x , лице u и числе z .

V_{xuz} - переходный глагол в прошедшем времени, роде x , лице u и числе z .

V'_{xuz} - глагол, управляющий предлогом за, в прошедшем времени, роде x , лице u и числе z .

Adv - обстоятельство образа действия.

Переменные x, y, z, u принимают следующие значения: x - I,2,3, интерпретируемые соответственно как мужской, женский и средний род; y - I,2,3,4,5,6 - соответственно именительный, родительный, дательный, винительный, творительный и предложный падеж; z - I,2 - соответственно единственное и множественное число; u - I,2,3 - соответственно I-е, 2-е и 3-е лицо. Подчеркнем, что выражения S_{xyz}, V_{xuz} и т.п. - не элементы вспомогательного словаря, а лишь сокращенные обозначения для групп элементов: S_{111}, S_{112} и т.д. Точно так же в записи схемы /см.ниже/ каждая строчка, содержащая символы с переменными индексами, есть сокращенная запись нескольких правил.

Схема грамматики имеет следующий вид:

(I) $\text{Пр} \longrightarrow S^{*}_{xyzu}\, V^{*}_{xuz}$

(П) $S^{*}_{xyz3} \longrightarrow S_{xyz}$

(Ш) $S^{*}_{xyz3} \longrightarrow S'_{xyz}$

(IУ) $S^{*}_{xyzu} \longrightarrow S''_{xyzu}$

(У) $S_{xyz} \longrightarrow S_{xyz}\, S_{x'2z'}$

(УI) $S_{xyz} \longrightarrow S_{xyz}\, S'_{x'2z'}$

(УП) $V^{*}_{xuz} \longrightarrow V^{*}_{xuz}\ \underline{\text{и}}\ V^{*}_{xuz}$

(УШ) $V^{*}_{xuz} \longrightarrow Adv\ V^{*}_{xuz}$

(IX) $V^{*}_{xuz} \longrightarrow V_{xuz}\, S^{*}_{x4zu}$

(X) $V^{*}_{xuz} \longrightarrow V'_{xuz}\ \underline{\text{за}}\ S^{*}_{x'5z'u'}$

(XI) $Adv \longrightarrow S_{x5z}$

(XП) $S_{1yz} \longrightarrow \underline{\text{народ}}_{yz}$; $S_{1yz} \longrightarrow \underline{\text{колокольчик}}_{yz}$

(XШ) $S'_{1yz} \longrightarrow \underline{\text{Пугачев}}_{yz}$

(XIУ) $S_{2yz} \longrightarrow \underline{\text{толпа}}_{yz}$

(XУ) $S''_{xyz1} \longrightarrow \underline{\text{я}}_{yz}$

(XУI) $V_{xuz} \longrightarrow \underline{\text{узнать}}_{xuz}$

(XУП) $V'_{xuz} \longrightarrow \underline{\text{бежать}}_{xuz}$

Примечания. I) x', z' - переменные, принимающие такие же значения, что и x, z соответственно. 2) Выражения $\underline{\text{народ}}_{yz}$, $\underline{\text{узнать}}_{xuz}$ и т.п. - сокращения для различных форм слов /например, $\underline{\text{народ}}_{11}$ означает народ, $\underline{\text{народ}}_{12}$ - народа и т.д./. 3) Ради простоты мы считаем, что слово я имеет любой из трех родов /правило XУ/. 4) Число правил такого типа, как XП-XУП, можно, очевидно, увеличивать совершенно механически, беря в качестве правых частей любые подходящие слова.

В числе предложений, выводимых в описанной грамматике из её начального символа Пр., имеется следующее: народ узнал колокольчик Пугачева и толпою бежал за нами. На черт. I3 приведена графическая схема вывода этого предложения. Слева в кружках - номера применяемых правил.

По чертежу I3 легко найти систему составляющих, ассоциированную с данным выводом /имеющим, очевидно, единственный способ проведения/, и для каждой составляющей определить её тип. Например, толпою бежал за нами - составляющая, имеющая тип V^{*}_{131}, т.е. группа глагола прошедшего времени, мужского рода, 3-го лица, единственного числа; колокольчик Пугачева - составляющая типа S_{141} /синтаксически эквивалентная неместоименному нарицательному существительному мужского рода в винительном падеже единственного числа/; толпою - составляющая типов Adv и S_{251}.

Пример 2. Построим а-грамматику с тем же основным словарем, что и в предыдущем примере, и вспомогательным словарем $\{S_{xyz}, A_{xyz}, V''_{xuz}, P\}$, где: S_{xyz} интерпретируется так же, как в предыдущем примере; A_{xyz} означает прилагательное рода x в падеже y и числе z ; V''_{xuz} - глагол, управляющий предлогом $\underline{\text{к}}$, в прошедшем времени, роде x, лице u и числе z ; P - предлог к /точнее, P есть грамматическая категория "предлог к"; символ P - вспомогательный, в то время как к - основной/. Начальным символом грамматики будет S_{211}.

Схема грамматики имеет следующий вид:

(I) $S_{211} \longrightarrow \underline{\text{кибитка}}\ V''_{231}$

(П) $V''_{xuz} \longrightarrow \underline{\text{подъехать}}_{xuz}\ P$

(Ш) $P \longrightarrow \underline{\text{к}}\ S_{x3z}$

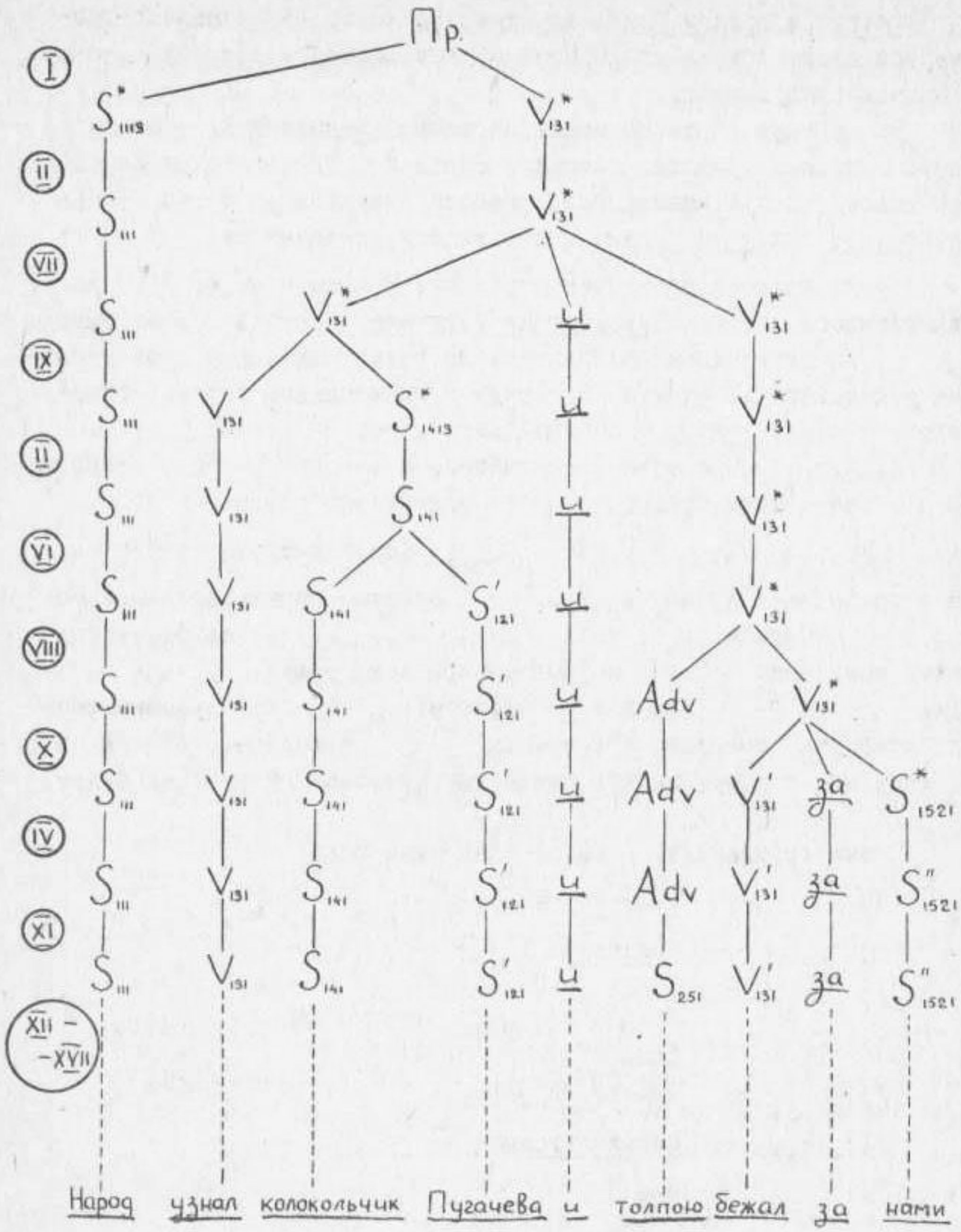
(IУ) $S_{3yz} \longrightarrow \underline{\text{крыльцо}}_{yz}\ S_{x'2z'}$

(У) $S_{3yz} \longrightarrow \underline{\text{крыльцо}}_{yz}\ A_{x'2z'}$

(УI) $A_{xyz} \longrightarrow \underline{\text{комендантский}}_{xyz}\ S_{xyz}$

(УП) $S_{1yz} \longrightarrow \underline{\text{дом}}_{yz}$

Пример вывода в этой грамматике:



Черт.13

S_{211}

I кибитка V''_{231}

II кибитка подъехала P

III кибитка подъехала к S_{331}

У кибитка подъехала к крыльцу A_{121}

УI кибитка подъехала к крыльцу комендантского S_{121}

УII кибитка подъехала к крыльцу комендантского дома

Следует обратить внимание на то, что естественная лингвистическая интерпретация вывода здесь не та, что в предыдущем примере. Именно, вхождение вспомогательного символа в одну из промежуточных цепочек вывода содержательно означает не тип "происходящей" от него составляющей, а тип того слова, которое возникает из этого вхождения на следующем шаге. Например, S_{331} – тип слова крыльцу, A_{121} – тип слова комендантского. Что же касается ассоциированной с выводом системы составляющих, то она в данном случае непосредственной лингвистической интерпретации не имеет.

Вообще, для а-грамматики ассоциированная с выводом система составляющих цепочки $a_1 a_2 \ldots a_n$ /где $a_1, \ldots, a_n$ – элементарные символы/ всегда имеет, как легко убедиться, вид $(a_1(a_2(a_3\ldots(a_{n-1}a_n)\ldots)))$. Такая система может быть естественной, как правило, только в случае, когда цепочка представляет собой не предложение, а некоторое более простое словосочетание. Например, такие системы естественны для элементарных именных групп русского языка /элементарная именная группа – это существительное со всевозможными словами, подчиненными ему и стоящими слева от него/.

§ I4. Обобщенные нс-грамматики

Определение нс-грамматики /стр. 51 / содержит требование непустоты цепочки ξ, которое на первый взгляд может показаться не очень существенным. На самом же деле это требование играет чрезвычайно важную роль – при отказе от него можно получить грамматику, порождающую любой язык, какой только вообще может быть порожден какой-либо грамматикой./Правда, нам еще неизвестно, не может ли любой язык, порождаемый грамматикой, быть порожден и нс-грамматикой; но в гл. V увидим, что класс нс-языков у́же класса языков, порождаемых произвольными грамматиками/.

Чтобы точно сформулировать указанное утверждение, введем следующие определения:

Две грамматики с одним и тем же основным словарем называются эквивалентными, если они порождают один и тот же язык.

Обобщенной нс-грамматикой /сокращенно-онс-грамматикой/ называется грамматика, каждое правило которой имеет вид $\eta_1 A \eta_2 \to \eta_1 \xi \eta_2$, где A-вспомогательный символ и ξ, η_1, η_2 - произвольные цепочки.

Теперь наше утверждение может быть сформулировано так:

Теорема 3.1. Для всякой грамматики можно эффективно построить эквивалентную ей онс-грамматику.

Доказательство. Пусть $\Gamma = \langle V, V_1, \Pi p, S \rangle$ - произвольная грамматика. Мы можем считать, что S не содержит правила $\Lambda \to \Lambda$ /ясно, что такое правило можно изъять, не меняя языка $L(\Gamma)$/. Каждому символу $a \in V$ сопоставим его "двойника" $\bar{a}$ - некоторый новый символ, не входящий в $V \cup V_1$. Множество всех "двойников" обозначим через $\bar{V}$. Построим грамматику $\Gamma' = \langle V, V_1', \Pi p, S' \rangle$ следующим образом: I) $V_1' = V_1 \cup \bar{V}$; 2) $S' = \bar{S} \cup S_1$, где: а) $\bar{S}$ получается из S заменой в каждом правиле каждого вхождения каждого символа $a \in V$ вхождением его "двойника"; б) S_1 состоит из всевозможных правил вида $\bar{a} \to a$, где $a \in V$. Очевидно, грамматика Γ' эквивалентна Γ и обладает тем свойством, что левые части ее правил не содержат основных символов.

Пусть $\Pi = \varphi \to \psi$ - некоторое правило грамматики Γ'. Правилу Π мы поставим в соответствие систему правил S_Π, которую будем строить в зависимости от вида Π следующим образом:

I) Если $l(\varphi) = 0$ (т.е. $\varphi = \Lambda$), S_Π состоит из одного правила $E \to \psi$, где E - символ, не входящий в $V \cup V_1'$.

2) Если $l(\varphi) = 1$, S_Π состоит из одного правила - самого Π.

3) Если $l(\varphi) > 1$, $\varphi = B_1 \ldots B_k$ $(B_1, \ldots, B_k \in V_1')$ то S_Π состоит из правил:

$$
\begin{array}{ll}
(I) & B_1 B_2 \ldots B_k \to F_1 B_2 \ldots B_k \\
\hdashline
(k-1) & F_1 \ldots F_{k-2} B_{k-1} B_k \to F_1 \ldots F_{k-2} F_{k-1} B_k \\
(k) & F_1 \ldots F_{k-1} B_k \to F_1 \ldots F_{k-1} F_k \\
(k+1) & F_1 F_2 \ldots F_k \to F_2 \ldots F_k \\
\hdashline
(2k-1) & F_{k-1} F_k \to F_k \\
(2k) & F_k \to \psi
\end{array}
$$

Здесь $F_1, \ldots, F_k$ - символы, не входящие в $V \cup V_1'$.

Пусть для каждого правила $\Pi \in S'$ построена система правил S_Π, причем все символы F_i, входящие в разные системы S_Π, соответствующие случаю 3), различны, а символ E во всех системах S_Π, соответствующих случаю I), один и тот же. Обозначим через S_1^* объединение всех S_Π для всех $\Pi \in S'$. Далее, через S_2^* обозначим множество всевозможных правил вида $\alpha \to \alpha E$ и $\alpha \to E \alpha$ для всех $\alpha \in V \cup V_1'$. Положим $S^* = S_1^* \cup S_2^*$. Кроме того, обозначим через W множество всех символов F_i, входящих в системы S_Π, и положим $V_1^* = V_1' \cup W \cup \{E\}$.

Пусть теперь $\Gamma^* = \langle V, V_1^*, \Pi p, S^* \rangle$. Из способа построения Γ^* ясно, что Γ^* - онс-грамматика. Покажем, что Γ^* эквивалентна Γ'.

I. Пусть $x \in L(\Gamma')$; тогда существует вывод в Γ' $(\Pi p = \omega_0, \omega_1, \ldots \omega_n = x)$. Пусть $\Pi_i = \varphi_i \to \psi_i$ - правило, применяемое при переходе от ω_{i-1} к ω_i $(i = 1, \ldots, n)$. Если $l(\varphi_i) > 1$, то, очевидно, ω_i можно получить из ω_{i-1}, применяя вместо Π_i правила системы S_{Π_i} /в том порядке, как они у нас записаны/. Если $l(\varphi_i) = 0$, возможны два случая: а) $\omega_{i-1} = \chi' \chi''$, $\omega_i = \chi' \psi_i \chi''$, где хотя бы одна из цепочек χ', χ'' непуста; б) $\omega_{i-1} = \Lambda$, $\omega_i = \psi_i$. В случае а) можно вместо Π_i применить сначала правило $\alpha \to \alpha E$ /где α - последний символ цепочки χ'/ или $\alpha \to E \alpha$ /где α - первый символ цепочки χ''/ и затем правило $E \to \psi_i$.

В случае б) имеем $i-1>0$,так что существует ω_{i-2} и $\omega_{i-2}\neq\Lambda$; правило Π_{i-1} в этом случае имеет вид $\omega_{i-2}\to\Lambda$.Применяя к ω_{i-2} сначала правило $\alpha\to E\alpha$ /где α - первый символ цепочки ω_{i-2} /, затем правила системы $S_{n_{i-1}}$ в нужном порядке и после этого правило $E\to\varphi_i$, получим цепочку $\varphi_i=\omega_i$. Наконец, если $\ell(\varphi_i)=1$, то правило Π_i само принадлежит S^* . Итак, мы можем каждый шаг данного вывода заменить серией шагов, состоящих в применении правил из S^* ; таким образом, цепочка $\omega_n=x$ выводима из Пр. в Γ^* ,т.е. $x\in L(\Gamma^*)$.

II. Положим $\tilde{\Gamma}=\langle V, V_i^*, \text{Пр}, S'\cup S^*\rangle$. Правило грамматики $\tilde{\Gamma}$ будем называть <u>новым</u>, если оно не входит в S'. Пусть $\mathfrak{A}=(\text{Пр}=\omega_0,\omega_1,\dots,\omega_n)$ - вывод в $\tilde{\Gamma}$ такой, что ω_n - цепочка над $V\cup V_i'$. Мы покажем, что если в выводе $\mathfrak{A}$ применяются новые правила, то можно построить другой вывод в $\tilde{\Gamma}$ с теми же первой и последней цепочками, в котором число применений новых правил будет меньше, чем в первоначальном. Отсюда, очевидно, будет следовать, что всякий вывод в Γ^*, начинающийся с Пр., может быть преобразован в вывод в Γ' с теми же первой и последней цепочками; доказательство эквивалентности Γ' и Γ^* будет тем самым завершено.

Пусть первое применение нового правила в выводе $\mathfrak{A}$ происходит при переходе от ω_{i-1} к ω_i . Обозначим это правило через π . Поскольку $\omega_0=\text{Пр}$ и ω_{i-1} получается из ω_0 только с помощью правил из S', ω_{i-1} есть цепочка над $V\cup V_i'$. Поэтому имеет место один из двух случаев: а) π имеет вид $\alpha\to E\alpha$ или $\alpha\to\alpha E$; б) π есть правило (I) некоторой системы S_n ,соответствующей случаю $\ell(\varphi)>1$.

Рассмотрим случай а). Поскольку ω_n состоит только из символов из $V\cup V_i'$, возникшее при переходе от ω_{i-1} к ω_i вхождение E на некотором шаге - при переходе от ω_{j-1} к ω_j, $j>i$, должно исчезнуть. Но исчезновение E может произойти только при применении правила $E\to\varphi\in S^*$. При переходе от ω_i к ω_{j-1} данное вхождение E никакими преобразованиями затрагиваться не может. Поэтому, если изъять из $\mathfrak{A}$ цепочку ω_i , а из всех цепочек $\omega_{i+1},\omega_{i+2},\dots,\omega_{j-1}$ вычеркнуть соответствующие вхождения E , то полученная последовательность цепочек будет

по-прежнему выводом в $\tilde{\Gamma}$; ω_j будет получаться в этом выводе из предыдущей цепочки применением правила $\Lambda\to\varphi\in S'$. Полученный вывод содержит на два применения новых правил меньше, чем первоначальный.

Перейдем к случаю б). Возникшее при переходе от ω_{i-1} к ω_i вхождение F_1 должно когда-нибудь исчезнуть; пусть это произойдет при переходе от ω_{j-1} к ω_j . Но исчезновение F_1 может произойти только при применении правила $(к+1)$ той же системы S_n ; а чтобы это правило можно было применить, между $i^м$ и $j^м$ шагами вывода должны быть применены все правила (2), (3), ...(к), и именно в этом порядке - применение их в ином порядке невозможно. Пусть правило (к) применяется на $m^м$ шаге, т.е. при переходе от ω_{m-1} к ω_m $(i<m<j)$. Ясно, что то вхождение цепочки $F_1F_2\dots F_{к-1}F_к$, которое возникает на m-м шаге из $F_1F_2\dots F_{к-1}B_к$, между m-м и j-м шагами не может участвовать ни в каких преобразованиях (правда,возможен случай, когда между двумя F вставляется некоторая цепочка в результате применения правила с пустой левой частью. Но эта цепочка до j -го шага должна исчезнуть, поэтому таких преобразований можно не делать). Поэтому мы можем все преобразования, происходящие между m-м и j-м шагами, без изменения результата произвести <u>до</u> $m^{\text{ого}}$ шага - иначе говоря, выполнить $m^{\text{й}}$ шаг непосредственно перед j-м . Далее, пусть правило (К-I) применяется на h-м шаге вывода. Возникающее на этом шаге вхождение цепочки $F_1F_2\dots F_{к-1}$ не может между h-м и $(j-1)$-м шагами участвовать ни в каких преобразованиях /на $(j-1)$-м шаге теперь применяется правило (к)/; что же касается вхождения $B_к$, следующего в цепочке ω_h за $F_1F_2\dots F_{к-1}$, то оно, вообще говоря, может между h-м и $(j-1)$-м шагами участвовать в преобразованиях, но только в таких, в результате которых после $F_1F_2\dots F_{к-1}$ снова появится $B_к$. Поэтому мы можем без изменения результата произвести все преобразования, происходящие между h-м и $(j-1)$-м шагами, до h -го шага, т.е. выполнить h -й шаг непосредственно перед $(j-1)$-м . Продолжая этот процесс, мы в конце концов получим новый вывод $\mathfrak{A}'$ в грамматике $\tilde{\Gamma}$ с теми же начальной и конечной цепочками, в котором правила (I), (2), ... , (к), (к+I) применяются подряд - на $(j-к)$-м, $(j-к+1)$-м,...,

..., j-м шагах. Очевидно, имеющееся в j-й цепочке этого вывода вхождение цепочки $F_2 \dots F_k$ должно исчезнуть, а это возможно только при последовательном применении правил (к+2), ..., (2к) той же системы S_Π. Аналогично предыдущему мы можем получить из $\mathfrak{A}'$ новый вывод $\mathfrak{A}''$ в $\tilde{\Gamma}$ с теми же начальной и конечной цепочками, в котором все шаги до j-го включительно – те же, что и в $\mathfrak{A}'$, а на шагах с номерами $j+1, \dots, j+k-1$ применяются соответственно правила (к+2), ..., (2к). Таким образо в выводе $\mathfrak{A}''$ правила системы S_Π применяются подряд на ша гах с номерами $j-k, j-k+1, \dots, j+k-1$, и мы можем заменить всю эту последовательность шагов одним применение правила $\Pi = B_1 \dots B_k \to \psi$. Получается новый вывод в $\tilde{\Gamma}$ с т ми же начальной и конечной цепочками, что и $\mathfrak{A}$, но содержащ на 2к применения новых правил меньше.

Итак, Γ^* эквивалентна Γ', а значит, и Γ. Доказа тельство закончено.

§ 15. Неукорачивающие грамматики.

Пусть $\varphi \to \psi$ – правило некоторой грамматики. Н зовем это правило <u>неукорачивающим</u>, если $\ell(\varphi) \le \ell(\psi)$.

<u>Неукорачивающей грамматикой</u> мы будем называть грамматику, в правила которой – неукорачивающие.

Очевидно, всякая нс-грамматика является неукорачивающей. Сл дующая теорема очень важна для выяснения объема понятия нс-языка.

<u>Теорема 3.2.</u> Для всякой неукорачивающей грамматики можно э фективно построить эквивалентную ей нс-грамматику.

<u>Доказательство.</u> Пусть $\Gamma = \langle V, V_1, \Pi p, S \rangle$ – неукора чивающая грамматика. Построим грамматику $\Gamma' = \langle V, V_1', \Pi p, S' \rangle$ тем же способом, который был применен в начале доказательства те ремы 3.1. Грамматика Γ' эквивалентна Γ, и левые части её п вил не содержат основных символов. Кроме того, из способа постро ния Γ' ясно, что, коль скоро Γ – неукорачивающая грамматика, Γ' также будет неукорачивающей.

Пусть $\Pi = \varphi \to \psi$ – произвольное правило грамматики Поставим ему в соответствие систему правил S_Π следующим обр зом:

1) Если $\ell(\varphi) = 0$, S_Π состоит из всевозможных правил вида $\alpha \to \alpha\psi$ и $\alpha \to \psi\alpha$, где $\alpha \in V \cup V_1'$.

2) Если $\ell(\varphi) > 0$, $\varphi = B_1 \dots B_k$, $\psi = C_1 \dots C_r$ $(k \le r)$, $B_1, \dots, B_k, C_1, \dots, C_r \in V \cup V_1'$, то S_Π состоит из правил:

$$
\begin{array}{ll}
(1) & B_1 B_2 \dots B_k \to F_1 B_2 \dots B_k \\
\hline
(k-1) & F_1 \dots F_{k-2} B_{k-1} B_k \to F_1 \dots F_{k-2} F_{k-1} B_k \\
(k) & F_1 \dots F_{k-1} B_k \to F_1 \dots F_{k-1} F_k \\
(k+1) & F_1 F_2 \dots F_{k-1} F_k \to C_1 F_2 \dots F_{k-1} F_k \\
\hline
(2k-1) & C_1 \dots C_{k-2} F_{k-1} F_k \to C_1 \dots C_{k-2} C_{k-1} F_k \\
(2k) & \begin{cases} C_1 \dots C_{k-1} F_k \to C_1 \dots C_{k-1} C_k C_{k+1} \dots C_r & (\text{если } k<r) \\ C_1 \dots C_{k-1} F_k \to C_1 \dots C_{k-1} C_k & (\text{если } k=r) \end{cases}
\end{array}
$$

Здесь $F_1, \dots, F_k$ – символы, не входящие в $V \cup V_1'$.

Пусть для каждого правила $\Pi \in S'$ построена система S_Π, причем все символы F_i, входящие в разные системы, попарно различны. Объединение всех S_Π по всем $\Pi \in S'$ обозначим через S^*. Множество всех символов F_i, входящих в системы S_Π, обозначим через W и положим $V_1^* = V_1' \cup W$.

Пусть теперь $\Gamma^* = \langle V, V_1^*, \Pi p, S^* \rangle$. Ясно, что Γ^* – нс-грамматика. Покажем, что Γ^* эквивалентна Γ'.

1. Пусть $x \in L(\Gamma')$. Тогда существует вывод в Γ' (Пр.$= \omega_0, \omega_1, \dots, \omega_n = x$). Пусть $\Pi_i = \varphi_i \to \psi_i$ – правило, применяемое при переходе от ω_{i-1} к ω_i. Если $\ell(\varphi_i) > 0$, то, очевидно, ω_i можно получить из ω_{i-1}, применяя вместо Π_i правила системы S_{Π_i} по порядку: (1), (2), ... , (2к). Если же $\ell(\varphi_i) = 0$, то, поскольку $\ell(\omega_{i-1}) > 0$ /грамматика Γ' – неукорачивающая!/, имеем $\omega_{i-1} = \chi' \chi''$, $\omega_i = \chi' \psi_i \chi''$, где хотя бы одна из цепочек χ', χ'' непуста; очевидно, ω_i можно получить из ω_{i-1}, применяя вместо Π_i правило $\alpha \to \alpha\psi$/где α – последний символ χ'/ или $\alpha \to \psi\alpha$/где α – первый символ χ''/. Итак, мы можем преобразовать данный вывод в вывод в грамматике Γ^*. Поэтому $x \in L(\Gamma^*)$.

П. Доказательство того, что всякая цепочка над $V \cup V_1'$, выводимая из Пр. в Γ^*, будет выводима из Пр. также и в Γ', совершенно аналогично доказательству соответствующей части теоремы 3.I /с тем очевидным упрощением, что применение правила вида $\alpha \to \alpha\psi$ или $\alpha \to \psi\alpha$, соответствующего правилу $\Lambda \to \psi$, теперь всегда можно непосредственно заменить применением последнего правила/. Мы опускаем эту часть рассуждения.

Итак, Γ^* эквивалентна Γ', а значит, и Γ. Теорема доказана.

Обратимся теперь к примерам I2 и I3 из § I2. Грамматики этих примеров - неукорачивающие. Поэтому порождаемые ими языки L_{10} и L_{11} являются нс-языками. Непосредственное построение для них нс-грамматик было бы весьма трудоемким делом, и грамматики получились бы намного более сложными.

§ I6. Грамматики без существенного укорачивания.

В ряде случаев теоремы 3.2 оказывается недостаточно, чтобы установить, что тот или иной язык является нс-языком. В этих случаях часто оказывается полезным более сильный критерий, который можно получить с помощью понятия грамматики без существенного укорачивания; это понятие мы сейчас введем.

Пусть Γ - произвольная грамматика и $\varphi \to \psi$ — некоторое её правило. Это правило мы будем называть заключительным, если $\psi \neq \Lambda$ и ни один символ, входящий в ψ, не входит в левую часть какого-либо правила Г. В частности, если левые части правил Г не содержат основных символов, то всякое правило, правая часть которого непуста и не содержит вспомогательных символов, будет заключительным.

Лемма 3.I. Для произвольной грамматики $\Gamma = \langle V, V_1, Пр., S \rangle$, левые части всех правил которой непусты, и произвольного вывода $(\omega_0, \omega_1, \dots, \omega_n)$ в Γ существует другой вывод в Γ с теми же начальной и конечной цепочками, в котором каждое применение незаключительного правила предшествует каждому применению заключительного правила.

Доказательство. Если данный вывод сам не обладает нужным свойством, то найдется число $i \, (1 \leq i \leq n)$ такое, что: I) при переходе от ω_{i-1} к ω_i применяется заключительное правило; 2) для некоторого $j > i$ при переходе от ω_{j-1} к ω_j применяется незаключительное правило. Пусть имеется всего t таких чисел i, и i_t - наибольшее из них. Тогда существует такое h, $1 \leq h \leq n - i_t$, что: а) на всех шагах с номерами $i_t+1, i_t+2, \dots, i_t+h$ применяются незаключительные правила*); б) либо $i_t+h=n$, либо на всех шагах с номерами $i_t+h+1, \dots, n$ применяются заключительные правила. Пусть правило, применяемое на шаге с номером i_t, есть $П = \varphi \to \psi$. Тогда $\omega_{i_t-1} = \chi'\varphi\chi''$, $\omega_{i_t} = \lambda'\psi\lambda''$, и поскольку $П$ - заключительное правило, а правил с пустой левой частью нет, выделенное вхождение ψ ни в каких дальнейших преобразованиях не участвует. Поэтому можно произвести шаг с номером i_t после шага с номером i_t+h, уменьшив тем самым число t на единицу. Повторяя это преобразование вывода, получим в конце концов $t=0$, т.е. вывод будет обладать нужным свойством.

Будем называть грамматику Γ грамматикой без существенного укорачивания, если каждое её незаключительное правило - неукорачивающее.

Теорема 3.3. Для всякой грамматики без существенного укорачивания можно эффективно построить эквивалентную ей нс-грамматику.

Доказательство. Пусть $\Gamma = \langle V, V_1, Пр, S \rangle$ - грамматика без существенного укорачивания. В силу теоремы 3.2 достаточно построить неукорачивающую грамматику, эквивалентную Γ.

Мы можем считать, что S не содержит правила $\Lambda \to \Lambda$, так что правые части всех правил Γ непусты. (В S не может быть правил вида $\varphi \to \Lambda$, $\varphi \neq \Lambda$, т.к. такие правила - -укорачивающие и незаключительные/. Кроме того, мы можем и левые части правил Γ считать непустыми. Если это не так, добавим к V_1 новый символ E и каждое правило вида $\Lambda \to \psi$ заменим на $E \to \psi$; кроме того, добавим к S всевозможные правила вида $\alpha \to \alpha E$ и $\alpha \to E\alpha$, где $\alpha \in V \cup V_1$ $(\alpha \neq E)$. Полученная грамматика эквивалентна Γ - это доказывается, как в теореме 3.I. Кроме того, новая грамматика будет, как и Γ, грам-

*) i-м шагом вывода мы называем переход от ω_{i-1} к ω_i.

матикой без существенного укорачивания, т.к. все новые правила - неукорачивающие. Обозначим через d максимум длин левых и правых частей правил Γ. Каждой непустой цепочке ω над $V \cup V_1$, длина которой меньше $2d$, поставим в соответствие некоторый новый символ $A_\omega \bar{\in} V \cup V_1$. Множество всех таких A_ω обозначим V_1^*.

Пусть теперь ω, θ - произвольные цепочки над $V \cup V_1$ такие, что: I) $\ell(\omega) < 4d$ 2) θ непосредственно выводима из ω с помощью некоторого незаключительного правила грамматики Γ. Паре (ω, θ) мы поставим в соответствие некоторую систему правил $S(\omega, \theta)$

При построении этой системы будем различать следующие случаи:

I) $\ell(\omega) < 2d$, $\ell(\theta) < 2d$;

2) $\ell(\omega) < 2d$, $2d \le \ell(\theta) < 3d$;

3) $2d \le \ell(\omega) < 4d$, $2d \le \ell(\theta) < 4d$;

4) $2d \le \ell(\omega) < 4d$, $4d \le \ell(\theta) < 5d$.

В случае I) $S(\omega, \theta)$ будет состоять из одного правила $A_\omega \to A_\theta$.

В случае 2) $S(\omega, \theta)$ будет состоять из одного правила $A_\omega \to A_{\theta_1} A_{\theta_2}$, где θ_1 и θ_2 - цепочки, удовлетворяющие условиям:

а) $\theta_1 \theta_2 = \theta$; б) $\ell(\theta_1) = d$.

В случае 3) $S(\omega, \theta)$ будет состоять из всевозможных правил вида $A_{\omega_1} A_{\omega_2} \to A_{\theta_1} A_{\theta_2}$, где $\omega_1, \omega_2, \theta_1, \theta_2$ - цепочки, удовлетворяющие условиям:
а) $\omega_1 \omega_2 = \omega$; б) $\theta_1 \theta_2 = \theta$; в) длина каждой из четырех цепочек не меньше d, но меньше $2d$.

В случае 4) $S(\omega, \theta)$ будет состоять из всевозможных правил вида $A_{\omega_1} A_{\omega_2} \to A_{\theta_1} A_{\theta_2} A_{\theta_3}$, где $\omega_1, \omega_2, \theta_1, \theta_2, \theta_3$ - цепочки, удовлетворяющие условиям: а) $\omega_1 \omega_2 = \omega$; б) $\theta_1 \theta_2 \theta_3 = \theta$; в) длина каждой из пяти цепочек не меньше d, но меньше $2d$.

Далее, произвольной цепочке ω над $V \cup V_1$ такой, что $0 < \ell(\omega) < 2d$, поставим в соответствие систему $T(\omega)$, состоящую из всевозможных правил вида $A_\omega \to \chi$, где χ либо выводима из ω с помощью одних только заключительных правил, либо совпадает с ω.

бо совпадает с ω.

Наконец, каждой цепочке ω над $V \cup V_1$ такой, что $d < \ell(\omega) < 4d$, поставим в соответствие систему $U(\omega)$, состоящую из всевозможных правил вида $A_{\omega_1} A_{\omega_2} \to \chi A_\theta$, где $\omega_1 \omega_2 = \omega$ и χ, θ - непустые цепочки, удовлетворяющие следующему условию: существует такая цепочка ψ, что $\psi\theta = \omega$ и χ либо выводима из ψ с помощью одних только заключительных правил, либо совпадает с ψ.

Обозначим теперь через S^* объединение всех построенных систем $S(\omega, \theta), T(\omega), U(\omega)$. Положим $\Gamma^* = \langle V, V_1^*, A_{\Pi p}, S^* \rangle$

Из способа построения S^* видно, что грамматика Γ^* - неукорачивающая. Покажем, что Γ^* эквивалентна Γ.

I. Пусть $\mathfrak{A} = (\omega_0, \omega_1, \dots, \omega_n)$ - вывод в Γ такой, что $\omega_0 = \Pi p$. В силу леммы 3.I можно считать, что для некоторого s, $0 \le s \le n$, выполняется следующее условие: на шагах вывода $\mathfrak{A}$ с номерами $1, \dots, s$ (если $s > 0$) применяются только незаключительные правила, а на шагах с номерами $s+1, \dots, n$ /если $s < n$ / - только заключительные. Каждую цепочку ω_i $(0 \le i \le s)$ мы представим в виде $\omega_i = \omega_{i1} \omega_{i2} \dots \omega_{it_i}$, $(*)$
где цепочки ω_{ij} удовлетворяют условиям:
(α) $\ell(\omega_{ij}) < 2d$; (β), если $t_i > 1$, то $\ell(\omega_{ij}) \ge d$

Представление $(*)$ мы определим индуктивно следующим образом:

I) ω_0 представляется как $\omega_{01} = \omega_0 = \Pi p$.

2) Пусть $0 \le i < s$ и для ω_i определено представление $(*)$, удовлетворяющее условиям (α) и (β). В силу условия (β) при переходе от ω_i к ω_{i+1} из цепочек $\omega_{i1}, \dots, \omega_{it_i}$ могут измениться самое большее две рядом стоящих цепочки, т.е. возможны следующие два случая:

I) $\omega_{i+1} = \omega_{i1} \dots \omega_{i,h-1} \theta \omega_{i,h+1} \dots \omega_{it_i}$;

2) $\omega_{i+1} = \omega_{i1} \dots \omega_{i,h-1} \theta \omega_{i,h+2} \dots \omega_{it_i}$

Здесь θ - некоторая цепочка, длина которой меньше $3d$ в случае (I) и меньше $5d$ в случае (2).

Представление $(*)$ для цепочки ω_{i+1} мы определим теперь так: (а) В обоих случаях положим $\omega_{i+1,1} = \omega_{i1}, \dots, \omega_{i+1,h-1} = \omega_{i,h-1}$ (б) В случае (I), если $l(\theta) = 2d$, положим $\omega_{i+1,h} = \theta$, $\omega_{i+1,h+1} = \omega_{i,h+1}, \dots, \omega_{i+1,t_{i+1}} = \omega_{i t_i}$.

/в/ случае (I), если $l(\theta) \geq 2d$, положим $\omega_{i+1,h} = \theta_1$, $\omega_{i+1,h+1} = \theta_2$, где $\theta_1\theta_2 = \theta$ и $l(\theta_1) = d$; кроме того, положим $\omega_{i+1,h+2} = \omega_{i,h+1}, \dots,$

$$\omega_{i+1,t_{i+1}} = \omega_{i+1,t_i+1} = \omega_{i,t_i}.$$

В случае (2), если $l(\theta) < 4d$, положим $\omega_{i+1,h} = \theta_1$, $\omega_{i+1,h+1} = \theta_2$, где $\theta_1\theta_2 = \theta$ и длины цепочек θ_1, θ_2 не меньше d, но меньше $2d$; цепочки $\omega_{i+1,h+2}, \dots, \omega_{i+1,t_{i+1}}$ определим, как в случае (в).(q)

В случае (2), если $l(\theta) > 4d$, положим $\omega_{i+1,h} = \theta_1$, $\omega_{i+1,h+1} = \theta_2$, $\omega_{i+1,h+2} = \theta_3$, где $\theta_1\theta_2\theta_3 = \theta$ и длины цепочек $\theta_1, \theta_2, \theta_3$ не меньше d, но меньше $2d$; кроме того, положим $\omega_{i+1,h+3} = \omega_{i,h+1}, \dots, \omega_{i+1,t_{i+1}} = \omega_{i+1,t_i+2} = \omega_{i,t_i}$.

Заметим, что представление (ж) определяется, вообще говоря, неоднозначно. Для наших целей годится любое представление, получаемое описанным способом.

Теперь для каждого $i = 0, 1, \dots, s$ положим $\Omega_i = A_{\omega_{i1}} \dots A_{\omega_{it_i}}$. Из только что описанного способа определения представления (ж) сразу следует, что при любом $i = 1, \dots, s$ Ω_i получается из Ω_{i-1} применением одного из правил, входящих в систему $S(\omega_{ih}, \theta)$ или $S(\omega_{ih}\omega_{i,h+1}, \theta)$. Поэтому цепочка Ω_s выводима в Γ^* из $\Omega_0 = A_{\text{Пр.}}$. В то же время ясно, что, поскольку ω_n выводима из ω_s с помощью одних только заключительных правил /в частности, ω_n может и совпадать с ω_s /, из Ω_s можно вывести ω_n с помощью правил, входящих в системы вида $T(\omega)$ и $U(\omega)$. /Заметим, что одних правил вида $T(\omega)$ для этого недостаточно - см. ниже пример/. Итак, цепочка ω_n выводима в грамматике Γ^* из $A_{\text{Пр.}}$.

П. Для произвольной цепочки ξ над словарем $V \cup V_1 \cup V_1^*$ мы будем обозначать через $\overline{\xi}$ цепочку, полученную из ξ заменой каждого вхождения каждого символа $A_\omega \in V_1^*$ вхождением соответствующей цепочки ω.

Пусть $\mathcal{L} = (\xi_0, \xi_1, \dots, \xi_n)$ - произвольный вывод в Γ^*. Из способа построения правил грамматики Γ^* ясно, что при

любом $i = 1, \dots, n$ цепочка $\overline{\xi_i}$ либо непосредственно выводима из $\overline{\xi_{i-1}}$ в грамматике Γ, либо просто совпадает с $\overline{\xi_{i-1}}$. Поэтому цепочка $\overline{\xi_n}$ либо выводима из $\overline{\xi_0}$ в Γ, либо совпадает с $\overline{\xi_0}$. В частности, если $\xi_0 = A_{\text{Пр.}}$ и ξ_n - цепочка над $V \cup V_1$, то цепочка $\xi_n = \overline{\xi_n}$ выводима в Γ из $\overline{\xi_0} = \text{Пр.}$. Доказательство закончено.

Пример. Пусть схема грамматики Γ имеет вид:

$$\begin{cases} \text{Пр.} \longrightarrow \text{Пр.}C \\ \text{Пр.} \longrightarrow \text{Пр.}D \\ \text{Пр.} \longrightarrow C \\ DC \longrightarrow CD \\ CD \longrightarrow a \\ C \longrightarrow b \end{cases}$$

Имеем $d = 2$

В ниже следующей таблице в первом столбце записан вывод цепочки aab из Пр. в грамматике Γ, во втором - соответствующий вывод в грамматике Γ^*; в третьем столбце для каждого шага вывода в Γ^* указано, к какой системе S, T или U принадлежит применяемое на этом шаге правило.

Пр.	$A_{\text{Пр.}}$	
Пр.C	$A_{\text{Пр.}C}$	$S(\text{Пр.}, \text{Пр.}C)$
Пр.CC	$A_{\text{Пр.}CC}$	$S(\text{Пр.}C, \text{Пр.}CC)$
Пр.CCC	$A_{\text{Пр.}C} A_{CC}$	$S(\text{Пр.}CC, \text{Пр.}CCC)$
Пр.$DCCC$	$A_{\text{Пр.}DC} A_{CC}$	$S(\text{Пр.}C, \text{Пр.}DC)$
Пр.$DDCCC$	$A_{\text{Пр.}D} A_{DC} A_{CC}$	$S(\text{Пр.}DC, \text{Пр.}DDC)$
Пр.$DDDCCC$	$A_{\text{Пр.}DD} A_{DC} A_{CC}$	$S(\text{Пр.}D, \text{Пр.}DD)$
Пр.$DDCDCC$	$A_{\text{Пр.}DD} A_{CD} A_{CC}$	$S(DC, CD)$
Пр.$DCDDCC$	$A_{\text{Пр.}DC} A_{DD} A_{CC}$	$S(\text{Пр.}DDCD, \text{Пр.}DCDD)$
Пр.$DCDCDC$	$A_{\text{Пр.}DC} A_{DC} A_{DC}$	$S(DDCC, DCDC)$
$CDCDCDC$	$A_{CDC} A_{DC} A_{DC}$	$S(\text{Пр.}DC, CDC)$
$aCDCDC$ $aaCDC$	$a\,a\,A_C A_{DC}$	$U(CDCDC)$
$aaaC$	$a\,a\,a\,A_C$	$U(CDC)$
$aaab$	$a\,a\,a\,b$	$T(C)$

Обратившись теперь к примеру I5 из § I2, мы увидим, что Γ_{13} - грамматика без существенного укорачивания, так что L_{13} - нс-язык. Впрочем, для L_{13} без особого труда можно было бы построить и неукорачивающую грамматику. Более существенным образом теорема 3.3 будет использована при доказательстве замкнутости класса нс-языков относительно пересечения /теорема 3.II/.

§ I7. Некоторые вспомогательные утверждения о контекстно-свободных грамматиках.

В § I4 мы видели, что отказ от требования непусты "ядер" пр вых частей правил в определении нс-грамматики приводит к существен ному расширению класса порождаемых языков. Естественно поставить вопрос: что произойдет, если аналогичным образом изменить определение кс-грамматики? Оказывается, что в этом случае существенного расширения класса языков не происходит. Мы дадим сейчас точную фо мулировку этого утверждения и докажем его.

Назовем обобщенной кс-грамматикой /окс-грамматикой/граммати ку, каждое правило которой имеет вид $A \to \omega$, где A-вспомогате ный символ и ω - произвольная цепочка.

Теорема 3.4. Для всякой окс-грамматики Γ может быть эффективно построена кс-грамматика Γ', порождающая язык $L(\Gamma) \setminus \{\Lambda\}$ /в частности, если $\Lambda \bar{\in} L(\Gamma)$, то Γ' эквивалентна Γ /.

Доказательство. Пусть $\Gamma = \langle V, V_1, \Pi p., S \rangle$ - окс-грамма тика. Назовем символ $A \in V_1$ исчезающим, если S содержит прави ло $A \to \Lambda$. Для произвольной непустой цепочки ω обозначим через $Q(\omega)$ множество всевозможных непустых цепочек, получающихся и ω вычеркиванием одного или нескольких вхождений исчезающих сим

волов; положим $R(\omega) = Q(\omega) \cup \{\omega\}$. Например, если символы A, B -исчезающие и C, D - неисчезающие, то

$$R(AACBD) = \{AACBD, ACBD, AACD, CBD, ACD, CD\};$$

$$R(AB) = \{AB, A, B\}.$$

Для каждого правила $\Pi = A \to \omega \in S$ обозначим через S_Π множество всевозможных правил вида $A \to \varphi$, где $\varphi \in R(\omega)$. Пусть теперь S' есть объединение множеств S_Π, построенных для всех правил $\Pi \in S$ с непустыми правыми частями. Положим $\Gamma' = \langle V, V_1, \Pi p., S' \rangle$. Покажем, что $L(\Gamma') = L(\Gamma) \setminus \{\Lambda\}$.

I. Докажем, что если $x \in L(\Gamma)$ и $x \neq \Lambda$, то $x \in L(\Gamma')$. Положим $\widetilde{\Gamma} = \langle V, V_1, \Pi p., S \cup S' \rangle$. Достаточно показать, что если вывод в $\widetilde{\Gamma}$ /с фиксированным способом проведения/, начинающийся с Пр. и заканчивающийся непустой цепочкой, содержит применения правил вида $A \to \Lambda$, то данный вывод и способ его проведения можно перестроить так, чтобы число применений таких правил стало меньше.

Пусть $\mathfrak{O} = (\Pi p. = \omega_0, \omega_1, \dots, \omega_n \neq \Lambda)$ — такой вывод, и пусть первое применение правила с пустой правой частью происходит на i -м шаге. Очевидно, $\omega_i \neq \Lambda$ и поэтому $i > 1$. Обозначим через λ ту точку цепочки ω_{i-1}, которая исчезает на i -м шаге, и через λ_h - предка этой точки в цепочке $\omega_h (h = 0, 1, \dots, i-2)$. Пусть h_0 - наибольшее из чисел $h = 0, 1, \dots, i-2$ таких, что на $(h+1)$ -м шаге происходит применение правила именно к точке λ_h, и при этом правая часть правила состоит более чем из одного символа./Например, пусть

$$\omega_0 = \Pi p., \quad \omega_2 = A\underline{B}C, \quad \omega_4 = ABB\underline{D},$$
$$\omega_1 = A\underline{B}, \quad \omega_3 = ABB\underline{C}, \quad \omega_5 = ABB;$$

те вхождения, к которым на соответствующих шагах применяются правила, подчеркнуты. Тогда $\lambda = (D, 4)$, $\lambda_3 = (C, 4)$, $\lambda_2 = (C, 3)$, $\lambda_1 = (B, 2)$, $\lambda_0 = (\Pi p., 1)$, $h_0 = 1$ /.

Такое h_0 обязательно существует - иначе было бы $l(\omega_{i-1}) = 1$, т. е. $\omega_i = \Lambda$. Пусть при переходе от ω_{h_0} к ω_{h_0+1} применяется правило $A \to \varphi$. Применим вместо него правило $A \to \varphi'$, где φ' получается из φ вычеркиванием той точки, которая при подстановке φ на место A в цепочку ω_{h_0} даёт Λ_{h_0+1} . Остальные шаги вывода $\mathfrak{O}$ оставим без изменения, за исключением i -го шага, который просто опустим. Очевидно, таким образом получится вывод в Γ с теми же начальной и конечной цепочками, но в этом выводе число применений правил с пустой правой частью будет на единицу меньше.

П. Очевидно, в произвольном выводе в грамматике Γ' каждое применение правила $\Pi' \in S_n$ $(\Pi' \neq \Pi)$ можно заменить применением правила Π с последующим применением одного или нескольких правил с пустыми правыми частями.

Теорема доказана.

Докажем еще несколько лемм, которые понадобятся нам в дальнейшем.

Лемма 3.2. Для всякой кс-грамматики можно эффективно построить эквивалентную ей кс-грамматику, начальный символ которой не входит в правые части правил.

Доказательство. Пусть $\Gamma = \langle V, V_1, \Pi p, S \rangle$ -кс-грамматика. Присоединим к V_1 новый символ $\mathcal{I}$, заменим в правой части каждого правила Γ все вхождения Пр. вхождениями $\mathcal{I}$ и для каждого правила Пр $\to \omega$ добавим к S правило $\mathcal{I} \to \omega$. Полученная так грамматика будет, очевидно, эквивалентна Γ .

Для дальнейшего заметим, что если Γ - а-грамматика, то и новая грамматика - автоматная.

Лемма 3.3. Для всякой кс-грамматики $\Gamma = \langle V, V_1, \Pi p, S \rangle$ можно эффективно построить эквивалентную ей кс-грамматику $\Gamma' = \langle V, V_1', \Pi p', S' \rangle$ такую, что длины правых частей всех правил Γ' , не имеющих вида Пр.$' \to a$, где $a \in V$, больше единицы.

Доказательство. Будем предполагать, что Γ удовлетворяет условию предыдущей леммы.

Допустим, что S содержит правила вида $A \to B$ *) ,где $B \in V_1$ — иначе говоря, что существуют такие $C, \mathcal{D} \in V_1$, что $C \vdash \mathcal{D}\ (\Gamma)$. Для каждой пары вспомогательных символов $C, \mathcal{D}$ таких, что $C \vdash \mathcal{D}\ (\Gamma)$ и каждого правила $\mathcal{D} \to \omega \in S$, правая часть которого не есть вспомогательный символ, добавим к схеме S новое правило $C \to \omega$. После этого выбросим из S все правила вида $A \to B$, где $B \in V_1$. Полученная таким образом кс-грамматика Γ'' будет, очевидно, эквивалентна Γ . Ясно также, что она вместе с Γ удовлетворяет условию леммы 3.2. При

*) При этом можно, разумеется, считать, что $A \neq B$ - правило $A \to A$ можно выбросить, не меняя языка.

этом Γ'' строится по Γ эффективно, т.к. отношение $\vdash$ на V_1 эффективно проверяемо. /Действительно, если $A \vdash B$, $A, B \in V_1$, то существует такая последовательность $A = A_0, A_1, \dots, A_n = B$, что $A_{i-1} \vDash A_i$ и все A_i попарно различны; но длина этой последовательности не больше мощности множества V_1 , так что все такие последовательности можно найти перебором/.

Остается выбросить из схемы грамматики Γ'' правила вида $A \to a$, где $a \in V$ и $A \neq \Pi p.$ /если такие имеются/. Это можно сделать, выбросив из схемы Γ'' все такие правила и одновременно добавив к ней всевозможные правила, получаемые из правил Γ'' следующим образом: каждое правило $B \to \omega$ преобразуется всеми возможными способами в правила вида $B \to \omega'$, где ω' получается из ω заменой всех или некоторых вхождений символов $A_1, \dots, A_t$, для которых в схеме Γ'' имеются правила $A_1 \to a_{11}, \dots, A_1 \to a_{1s_1}, \dots, A_t \to a_{t1}, \dots, A_t \to a_{ts_t}$, вхождениями соответствующих символов $a_{11}, \dots, a_{ts_t}$ в любых возможных комбинациях /разумеется, A_1 заменяется только на $a_{11}, \dots, a_{1s_1}$ и т.д./. Полученная так кс-грамматика Γ' эквивалентна Γ'' , а, значит, и Γ . Лемма доказана.

Примечания: (I) Разумеется, Γ' вместе с Γ удовлетворяет условию леммы 3.2. (2) Из доказательства леммы 3.3 непосредственно ясно, что в формулировке этой леммы можно заменить кс-грамматику Γ окс-грамматикой /тогда и Γ' будет окс-грамматикой/.

Лемма 3.4. Для произвольной кс-грамматики $\Gamma = \langle V, V_1, \Pi p, S \rangle$ можно эффективно построить эквивалентную ей кс-грамматику $\Gamma^* = \langle V, V_1^*, \Pi p^*, S^* \rangle$, каждое правило которой имеет либо вид $A \to BC$, где $A, B, C \in V_1^*$, либо вид $A \to a$, где $A \in V_1^*$ $a \in V$.

Доказательство. Мы будем считать, что Γ удовлетворяет условию леммы 3.3. Сопоставим каждому символу $a \in V$ его "двойника" - новый символ $\bar{a} \,\bar{\in}\, V \cup V_1$. Множество всех "двойников" обозначим через $\bar{V}$. Построим кс-грамматику $\Gamma' = \langle V, V_1', \Pi p, S' \rangle$ следующим образом: 1) $V_1' = V_1 \cup \bar{V}$; 2) $S' = S_1' \cup S_2' \cup S_3'$, где: а) S_1' состоит из всех правил вида $\Pi p. \to a$ $(a \in V)$, входящих в S ; б) S_2 получается из $S \setminus S_1'$ заменой в каждом пра -

виле каждого вхождения каждого символа $a \in V$ вхождением его "двойника" $\overline{a}$; в) S_3' состоит из всевозможных правил вида $\overline{a} \to a$, где $a \in V$. Очевидно, Γ' эквивалентна Γ и обладает тем свойством, что правая часть каждого её правила состоит либо из двух или более вспомогательных символов /правила из S_2' /, либо из одного основного символа /правила из S_1' и S_3' /.

Пусть теперь $\Pi = A \to B_1 \dots B_k \in S'$ и $k > 1$ $(A, B_1, \dots, B_k \in V_1')$. Если $k > 2$, обозначим через S_Π систему правил:

$$\begin{cases} A \to B_1 F_1 \\ F_1 \to B_2 F_2 \\ \dots\dots\dots \\ F_{k-3} \to B_{k-2} F_{k-2} \\ F_{k-2} \to B_{k-1} B_k, \end{cases}$$

где $F_1, \dots, F_{k-2}$ - новые символы. При $k=2$ будем считать, что S_Π состоит из одного правила Π. Обозначим через S_2^* объединение всех систем S_Π для всех правил $\Pi \in S_2'$, причем все символы F, входящие в разные системы S_Π, будем считать попарно различными. Множество всех символов F обозначим через W. Если положить теперь $\Gamma^* = \langle V, V_1' \cup W, \Pi p., S_2^* \cup S_1' \cup S_3' \rangle$, то очевидно, что Γ^* эквивалентна Γ и удовлетворяет нужному условию.

Пусть $\mathfrak{A} = (\omega_0, \omega_1, \dots, \omega_n)$ - вывод в некоторой кс-грамматике, способ проведения которого фиксирован. Мы будем называть $\mathfrak{A}$ *бесповторным*, если не существует таких i и j, $0 \le i < j \le n$, и такого вспомогательного символа A, что некоторое вхождение μ символа A в ω_j является потомком некоторого вхождения λ того же символа A в ω_i, причем на $(i+1)$-м шаге вывода заменяется как раз вхождение λ.

Лемма 3.5. Длина бесповторного вывода*) в кс-грамматике Γ, начинающегося одноэлементной цепочкой, не может превышать $(d+1)^{p-1}$, где p - мощность вспомогательного словаря грамматики Γ и d - максимум длин правых частей правил Γ.

Доказательство проведем индукцией по p. При $p=1$ $V_1 = \{\Pi p.\}$ бесповторный вывод, начинающийся одноэлементной цепочкой, всегда имеет вид $(\Pi p., \omega)$, где ω - цепочка над основным словарем, так что длина вывода равна $1 = (d+1)^0 = (d+1)^{p-1}$.

*) *Длиной* вывода $(\omega_0, \omega_1, \dots, \omega_n)$ называется число n.

Пусть лемма доказана для грамматик, вспомогательные словари которых содержат $p-1$ символ, и пусть $\Gamma = \langle V, V_1, \Pi p., S \rangle$ - кс-грамматика такая, что V_1 содержит p символов. Пусть $\mathfrak{A} = (A = \omega_0, \omega_1, \dots, \omega_n)$ - бесповторный вывод в Γ и $A \in V_1$. Пусть $\omega_1 = B_1 B_2 \dots B_k$ $(B_1, \dots, B_k \in V_1)$. Тогда каждую цепочку ω_i $(i = 2, \dots, n)$ можно представить в виде $\omega_i = \omega_{i1} \omega_{i2} \dots \omega_{ik}$, где для каждого $j = 1, \dots, k$ ω_{ij} есть потомок B_j. Положим $\mathfrak{A}_j' = (B_j = \omega_{1j}, \omega_{2j}, \dots, \omega_{nj})$. Последовательность $\mathfrak{A}_j'$ может, вообще говоря, содержать повторяющиеся цепочки; выбросив из $\mathfrak{A}_j'$ все повторения рядом стоящих цепочек, получим новую последовательность $\mathfrak{A}_j$, которая, очевидно, будет выводом в Γ. Пусть длина этого вывода равна n_j. Ясно, что $n = n_1 + \dots + n_k + 1$. Но поскольку вывод $\mathfrak{A}$ - бесповторный, ни одна цепочка, входящая в $\mathfrak{A}_j$, не содержит A; поэтому $\mathfrak{A}_j$ можно рассматривать как вывод в грамматике $\langle V, V_1 \setminus \{A\}, B_j, S_A' \rangle$, где S_A' получается из S выбрасыванием правил, содержащих А. Очевидно, вывод $\mathfrak{A}_j$ - также бесповторный. По индуктивному предположению $n_j \le (d+1)^{p-2}$, откуда $n \le k \cdot (d+1)^{p-2} + 1 \le d \cdot (d+1)^{p-2} + 1 \le$

$$\le (d+1)(d+1)^{p-2} = (d+1)^{p-1}.$$

§ 18. Распознавание пустоты и конечности кс-языка.

Пусть $\Gamma = \langle V, V_1, \Pi p., S \rangle$ -кс-грамматика. Будем говорить, что символ $A \in V_1$ $(A \ne \Pi p)$ удовлетворяет *первому условию существенности*, если в Γ выводима из Πp некоторая цепочка, содержащая вхождение A. Будем считать, кроме того, что символ Пр. всегда удовлетворяет первому условию существенности. Будем говорить, что символ $A \in V_1$ удовлетворяет *второму условию существенности*, если из A выводима в Γ некоторая цепочка над V. Символ $A \in V_1$ будем называть *существенным*, если он удовлетворяет обоим только что сформулированным условиям.

Ясно, что цепочка, входящая в вывод какой-либо цепочки над V из Пр., может содержать только существенные вспомогательные символы. Поэтому язык, порождаемый кс-грамматикой, не изменится, если из её вспомогательного словаря выбросить несущественный символ, а из схемы - все правила, в которые этот символ входит.

Лемма 3.6. Каждое из условий существенности эффективно проверяемо.

Доказательство. I. Пусть $\Gamma = \langle V, V_1, Пр., S \rangle$ - кс-грамматика, и пусть символ $A \in V_1$ $(A \neq Пр.)$ удовлетворяет первому условию существенности. Пусть $\sigma = (Пр. = \omega_0, \omega_1, \dots, \omega_n)$ - кратчайший из выводов, начинающихся символом Пр. и оканчивающихся цепочками, содержащими вхождения А. /Если таких кратчайших выводов несколько, берем любой из них/. Допустим, что $n>1$. Пусть λ - некоторое вхождение А в ω_n и λ_i $(i=0,1,\dots,n-1)$ — предок λ в цепочке ω_i. Заметим, что для любого $i=0,1,\dots,n-1$ на $(i+1)$-м шаге заменяется как раз вхождение λ_i. Действительно, предположим, что для некоторого i вхождение μ_i, заменяемое на $(i+1)$ -м шаге, отлично от λ_i. Пусть, например, λ_i - вхождение символа C, μ_i - вхождение символа D и $\omega_i = \varphi C \psi D \chi$, где выделены как раз нужные вхождения С и Д. Тогда $\omega_n = \varphi'\theta_1\psi'\theta_2\chi'$, где $\varphi', \theta_1, \psi', \theta_2, \chi'$ — потомки соответственно цепочек $\varphi, C, \psi, D, \chi$. При этом θ_1 содержит A. Но тогда цепочка $\omega' = \varphi'\theta_1\psi' D \chi'$ тоже содержит A, и ω' выводится из Пр. короче, чем ω_n - чтобы получить вывод ω' из Пр., нужно просто опустить в выводе σ все шаги, на которых заменяются вхождение μ_i символа D в ω_i и потомки этого вхождения.

Пусть теперь A_i $(i=0,1,\dots,n-1)$ - тот символ, вхождением которого является λ_i. Нетрудно видеть, что символы $A_0, A_1, \dots, A_{n-1}$ попарно различны. Действительно, пусть для некоторых i, j $(0 \le i < j \le n-1)$ имеет место $A_i = A_j$. Тогда $\omega_i = \varphi_1 A_i \varphi_2$,

$$\omega_j = \varphi_1\psi_1 A_j \psi_2\varphi_2 = \varphi_1\psi_1 A_i \psi_2 \varphi_2,$$

т.е. $A_i \vdash \psi_1 A_i \psi_2$, и, далее, $\omega_n = \varphi_1\psi_1\theta_1 A \theta_2 \psi_2 \varphi_2$, т.е. $A_i \vdash \theta_1 A \theta_2$. Но тогда цепочка $\varphi_1\theta_1 A \theta_2\varphi_2$ будет выводима из $\omega_i = \varphi_1 A_i \varphi_2$, а, значит, и из Пр., и вывод этой цепочки из Пр. будет иметь длину $n-(j-i) < n$.

Итак, мы показали, что вывод $(\omega_0, \omega_1, \dots, \omega_{n-1})$ является бесповторным. Отсюда по лемме 3.4 $n-1 \le (d+1)^{p-1}$.

Таким образом, если символ А удовлетворяет первому условию существенности, то длина кратчайшего вывода из Пр. цепочки, содержащей А, не превосходит числа $c = (d+1)^{p-1}+1$, эффективно определяемого по грамматике Г. Отсюда вытекает следующий алгоритм проверки первого условия существенности для заданного символа $A \in V_1$: Нужно вычислить c и проделать все выводы из Пр., длины которых не превосходят c; если в результате одного из этих выводов получится цепочка, содержащая вхождение символа А, то А удовлетворяет первому условию существенности; в противном случае A этому условию не удовлетворяет.

II. Пусть $\Gamma = \langle V, V_1, Пр, S \rangle$ - кс-грамматика. Сопоставим некоторым символам из $V \cup V_1$ числа, которые назовем рангами этих символов, следующим образом: 1) Всем символам из V, и только им, приписываем ранг 0. 2) Если $i > 0$ и для каждого числа $j = 0, 1, \dots, i-1$ понятие символа ранга j определено, будем приписывать символу А ранг i, если S содержит хотя бы одно правило $A \to \omega$ такое, что ω состоит целиком из символов рангов $0, 1, \dots, i-1$ и содержит хотя бы один символ ранга $i-1$.

Очевидно, найдется такое число i_0, что никакому символу не удастся приписать ранг i_0+1. Если при этом окажется, что ранги приписаны не всем символам из $V \cup V_1$, то каждому из оставшихся символов припишем ранг ∞.

Нетрудно показать, что каждый символ конечного ранга удовлетворяет второму условию существенности. В самом деле: 1) символы ранга 1 этому условию удовлетворяют, т.к. для каждого такого символа А имеется правило $A \to \omega$, где ω - цепочка над V. 2) Если доказано, что символы рангов $1, 2, \dots, i-1$ удовлетворяют условию, и А есть символ ранга i, то имеется правило $A \to B_1 \dots B_k$, где $B_1, \dots, B_k$ - символы рангов, меньших i. По индуктивному предположению для каждого B_j имеется такая цепочка ω_j над V, что $B_j \vdash \omega_j$ или $B_j = \omega_j$; отсюда $A \vdash \omega_1 \dots \omega_k$.

С другой стороны, если $(A = \omega_0, \omega_1, \dots, \omega_n)$ - вывод в Γ, и если при этом $A = \omega_0$ - символ бесконечного ранга, то и каждая цепочка ω_i содержит хотя бы один символ бесконечного ран -

га - это доказывается очевидной индукцией по i. Поэтому из символа бесконечного ранга не выводима никакая цепочка над V.

Итак, символ из V_1 удовлетворяет второму условию существенности тогда и только тогда, когда он имеет конечный ранг. Но процедура приписывания рангов эффективна, так что второе условие существенности эффективно проверяемо.

Будем называть кс-грамматику <u>приведенной</u>, если все её вспомогательные символы - существенные.

Из леммы 3.6 и замечания, предпосланного её формулировке, немедленно вытекает

<u>Лемма 3.7.</u> Для всякой кс-грамматики Г, порождающей непустой язык, может быть эффективно построена эквивалентная ей приведенная кс-грамматика Γ'. Если при этом Γ удовлетворяет условию леммы 3.2, 3.3 или 3.4, то и Γ' удовлетворяет этому условию.

Вспомогательный символ А кс-грамматики Г мы будем называть <u>циклическим</u>, если из A выводима в Γ цепочка, содержащая вхождение A. <u>Циклической кс-грамматикой</u> будем называть кс-грамматику, вспомогательный словарь которой содержит хотя бы один циклический символ.

<u>Лемма 3.8.</u> Свойство цикличности вспомогательного символа кс-грамматики эффективно проверяемо.

<u>Доказательство.</u> Проверка цикличности символа A происходит в точности так же, как проверка выполнения для этого символа первого условия существенности /см. доказательство леммы 3.6./, с той только разницей, что вместо выводов из Пр. рассматриваются выводы из А.

Понятие циклической кс-грамматики является очень важным ввиду следующей теоремы:

<u>Теорема 3.5.</u> Приведенная кс-грамматика, схема которой не содержит правил вида А→В, где $B \in V_1$, тогда и только тогда порождает бесконечный язык, когда она является циклической.

<u>Доказательство.</u> I. Пусть $\Gamma = \langle V, V_1, \text{Пр}, S \rangle$ - приведенная циклическая кс-грамматика без правил вида А→В, $B \in V_1$, и пусть $A \in V_1$ - циклический символ. Поскольку S не содержит правил вида А→В, где $B \in V_1$, соотношение $A \vdash A(\Gamma)$ невозможно; поэтому существуют такие цепочки φ и ψ, что $A \vdash \varphi A \psi (\Gamma)$ и хотя бы одна из этих цепочек непуста. По первому условию существенности имеются такие цепочки ω_1, ω_2, что Пр. $\vdash \omega_1 A \omega_2$ или Пр.$= \omega_1 A \omega_2$. По второму условию существенности имеются цепочки $x, y, z, \omega_1, \omega_2$ над V, выводимые соответственно из /или совпадающие с/ $A, \varphi, \psi, \omega_1$ и ω_2; при этом из цепочек y и z хотя бы одна непуста. Но из соотношения $A \vdash \varphi A \psi (\Gamma)$ следует, что при любом $n = 1, 2, \ldots$ будет иметь место $A \vdash \varphi^n A \psi^n (\Gamma)$. Поэтому из Пр. при любом n будет выводима цепочка $\omega_1 \varphi^n A \psi^n \omega_2$, а значит, и цепочка $\omega_1 y^n x z^n \omega_2$. Но цепочек такого вида - поскольку y или z непуста - бесконечно много.

II. В нециклической кс-грамматике все выводы, очевидно, бесповторные. Но по лемме 3.5 число бесповторных выводов, начинающихся символом Пр., конечно; значит, конечно и число цепочек, выводимых из Пр.

Из теоремы 3.5 и лемм 3.3, 3.7 и 3.8 немедленно вытекает алгоритм, позволяющий по любой кс-грамматике распознать, является ли порождаемый ею язык конечным. Именно, нужно сначала привести грамматику к виду, описанному в формулировке леммы 3.3, затем построить эквивалентную ей приведенную кс-грамматику и, наконец, для этой последней проверить, является ли она циклической. Итак, доказана

<u>Теорема 3.6.</u> Свойство кс-грамматики порождать конечный язык эффективно проверяемо.

Имеет место также

<u>Теорема 3.7.</u> Свойство кс-грамматики порождать пустой язык эффективно проверяемо.

<u>Доказательство.</u> Очевидно, язык, порождаемый кс-грамматикой Γ, будет непустым тогда и только тогда, когда её начальный символ является существенным; а это по лемме 3.6 эффективно проверяемо.

§ 19. <u>Необходимое условие контекстной свободности языка</u>

<u>Теорема 3.8.</u> Для любого бесконечного кс-языка L существуют такие числа r и s, что каждая цепочка $\omega \in L$, длина которой больше r, может быть представлена в виде $\omega = z_1 y_1 x y_2 z_2$ так, что: I) хотя бы одна из цепочек y_1, y_2 непуста; 2) $\ell(y_1 x y_2) \leq s$; 3) при любом n =I,2,3, ... цепочка

$w_n = z_1 y_1^n x y_2^n z_2$ принадлежит L .

<u>Доказательство.</u> Пусть L - бесконечный кс-язык и $\Gamma = \langle V, V_1, \text{Пр.}, S\rangle$ - кс-грамматика, порождающая L и удовлетворяющая условию леммы 3.3. Положим $r = d\cdot(d+1)^{p-1}$, $s = d^2\cdot(d+1)^{p-1}$. Пусть $w \in L = L(\Gamma)$ и $\ell(w) > r$. Рассмотрим какой-либо вывод w из Пр. : $\mathfrak{A} = (\text{Пр.} = w_0, w_1, \ldots, w_n = w)$. Очевидно, разность $\ell(w_i) - \ell(w_{i-1}) < d$ $(i = 1, \ldots, n)$. Но $\ell(w_n) = [\ell(w_n) - \ell(w_{n-1})] + \ldots + [\ell(w_1) - \ell(w_0)] + \ell(w_0)$, так что $\ell(w_n) < d\cdot n + \ell(w_0) = d\cdot n + 1$. Поэтому $r < d\cdot n$, т.е. $n > \frac{r}{d} = (d+1)^{p-1}$. Отсюда по лемме 3.5 следует, что вывод $\mathfrak{A}$ не может быть бесповторным. Значит, существуют такие числа i, j $(0 \le i < j \le n)$ и такой символ A, что некоторое вхождение μ символа A в w_j есть потомок некоторого вхождения λ символа A в w_i и при этом на $(i+1)$-м шаге вывода заменяется как раз вхождение λ. Фиксируем соответствующие i, j, A, λ, μ, причем таким образом, чтобы i было наибольшим числом, для которого существуют j, A, λ и μ с нужными свойствами. Имеем теперь $w_i = \theta_1 A \theta_2$, $w_j = \theta_1 \psi_1 A \psi_2 \theta_2$, где $A \vdash \psi_1 A \psi_2$ (Γ), причем в силу условия леммы 3.3 хотя бы одна из цепочек ψ_1, ψ_2 непуста. Цепочку $w = w_n$ можно представить теперь в виде $w = z_1 y_1 x y_2 z_2$, где каждая из цепочек z_1, z_2, y_1, y_2, x соответственно выводима из /или совпадает с/ $\theta_1, \theta_2, \psi_1, \psi_2, A$; при этом, очевидно, хотя бы одна из цепочек y_1 и y_2 непуста. Но при любом $n = 1, 2, 3, \ldots$ из A выводима цепочка $\psi_1^n A \psi_2^n$, так что $\text{Пр.} \vdash \theta_1 A \theta_2 \vdash$ $\vdash \theta_1 \psi_1^n A \psi_2^n \theta_2 \vdash z_1 y_1^n x y_2^n z_2$.

Остается оценить $\ell(y_1 x y_2)$. В силу нашего выбора числа i вывод $(w_{i+1}, \ldots, w_n)$ является бесповторным. Пусть φ_h $(h = i+1, \ldots, n)$ - подцепочка цепочки w_h, состоящая из всех потомков вхождения λ символа A в w_i. Очевидно, $\varphi_n = y_1 x y_2$. Пусть теперь $\varphi_{i+1} = B_1 \ldots B_k$, где $B_1, \ldots, B_k \in V \cup V_1$; поскольку φ_{i+1} есть правая часть некоторого правила грамматики Γ, имеем $k \le d$. Каждую цепочку φ_h $(h = i+1, \ldots, n)$ можно представить в виде $\varphi_h = \varphi_{h_1} \ldots \varphi_{h_k}$, где $B_f \vdash \varphi_{h_f}$ $(f = 1, \ldots k)$. Если из последовательности $(B_f, \varphi_{i+1,f}, \ldots, \varphi_{nf})$ выбросить повторения рядом стоящих цепочек, то эта последовательность станет выводом в Γ- разумеется, бесповторным. По лемме 3.5 длины этого вывода не превосходит $(d+1)^{p-1}$, откуда $\ell(\varphi_{nf}) \le (d+1)^{p-1}\cdot d$ и $\ell(y_1 x y_2) = \ell(\varphi_{n1} \ldots \varphi_{nk}) \le (d+1)^{p-1}\cdot d\cdot k \le$ $\le (d+1)^{p-1}\cdot d^2 = s$

Доказательство закончено.

Приведем примеры применения теоремы 3.8.

<u>Пример I.</u> Рассмотрим язык L_{10} примера I2 из § I2. Допустим, что L_{10} - кс-язык. Тогда при всяком достаточно большом k цепочка $w = a^k b^k a^k \in L_{10}$ должна представляться в виде $w = z_1 y_1 x y_2 z_2$ так, что хотя бы одна из цепочек y_1, y_2 непуста и при любом n $w_n = z_1 y_1^n x y_2^n z_2 \in L_{10}$. Заметим, что y_1 может содержать вхождения только одного из символов a, b . Действительно, если бы y_1 содержало вхождения a и b, то уже цепочка w_2 содержала бы вхождения подцепочки $a^{s_1} b^{t_1} a^{s_2} b^{t_2}$ или $b^{t_1} a^{s_1} b^{t_2} a^{s_2}$, а цепочки языка L_{10} таких подцепочек не содержат. То же верно для y_2 . Следовательно, y_1 целиком содержится в одной из трех частей w- первом вхождении a^k, вхождении b^k или втором вхождении a^k, и то же верно для y_2 ; значит, одна из этих трех частей не перекрывается ни с y_1, ни с y_2 . Пусть, например, первое вхождение a^k не перекрывается ни с y_1, ни с y_2, а вхождение b^k содержит ту из цепочек y_1, y_2, которая непуста. Тогда $w_2 = a^k b^{k+\ell_1} a^{k+\ell_2}$, где $\ell_1 > 0$, так что $w_2 \overline{\in} L_{10}$. Полученное противоречие доказывает, что L_{10} не есть кс-язык.

<u>Пример 2.</u> Рассмотрим язык L_{11} примера I3 из § I2. Допустим, что L_{11} - кс-язык, и положим $k = \max(r, s)$, где r, s - числа из теоремы 3.8. Возьмем цепочку $w = a_1^k a_2^k b a_1^k a_2^k \in L_{11}$ и рассмотрим её представление $w = z_1 y_1 x y_2 z_2$, удовлетворяющее условиям теоремы 3.8. Очевидно, ни y_1, ни y_2 не могут содержать вхождений b—иначе цепочка w_2 содержала бы больше одного вхождения b и поэтому не принадлежала бы L_{11} . В то же время $y_1 x y_2$ не может целиком располагаться с одной стороны от b - иначе w_2 имела бы вид $w' b w''$, где $\ell(w') \ne \ell(w'')$, так что w_2 не принадлежала бы L_{11} . Остается единственная возможность - y_1 располагается целиком слева от b , а y_2 — целиком справа. Отсюда, поскольку длины y_1 и y_2 не превосходят

$s \leq k$, следует, что $y_1 = a_2^{\ell_1}$, $y_2 = a_1^{\ell_2}$. Поэтому $w_2 = a_1^k a_2^{k+\ell_1} b a_1^{k+\ell_2} a_2^k$; а так как хотя бы одно из чисел ℓ_1, ℓ_2 больше нуля, $w_2 \bar{\in} L_{11}$. Итак, L_{11} не есть кс-язык.

Пример 3. Рассмотрим язык L_{13} примера I5 из § I2. Допустим, что L_{13} есть кс-язык, и положим $k = \mathrm{Max}(r, s)$, где r, s – числа из теоремы 3.8. Пусть $w = 1^{k^2}$. В представлении $w = z_1 y_1 x y_2 z_2$, удовлетворяющем условиям теоремы 3.8, будет $y_1 = 1^{\ell_1}$, $y_2 = 1^{\ell_2}$, причем $\ell_1 + \ell_2 > 0$. Имеем теперь $w_2 = 1^{k^2 + \ell_1 + \ell_2}$. Но $k^2 + \ell_1 + \ell_2$ не есть полный квадрат, поскольку $\ell_1 + \ell_2 < \ell(y_1 x y_2) \leq s \leq k$, и, значит, $k^2 < k^2 + \ell_1 + \ell_2 < k^2 + 2k + 1 = (k+1)^2$. Итак, $w_2 \bar{\in} L_{13}$. Следовательно, L_{13} не есть кс-язык.

§ 20. Конечно характеризуемые языки и кс-языки.

В § 7 был введен класс конечно характеризуемых языков, в который, по-видимому, должен входить всякий разумным образом формализованный естественный язык. Сейчас мы установим связь между понятиями конечно характеризуемого языка и кс-языка, доказав следующую теорему:

Теорема 3.9. Если L – конечно характеризуемый язык над словарем V и приведенная конфигурационная характеристика языка L задана, то может быть эффективно построена кс-грамматика, порождающая L.

Доказательство. Сопоставим каждому символу $a \in V$ его "двойника" – новый символ $\bar{a} \bar{\in} V$; множество всех "двойников" обозначим через $\bar{V}$. Введем, кроме того, еще один символ Пр. Для каждой цепочки $w = b_1 \ldots b_k$, где $b_1, \ldots, b_k \in V$, цепочку $\bar{b}_1 \ldots \bar{b}_k$ будем обозначать $\bar{w}$. Через $\bar{L}$ обозначим множество всех цепочек вида $\bar{w}$ таких, что $w \in L$. Очевидно, $Б(\bar{L})$ и $П(\bar{L})$ получаются соответственно из $Б(L)$ и $П(L)$ просто надчеркиванием всех цепочек. Определим теперь системы правил S_1, S_2, S_3 следующим образом: S_1 состоит из всевозможных правил вида Пр.$\to \bar{x}$, где $\bar{x} \in Б(\bar{L})$; S_2 – из всевозможных правил вида $\bar{a} \to \bar{y}$, где $(\bar{y}, \bar{a}) \in П(\bar{L})$; S_3 – из всевозможных правил вида $\bar{a} \to a$, где $a \in V$. Положим

$\Gamma = \langle V, \bar{V} \cup \{\text{Пр.}\}, \text{Пр.}, S_1 \cup S_2 \cup S_3 \rangle$. Покажем, что $L(\Gamma) = L$.

I. Установим, что $L(\Gamma) \subseteq L$. Пусть $z \in L(\Gamma)$. Очевидно, в каждом выводе цепочки z из Пр. на первом шаге, и только на нем, применяется правило из S_1. Ясно также, что найдется такой вывод z из Пр., в котором каждое применение правила из S_2 предшествует каждому применению правила из S_3. Пусть (Пр. $= w_0, w_1, \ldots, w_n = z$) – такой вывод, и пусть первое применение правила из S_3 происходит на К-м шаге. Тогда, очевидно, $w_{k-1} = \bar{w}_n = \bar{z}$. Поэтому достаточно доказать, что $w_{k-1} \in \bar{L}$. Но это верно для всех $w_1, \ldots, w_{k-1}$. В самом деле: (I) w_1 получается из Пр. применением некоторого правила системы S_1, так что $w_1 \in Б(\bar{L}) \subseteq \bar{L}$; (П) если $1 \leq i < k-1$ и $w_i \in \bar{L}$, то и $w_{i+1} \in \bar{L}$, т.к. w_{i+1} получается из w_i подстановкой некоторой конфигурации языка $\bar{L}$ на место её результирующего.

П. Чтобы показать, что $L \subseteq L(\Gamma)$, достаточно – поскольку схема грамматики Γ содержит S_3 – установить, что всякая цепочка $\bar{z} \in \bar{L}$ выводима в Γ из Пр. Мы докажем это индукцией по $\ell(\bar{z})$.

Базис. Если $\ell(\bar{z}) = 1$ и $\bar{z} \in \bar{L}$, то $\bar{z} \in Б(\bar{L})$. Поэтому Пр. $\models \bar{z}\ (\Gamma)$.

Индукционный шаг. Допустим, что всякая цепочка длины $\leq n$, принадлежащая $\bar{L}$, выводима из Пр. в Г. Пусть $\ell(\bar{z}) = n+1$ и $\bar{z} \in \bar{L}$. Если $\bar{z} \in Б(\bar{L})$, рассуждаем, как в базисе. Если же $\bar{z} \bar{\in} Б(\bar{L})$ и r – наименьший ранг конфигураций языка $\bar{L}$, содержащихся в $\bar{z}$, то $\bar{z}$ содержит и простую конфигурацию ранга r языка $\bar{L}$; иначе говоря, существуют такие $\bar{x}_1, \bar{x}_2, \bar{a}, \bar{y}$, что $\bar{z} = \bar{x}_1 \bar{y} \bar{x}_2$ и $(\bar{y}, \bar{a}) \in П(\bar{L})$. Поскольку $\bar{z}$ не содержит конфигураций рангов, меньших r, имеем $\bar{x}_1 \bar{a} \bar{x}_2 \in \bar{L}$. Так как длина цепочки $\bar{x}_1 \bar{a} \bar{x}_2$ меньше $n+1$, эта цепочка по индуктивному предположению выводима из Пр. Но $\bar{z}$ выводима из $\bar{x}_1 \bar{a} \bar{x}_2$ с помощью правила $\bar{a} \to \bar{y} \in S_2$.

Обращение теоремы 3.9 не имеет места. Более того, как показывает следующий пример, даже а-язык может не быть конечно характеризуемым.

Пусть $L_1 = \{bab\} \cup \{a^k \mid k = 1, 2 \dots\}$. Обозначим через L_2 язык, состоящий из всевозможных цепочек, получаемых из цепочек языка L_1 заменой всех или некоторых вхождений a вхождениями цепочек вида cb^mc $(m = 1, 2 \dots)$. Положим $L = L_1 \cup L_2$. Легко проверить, что L порождается а-грамматикой со следующей схемой:

$$
\begin{array}{ll}
\text{Пр.} \to a & \text{Пр.} \to bA \\
\text{Пр.} \to a\,\text{Пр.} & A \to aB_1 \\
\text{Пр.} \to cC & B_1 \to b \\
C \to bB & A \to cC_1 \\
B \to bB & C_1 \to bB_2 \\
B \to c\,\text{Пр.} & B_2 \to bB_2 \\
B \to c & B_2 \to cB_3 \\
 & B_3 \to b
\end{array}
$$

Ясно, что символ a замещаем в L на любую цепочку вида cb^mc, и всякая цепочка, на которую замещаем символ a, имеет такой вид. Кроме того, всякая цепочка вида cb^mc замещаема на a. Что касается символов b и c, то ни один из них не замещаем ни на какую цепочку, отличную от него самого. Например, в цепочке $bcbcb \in L$ замена первого вхождения b на что угодно приводит, очевидно, к цепочке, не принадлежащей L; в той же цепочке первое вхождение c можно заменить только на цепочку вида cb^k, но на такую цепочку нельзя заменить второе вхождение c. Итак, имеем $K(L) = \{(cb^mc, a) \mid m = 1, 2, \dots\}$.

По виду $K(L)$ ясно, во-первых, что все конфигурации - простые, так что $\Pi(L) = K(L)$, и, во-вторых, что $Б(L) = L$. Итак, $Б(L)$ и $\Pi(L)$ бесконечны.

§ 21. Автоматные грамматики и их модификации.

Напомним определение а-грамматики, данное в § II: Грамматика $\Gamma = \langle V, V_1, \text{Пр.}, S\rangle$ называется а-грамматикой, если каждое её правило имеет вид $A \to bB$ или $A \to b$, где $A, B \in V_1$ и $b \in V$.

При изучении языков, порождаемых а-грамматиками /а-языков/ часто бывают полезны видоизмененные формы этого определения, которыми мы сейчас и займемся.

Рассмотрим следующие типы правил:

(i) $A \to xB$

(ii) $A \to aB$

(iii) $A \to x$

(iv) $A \to a$

(v) $A \to B$

(vi) $A \to \Lambda$

Здесь A,B - вспомогательные символы, a-основной символ и x-непустая цепочка над основным словарем.

Покажем, что для всякой грамматики Г, каждое правило кото-[рой] принадлежит к одному из типов $(i) - (vi)$ (или, что то же [с]амое, к одному из типов $(i), (iii), (v), (vi)$), можно эффектив-[н]о построить а-грамматику Γ^*, порождающую язык $L(\Gamma) \setminus \{\Lambda\}$.

Действительно, пусть $\Gamma = \langle V, V_1, \text{Пр.}, S\rangle$ - грамматика, [к]аждое правило которой принадлежит к одному из типов $(i) - (vi)$. [П]оскольку Γ является окс-грамматикой, в силу теоремы 3.4 по Г [м]ожно эффективно построить кс-грамматику Γ', порождающую [я]зык $L(\Gamma) \setminus \{\Lambda\}$, причем, как видно из доказательства тео[р]емы 3.4, каждое правило Γ' либо совпадает с некоторым правилом [Г], либо получается из некоторого правила Г вычеркиванием в пра[во]й части вхождения вспомогательного символа. Поэтому все прави[ла] Г' принадлежат к типам $(i) - (v)$. К грамматике Г' мы [мо]жем применить лемму 3.3. В результате получим кс-грамматику [Г]'', эквивалентную Г' и не имеющую правил типа (v). При [эт]ом из доказательства леммы 3.3 видно, что каждое правило Г'' [ли]бо совпадает с некоторым правилом Γ', либо получается из не[к]оторого правила Γ' заменой вхождений вспомогательных символов [вх]ождениями других символов /вспомогательных или основных/; по-[это]му все правила Γ'' принадлежат к типам $(i) - (iv)$. Оста[ет]ся устранить правила типов (i) и (iii), не принадлежащие [с]оответственно к типам (ii) и (iv). Это легко сделать, заме[ни]в каждое правило вида $A \to b_1 \dots b_k B$ /где $b_1, \dots, b_k \in V, k > 1)$ системой правил $A \to b_1F_1$, $F_1 \to b_2F_2, \dots,$ $[F]_{k-2} \to b_{k-1}F_{k-1}, F_{k-1} \to b_kB$, а каждое правило вида $A \to b_1 \dots b_k$ [/г]де $b_1, \dots, b_k \in V, k > 1$/ системой правил $A \to b_1F_1$, $F_1 \to b_2F_2, \dots,$ $[F]_{k-2} \to b_{k-1}F_{k-1}$, $F_{k-1} \to b_k$ / в обоих случаях $F_1, \dots, F_{k-1}$ –

новые вспомогательные символы; необходимо, конечно, чтобы все новые символы, входящие в разные системы, были попарно различны/. Таким способом мы получим грамматику, эквивалентную Γ'' и имеющую только правила типов (ii) и (iv), т.е. являющуюся а-грамматикой.

Итак, если дана грамматика, каждое правило которой принадлежит к одному из типов $(i)-(vi)$, то порождаемый ею язык - за исключением пустой цепочки, если она в нем содержится - можно породить также, ограничиваясь правилами типов (ii) и (iv). Но можно и по-другому ограничить типы правил. Именно, мы сейчас покажем, что для любой грамматики, каждое правило которой принадлежит к одному из типов $(i)-(vi)$, можно построить эквивалентную ей грамматику, каждое из правил которой принадлежит к типу (ii) или (vi).

Это легко сделать следующим образом. I) В силу примечания (2) к лемме 3.3 правила типа (v) могут быть устранены так же, как в предыдущем случае. 2) После того, как правила типа (v) устранены, правила типов (i) и (iii), не принадлежащие соответственно к типам (ii) и (iv) устраняются точно так же, как в предыдущем случае. 3) Наконец, каждое правило типа (iv) $A\to b$ заменяется двумя новыми правилами $A\to bK$ и $K\to\Lambda$, где K - новый символ /можно считать, что этот символ - один и тот же для всех правил типа (iv)/. Эквивалентность полученной грамматики и первоначальной очевидна.

Грамматику, каждое правило которой принадлежит к одному из типов (ii) или (vi), мы будем называть <u>модифицированной а-грамматикой</u> /сокращенно - <u>ма-грамматикой</u>/.

Легко видеть, что язык, порождаемый ма-грамматикой, тогда и только тогда не содержит пустой цепочки, когда в этой грамматике нет правила $Пр.\to\Lambda$.

А-грамматики и ма-грамматики допускают простое графическое изображение. Пусть сначала $\Gamma=\langle V,V_1,Пр.,S\rangle$ - ма-грамматика. Сопоставим каждому символу $A\in V_1$ точку /которую обычно будем просто отождествлять с А/. Каждому правилу $A\to bB\in S$ сопоставим дугу /стрелку/, исходящую из А и входящую в В; эту дугу пометим символом b. В полученном таким образом графе /точнее, мультиграфе/ отметим вершину, соответствующую Пр. /её мы

будем называть <u>начальной</u>/, и вершины, соответствующие символам К, для которых имеются правила $K\to\Lambda$ /эти вершины мы назовем <u>заключительными</u>/. Построенный таким образом объект-мультиграф с помеченными ребрами и выделенными начальной и заключительными вершинами*) мы назовем <u>диаграммой</u> грамматики Г.

Чтобы получить диаграмму а-грамматики, мы будем предварительно превращать её в ма-грамматику следующим стандартным способом: к вспомогательному словарю добавляется один новый символ К, к схеме - правило $K\to\Lambda$, и каждое правило вида $A\to b$ заменяется правилом вида $A\to bK$. Диаграмму полученной так ма-грамматики мы и будем, по определению, считать диаграммой исходной а-грамматики. Заметим, что диаграмма а-грамматики всегда содержит точно одну заключительную вершину, которая не совпадает с начальной и из которой не исходят никакие дуги.

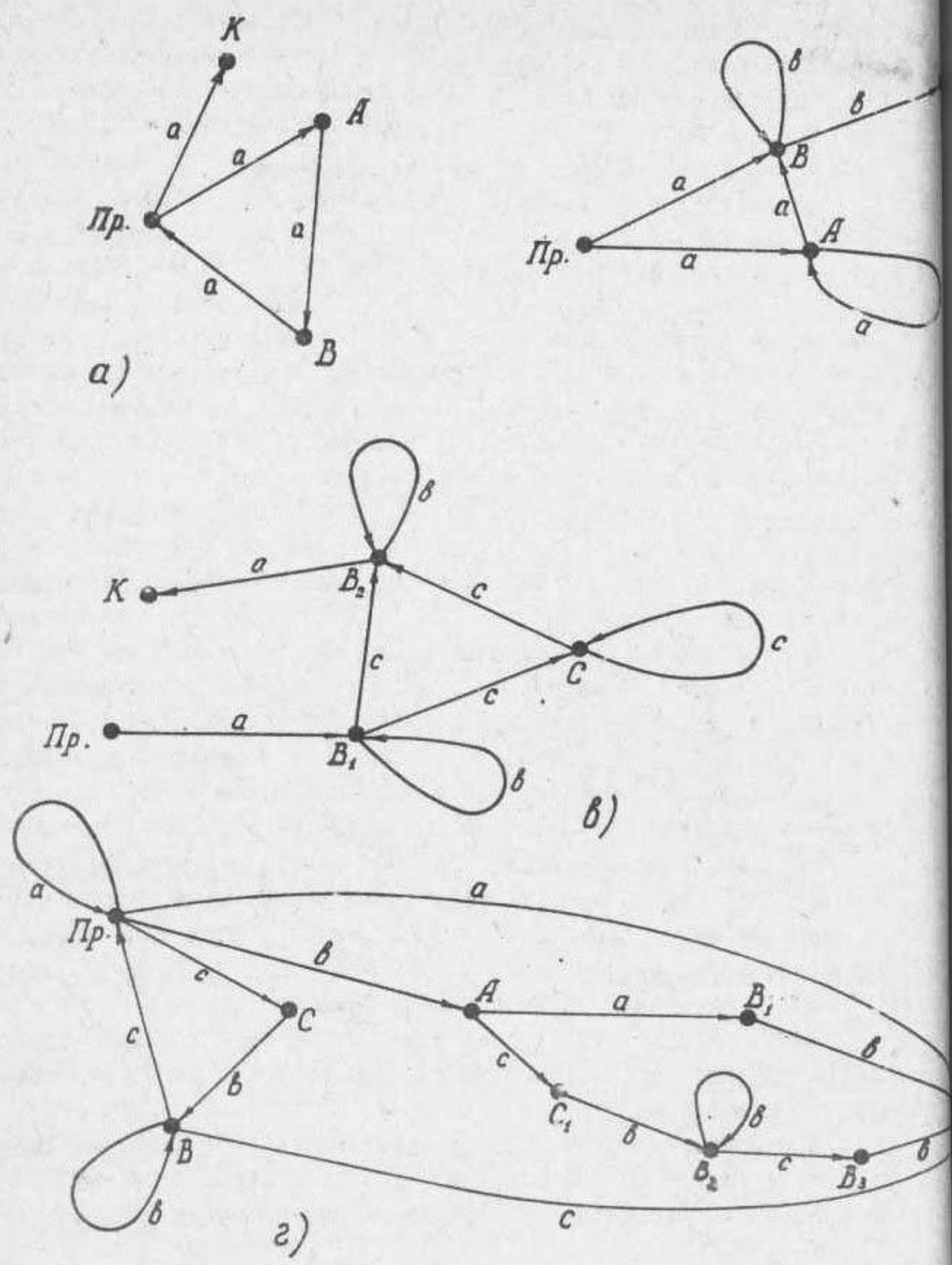
<u>Пример</u>. На черт. I4 а), б), в), г) показаны соответственно диаграммы грамматик Γ_1, Γ_2, Γ_3 примеров 3,4,5 из § I2 и грамматики примера из § 20. Начальная вершина всюду обозначена Пр., заключительная - К.

Если в диаграмме а- или ма-грамматики $\Gamma=\langle V,V_1,Пр.,S\rangle$ некоторое ребро, исходящее из A и входящее в В, помечено символом b, мы будем говорить просто, что символ b исходит из А и входит в В. Если $X=A_0A_1\ldots A_n$ - последовательность вершин диаграммы, $x=a_1\ldots a_n$ - цепочка над V и при этом для каждого $i=1,\ldots,n$ a_i исходит из A_{i-1} и входит в A_i, мы будем говорить, что X есть <u>путь, производящий</u> цепочку x. Будем считать также, что всякий путь, состоящий из одной вершины, производит пустую цепочку. Путь $A_0\ldots A_n$ будем называть <u>циклом</u>, если $A_0=A_n$, и <u>полным путем</u>, если A_0 - начальная вершина, а A_n - заключительная.

Легко видеть, что цепочка над V тогда и только тогда принадлежит $L(\Gamma)$, когда она производится некоторым полным путем диаграммы Γ.

С помощью диаграмм для а-грамматик особенно наглядно представляются понятия, введенные в § I8. Так, первое условие существенности для символа А означает существование пути, ведуще-

*) Вообще говоря, начальная вершина может одновременно быть заключительной.



Черт. I4.

го из начальной вершины в А, второе - существование пути, ведущего из А в заключительную вершину. Символ А является существенным тогда и только тогда, когда через А проходит полный путь. Приведенная а-грамматика характеризуется тем, что через каждую вершину диаграммы проходит полный путь. Циклический символ - это вершина, принадлежащая некоторому циклу; циклическая а-грамматика - такая а-грамматика, диаграмма которой содержит хоть один цикл.

Из последнего замечания сразу следует, что приведенная а-грамматика порождает бесконечный язык тогда и только тогда, когда она циклическая; это, впрочем, вытекает и из теоремы 3.5. Алгоритм распознавания конечности становится для а-грамматик весьма простым, поскольку по диаграмме сразу видно, какие вершины нужно отбросить, чтобы получить приведенную грамматику, и имеются ли в полученной после отбрасывания диаграмме циклы. Еще проще оказывается алгоритм распознавания пустоты - а-грамматика порождает пустой язык тогда и только тогда, когда её диаграмма не содержит полных путей.

§ 22. Теорема о конгруентностях конечного индекса.

В этом параграфе мы докажем теорему, играющую в теории а-грамматик примерно такую же роль, какую теорема 3.8 играет в теории кс-грамматик.

Предварительно введем некоторые новые понятия.

Индексом отношения эквивалентности R, заданного на множестве M, мы будем называть мощность множества классов, на которые это отношение разбивает М.

Пусть R - отношение эквивалентности на свободной полугруппе $F(V)$ и L - язык над словарем V. Мы будем говорить, что R согласовано с языком L, если для любых цепочек x, y над V из xRy и $x \in L$ следует $y \in L$ — иначе говоря, если каждый R - класс либо содержится в L, либо не пересекается с L /еще иначе: L есть объединение некоторых R - классов/.

Нас будут интересовать конгруентности на $F(V)$, согласованные с L. Примером такой конгруентности может служить отношение взаимозамещаемости в языке L /см. замечания к определению этого отношения в § 6/. Оказывается, что взаимозамещаемость является в некотором смысле максимальной конгруентностью, согласованной с L. Именно, справедливо следующее утверждение:

Лемма 3.9. Если К- произвольная конгруентность, согласованная с языком L над словарем V, то из xKy следует $x \Longleftrightarrow y (L)$; иначе говоря, каждый K - класс содержится в некотором классе взаимозамещаемости.

Доказательство. Пусть xKy и $z_1 x z_2 \in L$. Поскольку К-конгруентность, имеем $z_1 x z_2 K z_1 y z_2$, а отсюда, ввиду согласованности K с языком L, вытекает $z_1 y z_2 \in$

Теперь мы можем сформулировать теорему:

Теорема 3.10. Для произвольного языка L следующие три условия равносильны:

I) L порождается ма-грамматикой; 2) существует отношение конгруентности конечного индекса, согласованное с L; 3) отношение взаимозамещаемости в языке L имеет конечный индекс.

Доказательство. Достаточно доказать следующие три импликации: I) $\to$ 2), 2) $\to$ 3), 3) $\to$ I).

I) $\to$ 2). Пусть $\Gamma = \langle V, V_1, Пр, S \rangle$ - ма-грамматика, порождающая L. Будем говорить, что пара вспомогательных символов (A,B) обрамляет цепочку x над V, если диаграмма грамматики Г содержит путь, начинающийся символом А, оканчивающийся символом В и производящий x. Множество всех пар вспомогательных символов, обрамляющих x, обозначим $O(x)$. Определим теперь отношение K_0 на $F(V)$ следующим образом: $x K_0 y \underset{Def}{\equiv} O(x) = O(y)$. Непосредственно очевидно, что K_0- отношение эквивалентности. Ясно также, что K_0 имеет конечный индекс. Действительно, мощность множества K_0 - классов во всяком случае не больше мощности множества всех множеств, элементами которых являются упорядоченные пары вспомогательных символов; но таких множеств имеется лишь конечное число.

Легко видеть, что K_0 - конгруентность. В самом деле, пусть $x_1 K_0 y_1$ и $x_2 K_0 y_2$. Это означает, что $O(x_1) = O(y_1) = O_1$ и $O(x_2) = O(y_2) = O_2$. Пара (A, B) обрамляет цепочку $x_1 x_2$ тогда и только тогда, когда существует вспомогательный символ C такой, что пара (A,C) обрамляет x_1, а пара (C,B) обрамляет x_2. Таким образом,

$$O(x_1, x_2) = \{(AB) / \exists C [(A,C) \in O_1 \,\&\, (C,B) \in O_2]\}.$$

Но $O(y_1, y_2)$ определяется точно так же, так что $O(x_1, x_2) = O(y_1, y_2)$, т.е. $x_1 x_2 K_0 y_1 y_2$.

Наконец, согласованность отношения K_0 с языком L следует из того очевидного факта, что цепочка x тогда и только тогда принадлежит L, когда $O(x)$ содержит пару $(Пр, K)$, где K – какая-либо заключительная вершина диаграммы.

2) $\to$ 3). Непосредственно следует из леммы 3.9.

3) $\to$ I). Пусть отношение взаимозамещаемости в языке L над V имеет конечный индекс. Заметим прежде всего, что-поскольку взаимозамещаемость есть конгруентность - если к цепочкам, принадлежащим некоторому классу взаимозамещаемости А, приписывать справа один и тот же символ $b \in V$, то все получаемые при этом цепочки будут принадлежать одному классу взаимозамещаемости В. Мы будем говорить в этом случае, что символ b переводит А в В.

Определим теперь ма-грамматику $\Gamma = \langle V, V_1, Пр, S \rangle$ следующим образом: I) V_1 есть множество классов взаимозамещаемости; 2) Пр. - тот класс, который содержит пустую цепочку; 3) S состоит из всевозможных правил вида $A \to bB$, где b переводит А в В, и всевозможных правил вида $A \to \Lambda$, где $A \subseteq L$.

Докажем, что цепочка x над V принадлежит классу взаимозамещаемости А тогда и только тогда, когда диаграмма грамматики Γ содержит путь, ведущий из Пр. в A и производящий x. Доказательство проведем индукцией по длине x. Если $l(x) = 0$, т.е. $x = \Lambda$, справедливость утверждения очевидна. Пусть оно доказано для цепочек длины $n-1$, и пусть $x = b_1 \ldots b_{n-1} b_n$, где $b_1, \ldots, b_{n-1}, b_n \in V$. Обозначим через D класс взаимозамещаемости, содержащий x, и через C - класс взаимозамещаемости, содержащий цепочку $x' = b_1 \ldots b_{n-1}$. По индуктивному предположению x' производится путем y, ведущим из Пр. в С. Но b_n переводит С в Д; поэтому S содержит правило $C \to b_n D$, так что путь yD, ведущий из Пр. в Д, производит x. Обратно, пусть некоторый путь $B_0 B_1 \ldots B_{n-1} B_n$ такой, что $B_0 = Пр.$, производит цепочку x. Тогда путь $B_0 \ldots B_{n-1}$ производит цепочку x' и по индуктивному предположению $x' \in B_{n-1}$. Отсюда, поскольку b_n переводит B_{n-1} в B_n, следует, что $x \in B_n$.

Теперь легко видеть, что $L = L(\Gamma)$. Действительно, L есть объединение тех классов взаимозамещаемости, которые служат заключительными вершинами диаграммы Γ; но это объединение совпадает, в силу только что доказанного утверждения, с множеством цепочек, производимых полными путями, т.е. с $L(\Gamma)$.

Замечание. Из доказательство импликации 3) → I) ясно, что вместо взаимозамещаемости в этом пункте можно использовать произвольную конгруентность, согласованную с L. Мы можем, таким образом, утверждать следующее: Пусть на $F(V)$ задана произвольная конгруентность K конечного индекса, причем $A_0, A_1, \dots, A_s$ - все K-классы и класс A_0 содержит пустую цепочку. Пусть T - произвольное множество, являющееся объединением нескольких K-классов. Если S состоит из всевозможных правил вида $A_i \to b A_j$, где b- элемент V, переводящий A_i в A_j, и всевозможных правил вида $A_i \to \Lambda$, где $A_i \subseteq T$, то ма-грамматика $\langle V, \{A_0, \dots, A_s\}, A_0, S \rangle$ порождает T.

Приведем примеры применения теоремы 3.I0.

Пример I. Рассмотрим язык L_4 примера 6 из § I2. Допустим, что L_4 - а-язык. Тогда должна существовать конгруентность K конечного индекса, согласованная с L_4. В силу конечности индекса найдутся такие r и s, что $r \neq s$ и $a^r K a^s$. Но тогда будет $a^r b^r K a^s b^r$, что невозможно, т.к. $a^r b^r \in L_4$ и $a^s b^r \bar{\in} L_4$. Итак, L_4 - не а-язык.

Пример 2. Рассмотрим язык L_6 примера 8 из § I2. Допустим, что существует конгруентность K конечного индекса, согласованная с L_6. Тогда найдутся такие r и s, $r \neq s$, что

$$\underbrace{))\dots\dots)}_{r\ \text{раз}}\ K\ \underbrace{))\dots)}_{s\ \text{раз}}$$

Поэтому выражение $\neg(\neg(\dots\neg(p_1)\dots)$, содержащее по r вхождений знака отрицания, левой скобки и правой скобки, будет находиться в отношении K с аналогичным выражением, содержащим по r вхождений знака отрицания и левой скобки, но s вхождений правой скобки. Однако это невозможно, т.к. первое выражение принадлежит

L_6, а второе не принадлежит. Итак, L_6 - не а-язык.

Совершенно аналогично можно было бы показать, что языки L_5, L'_5, L_7 и $L_9 - L_{13}$ из § I2 - не а-языки. /Для языков L_{10}, L_{11} и L_{13} это следует также из того, что они, как показано в § I9, не являются контекстно-свободными/.

§ 23. Операции над языками.

В настоящем параграфе мы займемся вопросом о замкнутости различных классов языков относительно теоретико-множественных и некоторых других операций. Именно, мы будем рассматривать следующие пять операций над языками:

I. Теоретико-множественное объединение языков L_1 и L_2 /обозначение $L_1 \cup L_2$/.

2. Теоретико-множественное пересечение L_1 и L_2 /обозначение $L_1 \cap L_2$/.

3. Взятие дополнения языка L над словарем V относительно множества $F'(V)$ всех непустых цепочек над V /обозначение CL/.

4. Прямое умножение языка L_1 на язык L_2 /обозначение $L_1 \times L_2$/: $L_1 \times L_2 \overset{Def}{=} \{xy \mid x \in L_1 \,\&\, y \in L_2\}$.

5. Итерация языка L /обозначение L^*/: $L^{*)} \overset{Def}{=} \bigcup_{n=1}^{\infty} L^n$ /здесь L^n означает $(\underbrace{L \times L \times \dots \times L \times L}_{n\ \text{раз}})$*).

Например, если $L_1 = \{a^n \mid n = 1, 2, \dots\}$, $L_2 = \{b^m \mid m = 1, \dots\}$, то $L_1 \times L_2 = \{a^n b^m \mid n, m = 1, 2, \dots\}$; если $L = \{a, b\}$, то $L^* = F'(\{a, b\})$.

Замкнутость некоторого класса языков K относительно той или иной операции будет пониматься в обычном смысле: результат применения операции к языку /языкам/ класса K есть снова язык класса K.

Кроме этого, мы будем пользоваться понятием эффективной замк-

*) Обычно итерация определяется как $\bigcup_{n=0}^{\infty} L^n$, где L^0 означает $\{\Lambda\}$. Нам, однако, удобнее исключить из рассмотрения пустую цепочку, поскольку мы в основном будем иметь дело с нс-языками, которые пустой цепочки содержать не могут.

нутости. Именно, пусть М-некоторый класс грамматик и $L_V(M)$- класс всех языков над словарем V , порождаемых грамматиками класса M . Мы будем говорить, что класс $L_V(M)$ эффективно замкнут относительно некоторой операции O , если существует алгоритм, позволяющий по любой грамматике /любым грамматикам/ класса M построить грамматику того же класса, порождающую язык, являющийся результатом операции O над языком /языками/, порождаемым /-ыми/ данной /-ыми/ грамматикой /-ами/.

В дальнейшем мы будем обозначать класс всех грамматик через Γp , класс всех нс-грамматик - через НС, класс всех кс-грамматик - через КС и класс всех а-грамматик - через А.

Теорема 3.II. а) Для любого V каждый из классов $L_V(\Gamma p), L_V(HC), L_V(KC), L_V(A)$ эффективно замкнут относительно операций $\cup$, $\times$, $*$. б) Для любого V каждый из классов $L_V(\Gamma p), L_V(HC), L_V(A)$ эффективно замкнут относительно операции $\cap$; если V содержит не менее двух символов, то класс $L_V(KC)$ не замкнут относительно $\cap$. в) Для любого V класс $L_V(A)$ эффективно замкнут относительно операции C ; если V содержит не менее двух символов, то класс $L_V(KC)$ не замкнут относительно С.

Доказательство. а) Рассмотрим сначала операцию $\cup$. Пусть $\Gamma_1 = \langle V, V_{11}, \text{Пр}_1, S_1 \rangle$ и $\Gamma_2 = \langle V, V_{12}, \text{Пр}_2, S_2 \rangle$. Мы можем считать, не нарушая общности, что $V_{11} \cap V_{12} = \varnothing$. Введем новый символ $\text{Пр} \notin V \cup V_{11} \cup V_{12}$ и положим $\Gamma = \langle V, V_{11} \cup \{\text{Пр}\}, \text{Пр}, S_1 \cup S_2 \cup \{\text{Пр} \to \text{Пр}_1, \text{Пр} \to \text{Пр}_2\} \rangle$. Ясно, что $L(\Gamma) = L(\Gamma_1) \cup L(\Gamma_2)$. Если при этом каждая из грамматик Γ_1 и Γ_2 является нс-, соответственно кс-грамматикой, то и Г является таковой. Если Γ_1 и Γ_2 - а-грамматики, то все правила Г принадлежат к типам (ii) , (iv) и (v) (§ 2I), так что по Г можно эффективно построить а-грамматику, порождающую язык $L(\Gamma) \setminus \{\Lambda\} = L(\Gamma)$.

Перейдем к операции $\times$. Пусть Γ_1 , Γ_2 и Пр определяются также и удовлетворяют тем же условиям, что и в предыдущем случае. Положим $\Gamma = \langle V, V_{11} \cup V_{12} \cup \{\text{Пр}\}, \text{Пр}, S_1 \cup S_2 \cup \{\text{Пр} \to \text{Пр}_1 \text{Пр}_2\} \rangle$. Ясно, что $L(\Gamma) = L(\Gamma_1) \times L(\Gamma_2)$, и если Γ_1 и Γ_2 - нс- или кс-грамматики, то и Г - нс-, соответственно кс-грамматика.

Если Γ_1 и Γ_2 - а-грамматики, то язык $L(\Gamma_1) \times L(\Gamma_2)$

порождается, как легко проверить, а-грамматикой $\langle V, V_{11} \cup V_{12}, \text{Пр}_1, S_1' \cup S_2 \rangle$, где S_1' получается из S_1 заменой каждого правила вида $A \to b$ правилом $A \to b\,\text{Пр}_2$.

Рассмотрим, наконец, операцию* . Пусть $\Gamma = \langle V, V_1, \text{Пр}, S \rangle$. Положим $\Gamma^* = \langle V, V_1, \text{Пр}, S \cup \{\text{Пр} \to \text{Пр}\,\text{Пр}\} \rangle$. Ясно, что Γ^* порождает язык $(L(\Gamma))^*$, и если Γ - нс- или кс-грамматика, то и Γ^* - нс, соответственно кс-грамматика.

Если Γ - а-грамматика, то $(L(\Gamma))^*$ порождается, как легко проверить, а-грамматикой $\langle V, V_1, \text{Пр}, S^* \rangle$, где S^* получается из S добавлением для каждого правила вида $A \to b$ правила $A \to b\,\text{Пр}$.

б) Пусть $\Gamma_1 = \langle V, V_{11}, \text{Пр}_1, S_1 \rangle$; $\Gamma_2 = \langle V, V_{12}, \text{Пр}_2, S_2 \rangle$ причем $V_{11} \cap V_{12} = \varnothing$. Сопоставим каждому символу $a \in V$ два различных "двойника" $\overline{a}$ и $\overline{\overline{a}}$. Множество всех символов $\overline{a}$ обозначим $\overline{V}$, множество всех символов $\overline{\overline{a}}$ - $\overline{\overline{V}}$. Через $\overline{S}_1$, соответственно $\overline{\overline{S}}_2$,обозначим систему правил, полученную из S_1 (из S_2) заменой в каждом правиле каждого вхождения каждого символа $a \in V$ вхождением символа $\overline{a}$ /соответственно $\overline{\overline{a}}$ /. Положим $\overline{\Gamma}_1 = \langle \overline{V}, V_{11}, \text{Пр}_1, \overline{S}_1 \rangle$; $\overline{\Gamma}_2 = \langle \overline{\overline{V}}, V_{12}, \text{Пр}_2, \overline{\overline{S}}_2 \rangle$.

Введем теперь новый символ Пр., не входящий ни в один из словарей $V, \overline{V}, \overline{\overline{V}}, V_{11}, V_{12}$. Обозначим через S' следующую систему правил:

$$\text{Пр} \to \text{Пр}_1 \text{Пр}_2$$
$$\overline{a}\,\overline{\overline{b}} \to \overline{\overline{b}}\,\overline{a}, \quad a, b \in V$$
$$\overline{\overline{a}}\,\overline{a} \to a, \quad a \in V$$

Положим $\Gamma = \langle V, V_1, \text{Пр}, S \rangle$, где

$$V_1 = V_{11} \cup V_{12} \cup \overline{V} \cup \overline{\overline{V}} \cup \{\text{Пр}\}; \quad S = \overline{S}_1 \cup \overline{\overline{S}}_2 \cup S'.$$

Нетрудно показать, что $L(\Gamma) = L(\Gamma_1) \cap L(\Gamma_2)$. Действительно, всякая цепочка $x \in L(\Gamma_1) \cap L(\Gamma_2)$, $x = a_1 \ldots a_k$ $(a_1, \ldots, a_k \in V)$, может быть выведена в Г следующим образом: сначала выводится $\text{Пр}_1 \text{Пр}_2$, затем-по правилам систем $\overline{S}_1$ и $\overline{\overline{S}}_2$ - цепочка $\overline{a}_1 \ldots \overline{a}_k \overline{\overline{a}}_1 \ldots \overline{\overline{a}}_k$, потом $\overline{\overline{a}}_1 \overline{a}_1 \ldots \overline{\overline{a}}_k \overline{a}_k$ и, наконец, $a_1 \ldots a_k$. С другой стороны, ясно, что цепочка $a_1 \ldots a_k$ над V может быть выведена в Г из Пр. только тогда, когда в $\overline{\Gamma}_1$ выводима из Пр. цепочка $\overline{a}_1 \ldots \overline{a}_k$, а в $\overline{\Gamma}_2$ выводима из Пр_2 цепочка $\overline{\overline{a}}_1 \ldots \overline{\overline{a}}_k$ -иначе говоря, когда цепочка $a_1 \ldots a_k$ выводима в Γ_1 из Πp_1 и в Γ_2 из Πp_2. Таким образом, $L(\Gamma) = L(\Gamma_1) \cap L(\Gamma_2)$.

Если Γ_1 и Γ_2 – нс-грамматики, то Γ – грамматика без существенного укорачивания, т.к. в ее схеме укорачивающими являются только правила вида $\overline{\overline{a}}\,\overline{a} \to a$, а они, очевидно, заключительные. В силу теоремы 3.3 по Г можно эффективно построить эквивалентную ей нс-грамматику.

Итак, классы $L_V(\Gamma_p)$ и $L_V(HC)$ эффективно замкнуты относительно $\cap$.

Незамкнутость класса $L_V(KC)$ относительно $\cap$ следует из того, что языки L_5 и L'_5 примера 7 из § I2 являются кс-языками, в то время как их пересечение совпадает с языком L_{10} примера I2, который, как показано в § I9, не порождается никакой кс-грамматикой.

Наконец, эффективная замкнутость класса $L_V(A)$, в силу теоретико-множественного тождества $L_1 \cap L_2 = C(CL_1 \cup CL_2)$, следует из доказанной выше эффективной замкнутости класса $L_V(A)$ относительно $\cup$ и эффективной замкнутости того же класса относительно C, которая будет доказана в следующем пункте.

в) Если бы класс $L_V(KC)$ был замкнут относительно C, он был бы – в силу пункта а) и тождества $L_1 \cap L_2 = C(CL_1 \cup CL_2)$ – замкнут также и относительно $\cap$ *).

Остается доказать эффективную замкнутость класса $L_V(A)$ относительно C. Пусть $\Gamma = \langle V, V_1, \Pi p, S \rangle$ – а-грамматика. Определим конгруентность K_0 на $F(V)$, как в доказательстве теоремы 3.I0. Обозначим через V'_1 множество всех K_0-классов, через $\Pi p'$ – K_0-класс, содержащий пустую цепочку, и через S' – систему, состоящую из всевозможных правил двух видов: вида $A \to bB$, где A, B – K_0-классы и b-символ из V, переводящий A в B, и вида $A \to \Lambda$, где A – K_0-класс, содержащийся в $F(V) \setminus L(\Gamma)$. Положим $\Gamma' = \langle V, V'_1, \Pi p', S' \rangle$. Γ' есть ма-грамматика, и в силу замечания к теореме 3.I0 $L(\Gamma') = F(V) \setminus L(\Gamma)$; Γ' строится по Γ эффективно, т.к. K_0 – классы, в силу определения

*) Нетрудно было бы доказать незамкнутость класса $L_V(KC)$ относительно $\cap$ и непосредственно, например, построив кс-грамматику, порождающую CL_{10} / L_{10} – язык примера I2 из § I2/.

отношения K_0, легко найти эффективно, отношение "символ b переводит K_0 – класс А в K_0 – класс B" эффективно проверяемо, и по любому K_0 – классу легко распознать, содержится ли он в $F(V) \setminus L(\Gamma)$ и содержит ли он пустую цепочку. /Собственно, здесь удобнее было бы говорить не о самих K_0 -классах, а о взаимно-однозначно соответствующих им конечных множествах пар вспомогательных символов/.

Итак, по Γ можно эффективно построить ма-грамматику Γ', порождающую язык $F(V) \setminus L(\Gamma)$. Но по Γ' можно эффективно построить а-грамматику, порождающую язык $L(\Gamma') \setminus \{\Lambda\} = F'(V) \setminus L(\Gamma) = CL(\Gamma)$.

Доказательство закончено.

Замечания. I) В формулировке теоремы 3.II ничего не сказано о поведении классов $L_V(\Gamma_p)$ и $L_V(HC)$ по отношению к операции С. Класс $L_V(\Gamma_p)$ относительно C не замкнут; установить это мы сможем только в гл. V, т.к. доказательство требует привлечения понятий теории алгоритмов. Что же касается вопроса о замкнутости класса $L_V(HC)$ относительно C, то это еще не решенная задача.

2) Нетрудно было бы показать /мы не будем этого делать/, что если V состоит из одного элемента, то $L_V(KC) = L_V(A)$, так что в этом случае класс $L_V(KC)$ эффективно замкнут относительно всех пяти операций.

Полученные результаты можно свести в следующую таблицу (плюс означает эффективную замкнутость, минус – незамкнутость; имеется в виду словарь V, содержащий не менее двух элементов):

	$L_V(\Gamma_p)$	$L_V(HC)$	$L_V(KC)$	$L_V(A)$
$\cup$	+	+	+	+
$\cap$	+	+	–	+
C	–		–	+
$\times$	+	+	+	+
$*$	+	+	+	+

Докажем еще следующую теорему.

Теорема 3.12. Пересечение кс-языка и а-языка есть кс-язык. При этом по заданным кс-грамматике $\Gamma_1 = \langle V, V_{11}, \Pi p_1, S_1 \rangle$ и а-грамматике $\Gamma_2 = \langle V, V_{12}, \Pi p_2, S_2 \rangle$ можно эффективно построить кс-грамматику, порождающую язык $L(\Gamma_1) \cap L(\Gamma_2)$.

Доказательство. Мы будем считать, что Γ_1 удовлетворяет условиям лемма 3.4. Каждому символу $A \in V_{11}$ и каждой упорядоченной паре P, Q вершин диаграммы Γ_2 сопоставим новый символ $T(A, P, Q)$. Множество всех таких символов обозначим через V_1. Положим $\Pi p = T(\Pi p_1, \Pi p_2, K)$, где K - заключительная вершина диаграммы Γ_2. Каждому правилу $A \to BC \in S_1$ и каждой упорядоченной тройке P, Q, R вершин диаграммы Γ_2 сопоставим новое правило:

$$T(A, P, R) \to T(B, P, Q)\, T(C, Q, R)$$

Далее, если S_1 содержит правило $A \to a$ и S_2 содержит правило $P \to a\,Q$ $(Q \neq K)$, определим новое правило $T(A, P, Q) \to a$; если S_1 содержит правило $A \to a$ и S_2 содержит правило $P \to a$, определим новое правило $T(A, P, K) \to a$.

Множество всех новых правил обозначим через S.

Положим теперь $\Gamma = \langle V, V_1, \Pi p, S \rangle$ и покажем, что $L(\Gamma) = L(\Gamma_1) \cap L(\Gamma_2)$.

I. Пусть $x = a_1 \dots a_k \in L(\Gamma)$; $(a_1, \dots, a_k \in V)$. Тогда существуют такие символы $T_1, \dots, T_k \in V_1$, что $\Pi p \vdash T_1 \dots T_k (\Gamma)$ и для каждого $i = 1, \dots, k$ $T_i \models a_i (\Gamma)$. Но цепочка $T_1 \dots T_k$ над V_1 может быть выведена в Г из Пр. только в том случае, когда она имеет вид

$$T(A_1, P_0, P_1)\, T(A_2, P_1, P_2) \dots T(A_k, P_{k-1}, P_k),$$

где $P_0 = \Pi p_2$, $P_k = K$ и цепочка $A_1 \dots A_k$ выводима в Γ_1 из Πp_1. В то же время соотношение $T_i = T(A_i, P_{i-1}, P_i) \models a_i (\Gamma)$ означает, что: а) $A_i \models a_i (\Gamma_1)$ и при этом б) либо $P_{i-1} \models a_i P_i (\Gamma_2)$ /если $i < k$/, либо $P_{i-1} \models a_i (\Gamma_2)$ /если $i = k$/. В силу а) цепочка $x = a_1 \dots a_k$ выводима в Γ_1 из $A_1 \dots A_k$, а значит, и из Пр.$_1$; в силу б) цепочка x производится путем $P_0 \dots P_k$ диаграммы Γ_2, а поскольку этот путь - полный, то $x \in L(\Gamma_2)$. Итак, $x \in L(\Gamma_1) \cap L(\Gamma_2)$.

II. Пусть $x = a_1 \dots a_k \in L(\Gamma_1) \cap L(\Gamma_2)$. $(a_1, \dots, a_k \in V)$.

Тогда, с одной стороны, существуют такие $A_1, \dots, A_k \in V_{11}$, что $\Pi p \vdash A_1 \dots A_k (\Gamma_1)$ и для каждого $i = 1, \dots, k$ $A_i \models a_i (\Gamma_1)$; с другой стороны, существует полный путь $P_0 \dots P_k$ диаграммы Γ_2, производящий x, т.е. для каждого $i = 1, 2, \dots, k$ S_2 содержит правило $P_{i-1} \to a_i P_i$, если $i < k$, и $P_{i-1} \to a_i$, если $i = k$. Но соотношение $\Pi p_1 \vdash A_1 \dots A_k (\Gamma_1)$ влечет соотношение $\Pi p \vdash T(A_1, P_0, P_1)\, T(A_2, P_1, P_2) \dots T(A_k, P_{k-1}, P_k) (\Gamma)$, а из $A_i \models a_i (\Gamma_1)$ и $P_{i-1} \models a_i P_i (\Gamma_2)$ /или, при $i = k$ $P_{i-1} \models a_i (\Gamma_2)$/ следует, что $T(A_i, P_{i-1}, P_i) \models a_i (\Gamma)$. Итак, $\Pi p \vdash a_1 \dots a_k (\Gamma)$.

Глава IУ.

КАТЕГОРИАЛЬНЫЕ ГРАММАТИКИ.

§ 24. Основные определения.

Рассмотрим конечное множество W, которое мы будем называть словарем категорий. Элементы W будем называть элементарными категориями. Введем в рассмотрение четыре символа $\backslash, /, [,]$, не принадлежащие W. Из этих символов и элементов W будем строить специального вида выражения, которые будем называть категориями над W /или просто категориями/. Понятие категории определяется следующим образом:

I. Всякая элементарная категория есть категория.

П. Если Φ, Ψ - категории, то выражения $[\Phi/\Psi]$ /читается "Φ над Ψ"/ и $[\Phi\backslash\Psi]$ /читается "Φ под Ψ"/ - также категории.

Ш. Всякая категория является таковой либо в силу п.I, либо в силу п.П.

Множество всех категорий над W будем называть категориальной системой над W и обозначать $K(W)$. Множество всех цепочек, составленных из категорий над W (т.е. свободная полугруппа с системой образующих $K(W)$) будет обозначаться $F(K(W))$.

Пусть $\alpha, \beta \in F(K(W))$. Мы будем говорить, что α непосредственно сокращается до β, если имеет место одно из двух: либо $\alpha = \gamma\Phi[\Phi\backslash\Psi]\delta$, $\beta = \gamma\Psi\delta$, либо $\alpha = \gamma[\Phi/\Psi]\Psi\delta$, $\beta = \gamma\Phi\delta$ (здесь $\Phi, \Psi \in K(W)$, $\gamma, \delta \in F(K(W))$).

Замечания. I) Можно рассматривать категорию $[\Phi\backslash\Psi]$ как оператор, действующий справа на категорию Φ и превращающий её 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/ эта Ψ сокращается со "знаменателем".

Далее, будем говорить, что α сокращется до β, если существует такая последовательность цепочек категорий над W $\alpha_0, \alpha_1, \ldots, \alpha_n$, что $\alpha_0 = \alpha$, $\alpha_n = \beta$ и при каждом $i = 1, \ldots, n$ α_{i-1} непосредственно сокращается до α_i.

Теперь мы можем сформулировать определение категориальной грамматики.

Категориальной грамматикой называется упорядоченная четверка $G = \langle V, W, \Phi_0, f \rangle$, где: I) - 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 над V $(a_1, \ldots, a_k \in V)$ и $\xi = \Phi_1 \ldots \Phi_k$ - цепочка над $K(W)$ $(\Phi_1, \ldots, \Phi_k \in K(W))$. Мы будем говорить, что грамматика G сопоставляет цепочке x цепочку ξ /или просто, что ξ сопоставлена x /, если для каждого $i = 1, \ldots, k$ $\Phi_i \in f(a_i)$. Будем, далее, говорить, что грамматика G приписывает цепочке x категорию Φ /или просто, что Φ приписана цепочке x /, если либо $x = a \in V$ и $\Phi \in f(a)$, либо некоторая цепочка категорий, сопоставленная x, сокращается до Φ.

Языком, определяемым категориальной грамматикой G /обозначение $L(G)$ / называется множество тех цепочек над основным словарем, которым грамматика G приписывает главную категорию.

В главной интерпретации V - множество словоформ некоторого конкретного языка; $K(W)$ - множество грамматических категорий /точнее, как грамматические категории интерпретируются те элементы $K(W)$, которые входят в значения функции f или являются частями категорий, входящих в значения f /; Φ_0 есть категория "предложение". Некоторые категории рассматриваются при этом как операторы; так, если S - категория "группа существительного /в именительном падеже/", то категория "группа глагола" имеет вид $[S\backslash\Phi_0]$ /оператор, действующий на группу существительного справа и превращающий её в предложение/, а категория "прилагательное" имеет вид $[S/S]$ / оператор, действующий на группу существительного слева и дающий снова группу существительного/. Язык $L(G)$ - это множество грамматически правильных фраз соответствующего конкретного языка.

В отличие от порождающих грамматик категориальные грамматики являются распознающими: категориальная грамматика не порождает множество грамматически правильных цепочек, но дает алгоритм, позволяющий для каждой цепочки над основным словарем узнать, является ли она грамматически правильной.*) Этот алгоритм состоит в следующем; нужно выписать все цепочки категорий, сопоставляемые данной цепочке грамматикой /число таких цепочек категорий, очевидно, конечно/; затем каждую из полученных цепочек сократить всеми возможными способами. Если хотя бы для одной цепочки категорий хотя бы одно сокращение приведет к главной категории, то данная цепочка над основным словарем грамматически правильна /принадлежит $L(G)$ /; в противном случае она не является грамматически правильной /не принадлежит $L(G)$ /.

В заключение этого параграфа докажем три простых леммы, относящихся к сокращению цепочек категорий.

<u>Лемма 4.1</u>. Пусть $K(W)$ - категориальная система, $\varphi_1, \dots, \varphi_n \in K(W)$ и цепочка $\varphi_1 \dots \varphi_n$ сокращается до одной категории. Тогда каждое вхождение категории φ_i участвует хотя бы в одном непосредственном сокращении.

<u>Доказательство.</u> Если φ_i не участвует ни в одном сокращении, то результат сокращения всей цепочки $\varphi_1 \dots \varphi_n$ имеет вид $\alpha \varphi_i \beta$, где α - результат сокращения цепочки $\varphi_1 \dots \varphi_{i-1}$ и β - результат сокращения цепочки $\varphi_{i+1} \dots \varphi_n$. Из цепочек $\varphi_1 \dots \varphi_{i-1}$ и $\varphi_{i+1} \dots \varphi_n$ по крайней мере одна непуста; поэтому хотя бы одна из цепочек α, β тоже непуста, и цепочка $\alpha \varphi_i \beta$ состоит более чем из одной категории.

<u>Лемма 4.2.</u> Пусть $K(W)$ — категориальная система и ξ, η - цепочки над W такие, что ξ сокращается до η. Если при этом $\eta = \eta_1 \eta_2$, то ξ можно представить в виде

*) Такие алгоритмы, впрочем, существуют и для многих порождающих грамматик /см. ниже § 30/. Но алгоритм, соответствующий категориальной грамматике, получается из её определения непосредственно - работа категориальной грамматики и есть применение алгоритма.

$\xi = \xi_1 \xi_2$, где ξ_1 сокращается до η_1 или равна η_1, а ξ_2 сокращается до η_2 или равна η_2.

<u>Доказательство</u>. Т.к. ξ сокращается до η, существует такая последовательность $\xi_0, \xi_1, \dots, \xi_n$ цепочек категорий, что $\xi_0 = \xi$, $\xi_n = \eta$ и ξ_{i-1} непосредственно сокращается до ξ_i. Утверждение леммы доказывается очевидной индукцией по n.

<u>Лемма 4.3.</u> Никакая цепочка категорий не может быть двумя разными способами непосредственно сокращена до одной и той же цепочки.

<u>Доказательство.</u> Пусть цепочка категорий $\xi = \varphi_1 \varphi_2 \dots \varphi_n$ ($\varphi_1, \dots, \varphi_n$ - категории) непосредственно сокращается двумя разными способами до цепочек η и η'. Это означает существование таких чисел i и j, $1 \le i < j \le n$, что η получается из ξ путем замены подцепочки $\varphi_i \varphi_{i+1}$ одной категорией, а η' - путем замены подцепочки $\varphi_j \varphi_{j+1}$ одной категорией. Ясно, что если $\varphi_i = [\Psi/\theta]$, $\varphi_{i+1} = \theta$, то $\eta \neq \eta'$; действительно, в этом случае в цепочке η на i-м месте, считая слева, будет стоять категория Ψ, а в η' на том же месте будет категория $[\Psi/\theta] \neq \Psi$. Аналогично обстоит дело в случае, когда $\varphi_j = \theta$, $\varphi_{j+1} = [\theta \backslash \Psi]$. Остается случай $\varphi_i = \theta$, $\varphi_{i+1} = [\theta \backslash \Psi]$, $\varphi_j = [\Psi'/\theta']$, $\varphi_{j+1} = \theta'$. Допустим, что $\eta = \eta'$; рассмотрим некоторое число s, удовлетворяющее условию $i+1 \le s < j$. (Равенство $i+1 = j$ невозможно, т.к. в рассматриваемом случае заведомо $\varphi_{i+1} \neq \varphi_j$). В цепочке η на s-м месте, считая слева, стоит категория φ_{s+1}, а в η' на том же месте стоит φ_s. Поэтому $\varphi_s = \varphi_{s+1}$. Итак, $\varphi_{i+1} = \varphi_{i+2} = \dots = \varphi_{j-1} = \varphi_j$, что невозможно.

<u>§ 25. Примеры.</u>

В каждом из нижеследующих примеров явно задается только приписывающая функция. Во всех случаях, когда некоторое значение $f(a)$ приписывающей функции f состоит из одного элемента Ψ, будем вместо $f(a) = \{\Psi\}$ писать $f(a) = \Psi$. Словарь категорий каждой грамматики будет состоять из всех встречающихся в описании её приписывающей функции прописных латинских и греческих букв. φ_0 всегда будет главной категорией.

<u>Пример. I.</u> Определим приписывающую функцию f_1 на сло-

варе $\{a,b\}$ так: $f_1(a)=\Phi_0$, $f_1(b)=[\Phi_0\setminus\Phi_0]$.
Легко видеть, что цепочка, состоящая из таких категорий, сокращается до Φ_0 /или равна Φ_0 / тогда и только тогда, когда она имеет вид $\Phi_0([\Phi_0\setminus\Phi_0])^n$ $(n=0,1,\ldots)$.

Таким образом, соответствующая грамматика G_1 определяет язык $\{ab^n/n=0,1,\ldots\}$.

Пример 2. Определим функцию f_2 на $\{a,b\}$ так: $f_2(a)=\{A,[A/\Phi_0]\}$, $f_2(b)=[A\setminus\Phi_0]$.
Ясно, что всякая цепочка категорий вида $([A/\Phi_0])^{n-1}A([A\setminus\Phi_0])^n$ $(n=1,2,\ldots)$ сокращается до Φ_0. С другой стороны, легко доказывается, что при любом $n=1,2,\ldots$ цепочка категорий, являющихся значениями f_2, длина которой равна $2n$, сокращается до Φ_0, только тогда, когда она имеет вид $([A/\Phi_0])^{n-1}A([A\setminus\Phi_0])^n$. (Для доказательства нужно к этому утверждению присоединить еще одно, а именно: при любом $n=1,2,\ldots$ цепочка категорий, являющихся значениями функции f_2 или частями значения f_2, длина которой равна $2n-1$, сокращается до Φ_0 /или равна Φ_0 /только тогда, когда она имеет вид $([A/\Phi_0])^{n-1}\Phi_0([A\setminus\Phi_0])^{n-1}$; оба утверждения без труда доказываются одновременной индукцией по n). Итак, категория Φ_0 приписывается соответствующей грамматикой G_2 всем цепочкам вида a^nb^n $(n=1,2,\ldots)$ и только им, т.е. $L(G_2)=\{a^nb^n/n=1,2,\ldots\}$

Пример 3. Определим функцию f_3 на словаре $V=\{p_1,\ldots,p_k,\neg,\&,\vee,\supset,(,)\}$ так: $f_3(p_i)=\Phi_0$ $(i=1,\ldots,k)$, $f_3(()=K$,

$f_3())=[\Phi_0\setminus[K\setminus\Phi_1]]$; $f_3(\neg)=[\Phi_0/\Phi_1]$;

$f_3(\&)=f_3(\vee)=f_3(\supset)=[\Phi_1\setminus[\Phi_0/\Phi_1]]$.

Соответствующую грамматику обозначим G_3.

Поясним содержательный смысл введенных категорий. Φ_0 означает правильно построенную формулу (ппф) алгебры логики в варианте, рассмотренном в примере 8 из § 12. Φ_1 означает ппф, заключенную в скобки. $[\Phi_0/\Phi_1]$ – оператор, который, действуя слева на ппф, заключенную в скобки, переводит её в ппф. $[\Phi_1\setminus[\Phi_0/\Phi_1]]$ – оператор, который, действуя справа на ппф, заключенную в скобки, превращает её в выражение вида $(\alpha)\&$, или $(\alpha)\vee$, или $(\alpha)\supset$, где α означает ппф, т.е. в оператор $[\Phi_0/\Phi_1]$ (Можно также рассматривать $[\Phi_1\setminus[\Phi_0/\Phi_1]]$ как двуместный оператор, который, будучи поставлен <u>между</u> двумя ппф, заключенными в скобки, превращает их в одну новую ппф). Наконец, $[\Phi_0\setminus[K\setminus\Phi_1]]$ – оператор, который, действуя справа на ппф α, превращает её в выражение $\alpha)$, т.е. в оператор $[K\setminus\Phi_1]$, действующий справа на левую скобку и превращающий её в ппф, заключенную в скобки.

Заметим, что, поскольку каждое значение f_3 есть одноэлементное множество, грамматика G сопоставляет каждой цепочке над V единственную цепочку категорий.

Покажем, что грамматика G_3 приписывает цепочке x над V категорию Φ_0 тогда и только тогда, когда x есть ппф /т.е. что $L(G_3)$ совпадает с языком L_c примера 8 из § 12/.

I. Докажем, что если x есть ппф, то G_3 приписывает цепочке x категорию Φ_0. Доказательство проведем индукцией по сложности ппф, т.е. по общему числу вхождений в ппф логических констант $\neg,\&,\vee,\supset$. Если x имеет сложность 0, то $x=p_i$ и $f(x)=\Phi_0$. Пусть утверждение доказано для всех ппф сложности $<n$, и пусть ппф x имеет сложность n. Возможны два случая: а) $x=\neg(y)$, где y – ппф сложности $<n$; б) $x=(y_1)\alpha(y_2)$, где y_1 и y_2 – ппф сложности $<n$ и α означает один из символов $\&,\vee,\supset$. В случае а) цепочка категорий ξ, сопоставляемая цепочке x, имеет вид $\xi=[\Phi_0/\Phi_1]K\eta[\Phi_0\setminus[K\setminus\Phi_1]]$, где η – цепочка категорий, сопоставляемая цепочке y. Поскольку η сокращается до Φ_0, ξ тоже сокращается до Φ_0. В случае б) цепочка ξ имеет вид

$$\xi=K\eta_1[\Phi_0\setminus[K\setminus\Phi_1]][\Phi_1\setminus[\Phi_0/\Phi_1]]K\eta_2[\Phi_0\setminus[K\setminus\Phi_1]],$$

где η_1 и η_2 – цепочки категорий, сопоставляемые соответственно цепочкам y_1 и y_2. Поскольку η_1 и η_2 сокращаются до Φ_0, цепочка ξ, как легко видеть непосредственно, тоже сокращается до Φ_0.

II. Докажем, что если x – цепочка над V и грамматика G_3 приписывает цепочке x категорию Φ_0, то x есть ппф.

Нам будет удобно доказывать вместо этого утверждения другое, более сильное, а именно: Пусть для категории θ $F(\theta)$ означает множество всех тех цепочек над V, которым грамматика G_3 приписывает категорию θ. Тогда: а) всякий элемент $F(\varphi_0)$ есть ппф; б) всякий элемент $F(\varphi_1)$ имеет вид (α), где α -ппф; в) всякий элемент $F([\varphi_0/\varphi_1])$ либо есть $\neg$, либо имеет вид $(\alpha)\lambda$, где α - ппф и λ - один из символов $\&, V, \supset$; г) $F(K)$ состоит из одного символа - левой скобки; д) $F([\varphi_0\backslash[K\backslash\varphi_1]])$ состоит из одного символа - правой скобки; е) $F([\varphi_1\backslash[\varphi_0/\varphi_1]]) = \{\&, V, \supset\}$.

Чтобы доказать г), д) и е), достаточно заметить, что никакая пара категорий, являющихся значениями или частями значений функции f_3, не может непосредственно сократиться до К, или до $[\varphi_0\backslash[K\backslash\varphi_1]]$, или до $[\varphi_1\backslash[\varphi_0/\varphi_1]]$. Отсюда сразу следует, что ни одна из этих трех категорий не может быть результатом сокращения цепочки категорий, сопоставленной какой-либо цепочке над V.

Утверждения а), б) и в) мы докажем одновременной индукцией по длине цепочки x.

Если цепочка x над V имеет длину I и ей приписана категория φ_0 или $[\varphi_0/\varphi_1]$, то, очевидно, x есть соответственно p_i или $\neg$; категория φ_1 не может быть приписана одноэлементной цепочке.

Допустим, что утверждения а), б), в) доказаны для всех цепочек длины $< n$. Пусть x - цепочка длины n, которой приписана категория θ. Фиксируем последовательность цепочек категорий $\xi_0, \xi_1, \ldots, \xi_n$ такую, что ξ_0 сопоставлена цепочке x, $\xi_n = \theta$ и для каждого $i = 1, \ldots, n$ ξ_{i-1} непосредственно сокращается до ξ_i.

Пусть сначала $\theta = \varphi_0$. Тогда непосредственно ясно, что ξ_{n-1} может иметь только вид $[\varphi_0/\varphi_1]\varphi_1$. По лемме 4.2 ξ_0 можно представить в виде $\xi_0'\xi_0''$, где ξ_0' сокращается до /или совпадает с/ $[\varphi_0/\varphi_1]$, а ξ_0'' сокращается до /или совпадает с/ φ_1. Далее, x можно представить в виде $x = x'x''$, где цепочке x' сопоставлена цепочка категорий ξ_0', а цепочке x'' - цепочка категорий ξ_0''. По индуктивному предположению x' есть либо $\neg$, либо $(\alpha)\lambda$, где α - ппф и λ - один из символов $\&, V, \supset$; по той же причине x'' имеет вид (β), где β -ппф. Следовательно, x есть либо $\neg(\beta)$, либо $(\alpha)\lambda(\beta)$; в обоих случаях x - ппф.

Пусть теперь $\theta = \varphi_1$. Ясно, что в этом случае $\xi_{n-1} = K[K\backslash\varphi_1]$; а эта цепочка, в свою очередь, может получиться только непосредственным сокращением из цепочки $K\varphi_0[\varphi_0\backslash[K\backslash\varphi_1]]$, так что именно такой вид имеет ξ_{n-2}. В силу леммы 4.2 /которая, очевидно, верна не только для случая двух сомножителей, но и для случая трех/ ξ_0 представляется в виде $\xi_0 = \xi_0'\xi_0''\xi_0'''$, где цепочки $\xi_0', \xi_0'', \xi_0'''$ соответственно сокращаются до /или совпадают с/ $K, \varphi_0, [\varphi_0\backslash[K\backslash\varphi_1]]$. Значит, x представляется в виде $x'x''x'''$, где цепочкам x', x'', x''' приписаны соответственно категории $K, \varphi_0, [\varphi_0\backslash[K\backslash\varphi_1]]$. В силу утверждений г) и д) x' есть левая скобка и x''' есть правая скобка, а по индуктивному предположению x'' есть ппф. Итак, $x = (\alpha)$, где α - ппф.

Пусть, наконец, $\theta = [\varphi_0/\varphi_1]$. Тогда $\xi_{n-1} = \varphi_1[\varphi_1\backslash[\varphi_0/\varphi_1]]$. Отсюда, аналогично предыдущему, следует, что x представляется в виде $x = x'x''$, где цепочкам x', x'' приписаны соответственно категории φ_1 и $[\varphi_1\backslash[\varphi_0/\varphi_1]]$. По индуктивному предположению x' имеет вид (α), где α -ппф, а в силу утверждения е) x'' есть $\&, V$ или $\supset$. Итак, x имеет вид $(\alpha)\lambda$, где α - ппф и $\lambda = \&, V, \supset$.

Пример 4. Назовем квазиформулой цепочку, которая получается из ппф в смысле предыдущего примера вычеркиванием всех скобок. Покажем, что множество всех квазиформул также определяется категориальной грамматикой.

Для этого определим функцию f_4 на словаре $V = \{p_1, \ldots, p_k, \neg, \&, V, \supset\}$ так: $f_4(p_i) = \varphi_0$; $f_4(\neg) = [\varphi_0/\varphi_0]$, $f_4(\&) = f(V) = f_4(\supset) = [\varphi_0\backslash[\varphi_0/\varphi_0]]$. Соответствующую грамматику обозначим G_4.

Докажем, что $L(G_4)$ совпадает с множеством всех квазиформул.

I. Тот факт, что всякой квазиформуле приписывается категория φ_0, доказывается очевидной индукцией по сложности квазиформулы.

П. Чтобы доказать, что всякая цепочка над V, которой приписывается категория Φ_0, есть квазиформула, установим справедливость более общего утверждения, а именно: Пусть для категории θ $F(\theta)$ означает множество всех тех цепочек, которым грамматика G_4 приписывает категорию θ. Тогда: а) всякий элемент $F(\varphi_0)$ есть квазиформула; б) всякий элемент $F([\varphi_0/\varphi_0])$ либо есть $\neg$, либо имеет вид $\alpha\lambda$, где α - квазиформула и $\lambda = \&, V, \supset$; в) $F([\varphi_0\backslash[\varphi_0/\varphi_0]]) = \{\&, V, \supset\}$.

Утверждение в) доказывается в точности так же, как утверждение е) в предыдущем примере. Утверждения а) и б) доказываются одновременной индукцией по длине цепочки; детальное проведение соответствующих рассуждений предоставляется читателю.

Пример 5. Рассмотрим словарь $V=\{a_1,\dots,a_\varkappa,+,\cdot,(,),=\}$. Определим терм следующим образом: (i) все a_i - термы; (ii) если t_1 и t_2 - термы, то $(t_1)+(t_2)$ и $(t_1)\cdot(t_2)$ - также термы; (iii) всякий терм является таковым только в силу (i) или (ii). Выражения вида $t_1=t_2$, где t_1, t_2 - термы, назовем формулами.

Построим категориальную грамматику, определяющую множество всех формул. Для этого определим функцию f_5 на словаре V так: $f_5(a_i)=T$; $f_5(+)=f_5(\cdot)=[T_1\backslash[T/T_1]]$;
$f_5(()=K$; $f_5())=[T\backslash[K\backslash T_1]]$;
$f_5(=)=[T\backslash[\varphi_0/T]]$

Соответствующую грамматику обозначим G_5.

Рассуждением, совершенно аналогичным примененному при рассмотрении примера 3, можно доказать, что грамматика G_5 приписывает цепочке x над V категорию T тогда и только тогда, когда x есть терм. Отсюда сразу следует, что всякой формуле грамматика G_5 приписывает категорию φ_0. Обратно, пусть G_5 приписывает цепочке x над V категорию φ_0, и пусть $\xi_0,\xi_1,\dots,\xi_n$ — последовательность цепочек категорий такая, чт[о] ξ_0 сопоставлена цепочке x, $\xi_n=\varphi_0$ и ξ_{i-1} непосредственно сокращается до ξ_i. Ясно, что $\xi_{n-1}=[\varphi_0/T]\,T$, откуда $\xi_{n-2}=T[T\backslash[\varphi_0/T]]T$. Поэтому x можно представить в виде $x'x''x'''$, где цепочкам x', x'', x'''

и положим $\Gamma^*=\langle V, V_1^*, Пр.^*, S^*\rangle$,
где $V_1^*=\bar V\cup V_1\cup V_1'\cup\{Пр.^*\}\cup V'$,
а S^* получается из $\bar S\cup S'$ добавлением новых правил:

$$Пр.^*\longrightarrow Пр.\,Пр.'$$

$$\bar a b\longrightarrow b\bar a,\qquad a\in V,\ b\in V'$$

$$b\bar a\longrightarrow a,\qquad a\in V,\ b\in V'.$$

Построенная грамматика, очевидно, является грамматикой без существенного укорачивания, т.к. в ее схеме укорачивающими являются только правила вида $b\bar a\to a$, а они- заключительные.

Легко видеть, что цепочка x над V выводима из $Пр.$ в Γ^* тогда и только тогда, когда $x\in L(\Gamma)$ и язык $L(\Gamma')$ содержит цепочку, длина которой равна длине x. Лемма доказана.

Для произвольного языка L и произвольного натурального числа n будем полагать

$$L^{(n)}=\{x\mid x\in L \,\&\, \ell(x)\le n\},$$

$$\overline{L}^{(n)}=\{x\mid x\in L \,\&\, \ell(x)\ge n\}$$

Лемма 5.5. Если L - нс-язык и K -класс языков, содержащий пустой язык и не содержащий ни одного из языков $\overline{L}^{(n)}$, то свойство принадлежать классу K нераспознаваемо в классе нс-грамматик.

Лемма 5.6. Если L -нс-язык и K -класс языков, содержащий L и не содержащий ни одного из языков $L^{(n)}$, то свойство принадлежать классу K нераспознаваемо в классе нс-грамматик.

Доказательство леммы 5.5.

Пусть T - произвольная машина Тьюринга и f_T-вычисляемая тор, который, будучи поставлен между этими группами, дает одну такую же группу. $[V^{*}_{xux} / V^{*}_{xux}]$ - препозитивное определение к глаголу /обстоятельственное слово/, т.е. опера - тор, действующий слева на глагольную группу и переводящий её снова в глагольную группу. $[V'_{xux} \backslash V^{*}_{xux}]$ - дополнение, выраженное предложной группой с предлогом за, т.е. оператор, действующий справа на глагол, управляющий предлогом за, и превращающий его в "полную" группу глагола. Наконец, $[Z \backslash [V'_{xux} \backslash V^{*}_{xux}]]$ — оператор, действующий справа на предлог за и превращающий его в дополнение, выраженное предложной группой с за.

Рассмотрим цепочку народ узнал колокольчик Пугачева и толпою бежал за нами. Грамматика G_6 сопоставляет ей, в числе про - чих, следующую цепочку категорий:

$$[\Phi_0 / V^{*}_{131}]\, V_{131}\, [V_{131} \backslash V^{*}_{131}]\, [[V_{131} \backslash V^{*}_{131}] \backslash [V_{131} \backslash V^{*}_{131}]]$$

$$[V^{*}_{131} \backslash [V^{*}_{131} / V^{*}_{131}]]\, [V^{*}_{131} / V^{*}_{131}]\, V'_{131}\, Z\, [Z \backslash [V'_{131} \backslash V^{*}_{131}]]$$

Читателю предоставляется проверить, что эта цепочка категорий сокращается до Φ_0, так что соответствующая цепочка над V принадлежит языку $L(G_6)$.

§ 26. Односторонние монотонные категориальные системы.

Пусть $K(W)$ - категориальная система над словарем категорий W. Среди категорий, принадлежащих $K(W)$, мы выделим некоторые категории специального вида, которые будем называть левосторонними монотонными категориями над W / сокращенно лм-категориями/. Определение лм-категории формулируется следующим образом:

I. Всякая элементарная категория есть лм-категория.

П. Если Θ - лм-категория и Φ - элементарная категория, то $[\Phi \backslash \Theta]$ есть лм-категория. Ш. Всякая лм-категория является таковой либо в силу п.I, либо в силу п.П.

Очевидно, всякая неэлементарная лм-категория имеет вид

$$[\Phi_1 \backslash [\Phi_2 \backslash [\Phi_3 \backslash \ldots \backslash [\Phi_{n-1} \backslash \Phi_n] \ldots]],$$

где $\Phi_1, \ldots, \Phi_n$ - элементарные категории. Отсюда ясно, что всякая цепочка, получающаяся в результате сокращения цепочки лм-категорий, также состоит из лм-категорий.

Множество всех лм-категорий над W мы будем называть левосторонней монотонной категориальной системой над W и обозначать $K_{лм}(W)$.

Категориальную грамматику $\langle V, W, \Phi_0, f \rangle$ мы будем называть левосторонней монотонной грамматикой / сокращенно лм-грамматикой/, если все значения функции f содержатся в $K_{лм}(W)$.

Совершенно аналогично /заменив символ \ на / / можно было бы определить правосторонние монотонные категории и правосторонние монотонные грамматики.

Докажем теперь две леммы, которые понадобятся нам в следующем параграфе.

Лемма 4.4. Если цепочка лм-категорий $\xi = \Phi_1 \Phi_2 \ldots \Phi_k$ ($\Phi_1, \ldots, \Phi_k \in K_{лм}(W)$) сокращается до одной категории Δ, то : а) Φ_1 есть элементарная категория ; б) Φ_k имеет вид $\Phi_k = [\Psi_1 \backslash [\Psi_2 \backslash \ldots \backslash [\Psi_r \backslash \Delta] \ldots]$, где $\Psi_1, \ldots, \Psi_r$ - элементарные категории.

Доказательство. а) По лемме 4.I Φ_1 должна участвовать хотя бы в одном непосредственном сокращении. Но Φ_1 может сокращаться только с категорией, стоящей справа от неё; а в цепочке лм-категорий при непосредственном сокращении из двух участвующих в этом сокращении категорий левая всегда является элементар - ной. б) Поскольку категория Φ_k также участвует хотя бы в одном непосредственном сокращении и при этом может сокращаться только с категориями, стоящими слева от неё, Φ_k не может быть элементарной. Следовательно, $\Phi_k = [\Psi_1 \backslash [\Psi_2 \backslash \ldots [\Psi_{s-1} \backslash \Psi_s] \ldots]$, где $\Psi_1, \ldots, \Psi_s$ - элементарные категории. Поскольку Φ_k в каждом непосредственном сокращении, в котором она участвует, является "поглощающим" участником, она ни в одном сокращении не исчезнет полностью. Та категория, в которую превратится Φ_k после всех сокращений, и есть, очевидно, Δ. Но эта категория имеет вид $[\Psi_{r+1} \backslash [\Psi_{r+2} \backslash \ldots \backslash [\Psi_{s-1} \backslash \Psi_s] \ldots]$, где r - число непосред-

Доказательство. а/ сразу следует из того, что число пучков бесконечно; б/ – из того, что конечные языки образуют пучок (и из того, что отрицание нераспознаваемого свойства нераспознаваемо). Чтобы доказать в/, достаточно заметить, что всякий язык, почти совпадающий с кс-языком, сам является кс-языком (это следует из теорем 3.11 и 3.12 , поскольку всякий язык с конечным дополнением – автоматный), и поэтому существуют пучки, не содержащие кс-языков.

В частности, нераспознаваемы в классе нс-грамматик следующие свойства языков: быть пустым (по пункту а/ или в/), иметь пустое дополнение (по пункту а/или в/), быть конечным (по пункту б/или в/), иметь конечное дополнение (по пункту в/), быть автоматным (по пункту б/или в/), быть контекстно-свободным (по пункту б/или в/), иметь контекстно-свободное дополнение (по пункту в/).

Следствие 2. Ни для одной нс-грамматики Γ_0 не существует алгоритма, позволяющего по любой нс-грамматике Γ узнавать, эквивалентна ли она грамматике Γ_0 .

Доказательство. Обозначим через Θ_0 свойство языка совпадать с языком $L(\Gamma_0)$. Поскольку этим свойством обладает только один язык, оно нераспознаваемо по пункту а) следствия I.

Следствие 3. Если язык L_0 (не обязательно нс-) имеет бесконечное дополнение, то не существует алгоритма, позволяющего по нс-грамматике Γ узнавать, содержится ли язык $L(\Gamma)$ в языке L_0 .

Доказательство. Обозначим через Θ_0' свойство языка содержаться в L_0 . Всякий язык, обладающий свойством Θ_0' , имеет бесконечное дополнение, но языки с конечными дополнениями образуют пучок.

Следствие 4. Если язык L_0 бесконечен, то не существует: а/ алгоритма, позволяющего по нс-грамматике Γ узнавать, содержится ли язык L_0 в языке $L(\Gamma)$; б/ алгоритма, позволяющего по нс-грамматике Γ узнавать, является ли пересечение $L_0 \cap L(\Gamma)$ пустым.

Доказательство. а/ Обозначим через Θ_0'' свойство языка содержать L_0 и через $\overline{\Theta_0''}$ отрицание свойства Θ_0'' .

Всякий язык, обладающий свойством Θ_0'' , бесконечен; поэтому все конечные языки обладают свойством $\overline{\Theta_0''}$, и по пункту б/ следствия I оно– а вместе с ним и Θ_0'' -нераспознаваемо. б/ вытекает из следствия 3, поскольку $L_0 \cap L(\Gamma) = \varnothing$ равносильно $L(\Gamma) \subseteq CL_0$.

Замечание. Нетрудно видеть, что для конечного языка L_0 алгоритмы, указанные в формулировке следствия 4, существуют. Это вытекает из замечания, сделанного в начале настоящего параграфа. Точно так же, если язык L_0 имеет конечное дополнение, то существует алгоритм, указанный в формулировке следствия 3.

§ 33. Проблемы распознавания свойств кс-языков.

Для кс-языков класс распознаваемых свойств шире, чем для нс-языков – пустота, конечность и некоторые другие свойства, нераспознаваемые в классе нс- языков, в классе кс-языков распознаваемы. Однако многие важные свойства остаются нераспознаваемыми и здесь. Чтобы установить это, введем одно новое понятие, относящееся к машинам Тьюринга.

Пусть T -машина Тьюринга с внешним словарем $\{a_0, a_1\}$ и внутренним словарем $\{q_0, q_1, \dots, q_s\}$ (q_1 – начальное состояние , q_0 – заключительное). Введем два новых символа q, d . Для произвольной машинной цепочки ω назовем ее образом цепочку, полученную из ω заменой вхождения символа q_i вхождением подцепочки q^{i+1} . Будем называть протоколом машины T всякую цепочку вида

$$d\omega_0 dd\omega_1 d \dots d\omega_t d$$,где:

1/ ω_0 – образ машинной цепочки ω , содержащей q_1 ;
2/ для всякого $i=1,\dots,t$ ω_i есть образ цепочки $\omega^{(i)}$;
3/ ω_t является образом цепочки, содержащей q_0 . Множество всех протоколов машины T будем обозначать Π_T . Положим $W=\{a_0, a_1, q, d\}$ и $\Pi_T' = F'(W) \setminus \Pi_T$ ($F'(W)$ означает множество всех непустых цепочек над W) .

Основную роль в конструкциях этого параграфа будет играть следующая лемма:

Лемма 5.7. Для любой машины Тьюринга T с внешним словарем

Доказательство теоремы 4.2. Пусть $\Gamma = \langle V, V_1, \text{Пр.}, S \rangle$ - кс-грамматика. Мы будем считать, что Γ удовлетворяет условию леммы 3.4.

Пусть задан некоторый вывод $\mathfrak{A} = (X_0, X_1, \ldots, X_n)$ в грамматике Γ, причем X_0 состоит из одного символа B_0 и все цепочки $X_0, \ldots, X_n$ состоят только из вспомогательных символов. Мы будем называть вывод $\mathfrak{A}$ простым, если при некотором способе проведения этого вывода на каждом его шаге заменяется последняя точка соответствующей цепочки, иначе говоря, если цепочки $X_0, \ldots, X_n$ можно представить в виде

$$\begin{aligned} X_0 &= B_0 \\ X_1 &= A_1 B_1 \\ X_2 &= A_1 A_2 B_2 \\ &\ldots\ldots \\ X_{n-1} &= A_1 A_2 \ldots A_{n-1} B_{n-1} \\ X_n &= A_1 A_2 \ldots A_{n-1} A_n B_n \end{aligned}$$

и схема S содержит правила $B_0 \to A_1 B_1,\ B_1 \to A_2 B_2, \ldots, B_{n-1} \to A_n B_n$.

Теперь мы обобщим понятие простого вывода на случай, когда исходная цепочка состоит более чем из одного символа. Именно, последовательность цепочек над V_1 $(Y_0, Y_1, \ldots, Y_n)$ мы будем называть простым выводом, если цепочки $Y_0, Y_1, \ldots, Y_n$ представимы в виде

$$\begin{aligned} Y_0 &= Y' B_0 Y'' \\ Y_1 &= Y' A_1 B_1 Y'' \\ &\ldots\ldots \\ Y_n &= Y' A_1 \ldots A_n B_n Y'', \end{aligned}$$

где последовательность $(B_0, A_1 B_1, \ldots, A_1 \ldots A_n B_n)$ является простым выводом в смысле предыдущего определения. Точку B_0 цепочки Y_0 мы будем в этом случае называть главной точкой данной цепочки.

Пусть теперь $\mathfrak{A} = (Z_0, Z_1, \ldots, Z_n)$ - вывод в Γ с фиксированным способом проведения, причем все цепочки $Z_0, \ldots, Z_n$ состоят только из вспомогательных символов и $Z_0 = \text{Пр.}$ Для каждого $i = 0, 1, \ldots, n$ мы определим понятия левой и правой точки цепочки Z_i. Эти понятия будут определяться индуктивно следующим образом: (i) единственная точка цепочки Z_0 считается левой; (ii) если для цепочки Z_{i-1} понятия левой и правой точки определены и если $Z_{i-1} = Z' A Z''$, $Z_i = Z' B C Z''$, то точка B считается левой, точка C - правой, а все остальные точки цепочки Z_i считаются левыми, соответственно правыми, если таковыми являются их непосредственные предки.

Далее, вывод $\mathfrak{A} = (Z_0, Z_1, \ldots, Z_n)$ с фиксированным способом проведения, состоящий из цепочек над V_1, и такой, что $Z_0 = \text{Пр.}$, мы будем называть кусочно-простым, если существует такая последовательность чисел $0 = i_0 < i_1 < \ldots < i_k = n$, что для каждого $j = 1, \ldots, k$ последовательность цепочек $(Z_{i_{j-1}}, Z_{i_{j-1}+1}, \ldots, Z_{i_j})$ является простым выводом, и при этом главная точка цепочки $Z_{i_{j-1}}$ является левой.

Таким образом, кусочно-простой вывод состоит из "блоков" или "макрошагов", на каждом из которых в цепочку вместо некоторой точки B_0 вставляется цепочка $A_1 \ldots A_n B_n$, просто выводимая из B_0, и при этом на следующих "макрошагах" может заменяться любая из точек $A_1, \ldots, A_n$ /все эти точки - левые/, но не B_n /эта точка - правая/.

Пример. Пусть грамматика Γ содержит правила $\text{Пр.} \to AB$, $A \to C\mathcal{D}$, $\mathcal{D} \to BA$, $C \to \mathcal{D}A$. Тогда следующий вывод будет кусочно-простым:

$$\begin{aligned} X_0 &= \text{Пр.} \\ X_1 &= A\,\underline{B} \\ X_2 &= C\,\underline{\mathcal{D}}\,\underline{B} \\ X_3 &= C B \underline{A}\,\underline{B} \\ X_4 &= C B C \underline{\mathcal{D}}\,\underline{B} \\ X_5 &= C B C B \underline{A}\,\underline{B} \\ X_6 &= C B \mathcal{D} \underline{A} B \underline{A}\,\underline{B} \\ X_7 &= C B \mathcal{D} C \underline{\mathcal{D}} B \underline{A}\,\underline{B} \\ X_8 &= C B \mathcal{D} C B \underline{A} B \underline{A}\,\underline{B} \end{aligned}$$

R_1 порождается, как легко проверить, кс-грамматикой со следующей схемой :

$$
\begin{aligned}
Пр &\longrightarrow dAd \\
A &\longrightarrow A a_i \,, \quad i = 0,1 \\
A &\longrightarrow A q \\
A &\longrightarrow B a_i \,, \quad i = 0,1 \\
B &\longrightarrow a_i B a_j \,, \quad i,j = 0,1 \\
B &\longrightarrow B q \\
B &\longrightarrow q B \\
B &\longrightarrow d d
\end{aligned}
$$

Совершенно аналогично строится кс- грамматика, порождающая R_2 . Отсюда, поскольку $R = R_1 \cup R_2$, следует, что и R порождается кс- грамматикой.

Пусть теперь $\omega_1 = \alpha_u \alpha_{u-1} \dots \alpha_1$, $\omega_2 = \beta_v \beta_{v-1} \dots \beta_1$,

где каждое α_k и β_m есть либо один из символов a_0, a_1 , либо степень символа q , и $u \le v$. Мы будем говорить, что цепочка ω_2 нарушает согласование с цепочкой ω_1 на к-м месте ($k = 1, 2, \dots, u-1$) , если выполняется одно из двух: I/ $\alpha_{k+1}\alpha_k \neq q^{\alpha+1} a_i$ и $\beta_{k+1}\beta_k \neq \alpha_{k+1}\alpha_k$;

2/ $\alpha_{k+1}\alpha_k = q^{\alpha+1} a_i$ и $\beta_{k+1}\beta_k \neq q^{\beta+1} a_j$.

Множество всех цепочек вида $d\omega_1 dd\omega_2 d$,где ω_1 , ω_2 - цепочки над $\{a_0, a_1, q\}$ такие, что ω_2 на некотором месте нарушает согласование с ω_1 , мы обозначим $S^{i\alpha}_{j\beta}$.

Ясно, что цепочка $d\omega_1 dd\omega_2 d$ принадлежит $P^{i\alpha}_{\tau}$ тогда и только тогда, когда она, во-первых, принадлежит $Q_{i\alpha}$ и, во-вторых, либо $\ell'(\omega_1) \neq \ell'(\omega_2)$,либо цепочка ω_2 на каком-то месте нарушает согласование с ω_1 .

определению числа j точка $C_j = B_{j-k}$ не заменяется на $(j+1)$-м шаге, а это возможно только при $j = \ell$.

Итак, последний из потомков точки C , к которому применяется правило, есть $C_{\ell-1} = B_{j-1-k}$.Начиная с C_ℓ ,все потомки переписываются без изменения, так что символ C_ℓ совпадает с C . Мы можем, следовательно, после ℓ-го шага вывода $\mathcal{L}_0$ заменить точку $B_{\ell-k} = C_\ell$ на $\mathcal{D}F$,а затем проделать все шаги, как в выводе $\mathcal{L}_0$; полученный вывод будет, как и $\mathcal{L}_0$,кусочно-простым, и его последний цепочкой будет Z_n .Утверждение доказано.

Теперь мы можем перейти к построению лм-грамматики.

Каждой упорядоченной паре (С,Д), где $C, \mathcal{D} \in V_1$, сопоставим новый символ $C^{\mathcal{D}}$. Множество всех таких новых символов обозначим W' . Введем, кроме того, еще один новый символ $\Phi_0 \bar{\in} W'$. Положим, $W = W' \cup \{\Phi_0\}$.

Прежде чем определять приписывающую функцию f , построим вспомогательную функцию f^* , сопоставляющую каждому символу из V_1 конечное множество категорий над W . Эту функцию мы определим следующим образом:

а) $\Phi_0 \in f^*(Пр.)$;

б) Если $A \to BC$ - правило грамматики Γ и $\mathcal{D}$ - произвольный элемент V_1 , то категории C^A и $[A^{\mathcal{D}} \setminus C^{\mathcal{D}}]$ принадлежат $f^*(B)$.

в) Если $B, \mathcal{D} \in V_1$ и Δ - категория, приписанная символу $\mathcal{D}$ по одному из пунктов а), б), то $[B^{\mathcal{D}} \setminus \Delta] \in f^*(B)$.

г) Для каждого $B \in V_1$ множество $f^*(B)$ состоит только из тех категорий, которые входят в него согласно пунктам а), б), в).

Категории, построенные согласно пунктам а) и б), мы будем называть левыми, построенные согласно пункту в) - правыми.

Заметим, что если существует простой вывод цепочки $A_1 \dots A_n B_n$ из символа B_0 и если Δ - произвольная левая категория, принадлежащая $f^*(B_0)$, то функция f^* сопоставляет цепочке $A_1 \dots A_n B_n$ некоторую цепочку категорий, сокращающуюся до Δ , причем в этой цепочке категория, сопоставленная B_n — правая, а все остальные - левые. Действительно, в

этом случае схема грамматики Γ содержит правила вида
$B_0 \to A_1 B_1,\ B_1 \to A_2 B_2, \ldots, B_{n-1} \to A_n B_n$;
по пункту б) определения f^* имеем $B_1^{B_0} \in f^*(A_1)$,
$[B_1^{B_0} \backslash B_2^{B_0}] \in f^*(A_2), \ldots, [B_{n-1}^{B_0} \backslash B_n^{B_0}] \in f^*(A_n)$,
а по пункту в) — $[B_n^{B_0} \backslash \Delta] \in f^*(B_n)$. Цепочка
$B_1^{B_0}[B_1^{B_0} \backslash B_2^{B_0}] \ldots [B_{n-1}^{B_0} \backslash B_n^{B_0}][B_n^{B_0} \backslash \Delta]$, очевидно, сокращается до Δ.

Определим теперь функцию f следующим образом: $\Psi \in f(a)$ тогда и только тогда, когда существует такой символ $A \in V_1$, что $A \models a(\Gamma)$ и $\Psi \in f^*(A)$.

Положим $G = \langle V, W, \Phi_0, f \rangle$. Из определения функции f ясно, что G есть лм-грамматика и её сложность равна двум*).

Покажем, что $L(G) = L(\Gamma)$. Для этого, как сразу видно из определения функции f, достаточно доказать, что цепочка над V_1 выводима в грамматике Γ из /или совпадает с/ Пр. тогда и только тогда, когда некоторая цепочка категорий, сопоставляемая ей с помощью функции f^*, сокращается до /или совпадает с/ Φ_0.

I. Пусть цепочка X над V_1 выводима в грамматике Г из /или совпадает с/ Пр. Покажем, что цепочке X можно сопоставить с помощью функции f^* некоторую цепочку категорий, сокращающуюся до /или совпадающую с/ Φ_0, и притом так, чтобы выполнялось следующее условие: каков бы ни был кусочно-простой вывод цепочки X из Пр., если в соответствии с этим выводом определить левые и правые точки цепочки X, то соответствующую цепочку категорий можно подобрать так, что левым точкам будут сопоставлены левые категории, а правым - правые. Доказательство проведем индукцией по длине X.

Если $\ell(X) = 1$, то $X = Пр.$, так что $\Phi_0 \in f^*(X)$.

Пусть утверждение доказано для всех цепочек длины $< n$,

*) В случае, когда $\mathcal{S}$ содержит хоть одно правило вида $A \to BC$. Если $\mathcal{S}$ состоит только из правил вида Пр. $\to a$, то сложность G равна I.

и пусть $\ell(X) = n$, $X = C_1 C_2 \ldots C_n$. По ранее доказанному имеется кусочно-простой вывод цепочки X из Пр. Пусть $(Пр. = X_0, X_1, \ldots, X_n = X)$ - произвольный такой вывод. В силу кусочной простоты этого вывода существует такое число s, $0 \leq s < n$, что цепочка $X_n = X$ просто выводима из X_s и главная точка цепочки X_s - левая; при этом вывод $(X_0, X_1, \ldots, X_s)$ - также кусочно-простой. Цепочки X_s и X можно представить соответственно в виде
$X_s = X' B_0 X''$, $X = X' A_1 \ldots A_t B_t X''$, где цепочка $A_1 \ldots A_t B_t$ просто выводима из символа B_0. По индуктивному предположению существует цепочка категорий ξ, сопоставляемая функцией f^* цепочке X_s, сокращающаяся до Φ_0 и такая, что левым точкам цепочки X_s /определенным в соответствии с выводом $(X_0, \ldots, X_s)$/ сопоставлены левые категории, а правым - правые. Пусть $\xi = \xi' \Delta \xi''$, где ξ', ξ'' - цепочки категорий, сопоставленные соответственно цепочкам X' и X'', а Δ - категория, сопоставленная символу B_0. Поскольку B_0 - главная точка цепочки X_s, категория Δ - левая. По доказанному выше цепочке $A_1 \ldots A_t B_t$ можно сопоставить цепочку категорий $\Psi_1 \Psi_2 \ldots \Psi_t \theta$, сокращающуюся до Δ, и такую, что категории $\Psi_1, \ldots, \Psi_t$ - левые, а θ - правая. Следовательно, цепочке $X = X' A_1 \ldots A_t B_t X''$ сопоставляется цепочка категорий $\xi' \Psi_1 \ldots \Psi_t \theta \xi''$, сокращающаяся до $\xi' \Delta \xi''$ и далее до Φ_0; в этой цепочке все категории, соответствующие левым точкам цепочки X, /определенным в соответствии с выводом $(X_0, \ldots, X_n = X)$/ - левые, а соответствующие правым-правые.

Пример. Если мы обратимся к примеру кусочно-простого вывода, рассмотренному выше, то его "главным" цепочкам X_0, X_1, X_5 и X_8 описанная только что процедура сопоставит следующие цепочки категорий/правые категории подчеркнуты/:

$\xi_0 = \Phi_0$

$\xi_1 = B^{Пр}\underline{[B^{Пр} \backslash \Phi_0]}$

$\xi_5 = \Phi^A[\Phi^A \backslash A^A][A^A \backslash \Phi^A][\Phi^A \backslash A^A]\underline{[A^A \backslash B^{Пр}]}\,\underline{[B^{Пр} \backslash \Phi_0]}$

$\xi_8 = \Phi^A[\Phi^A \backslash A^A] A^C [A^C \backslash \Phi^C][\Phi^C \backslash A^C]\underline{[A^C \backslash [A^A \backslash \Phi^A]]}[\Phi^A \backslash A^A]\underline{[A^A \backslash B^{Пр}]} \cdot$

$\cdot \underline{[B^{Пр} \backslash \Phi_0]}$

Здесь, например, подцепочка цепочки ξ_8

$A^c[A^c\backslash \mathcal{D}^c][\mathcal{D}^c\backslash A^c][A^c\backslash [A^A\backslash \mathcal{D}^A]]$ сокращается до $[A^A\backslash \mathcal{D}^A]$, и при этом ξ_8 превращется в ξ_5 .

П. Пусть цепочке X над V_1 функция f^* сопоставляет /в числе прочих/ некоторую цепочку категорий ξ , сокращающуюся до /или совпадающую с/ φ_0 . Покажем, что цепочка X выводима в грамматике из /или совпадает с/ Пр. Доказательство проведем индукцией по длине X .

Если $\ell(X)=1$,то $\xi = \varphi_0$;но единственный символ $A \in V_1$, для которого $\varphi_0 \in f^*(A)$, есть Пр.

Пусть утверждение доказано для всех цепочек длины $< n$ и пусть $\ell(X)=n$, $X=C_1 \ldots C_n$. Тогда $\xi=\Psi_1 \ldots \Psi_n$, где $\Psi_1, \ldots, \Psi_n$ - категории. В силу леммы 4.4. Ψ_1 есть элементарная категория, а Ψ_n имеет вид $[\theta\backslash \varphi_0]$ или $[\theta_1\backslash [\theta_2\backslash \varphi_0]]$. По определению функции f^* φ_0 не может входить в состав неэлементарных левых категорий. Поэтому категория Ψ_n - правая. Итак, цепочка $\xi=\Psi_1 \ldots \Psi_n$ начинается элементарной категорией и заканчивается правой категорией. Поэтому существуют такие числа r и s , $1 \leq r < n$, $1 \leq s \leq n-r$, что Ψ_r - элементарная категория, Ψ_{r+s} - правая категория, а все категории $\Psi_{r+1}, \ldots, \Psi_{r+s-1}$ /если только $s>1$ / - левые неэлементарные. В силу определения правой категории $\Psi_{r+s}=[\Omega\backslash\Delta]$, где Δ - левая категория. В то же время сложность каждой из категорий $\Psi_{r+1}, \ldots, \Psi_{r+s-1}$ равна единице. По лемме 4.5 существует такой способ сокращения цепочки $\xi=\Psi_1 \ldots \Psi_n$ до φ_0 , при котором одна из промежуточных цепочек есть $\xi'=\Psi_1 \ldots \Psi_{r-1}\,\Delta\,\Psi_{r+s+1} \ldots \Psi_n$.

Пусть $\mathcal{D}$ - тот символ из V_1 , для которого $\Delta \in f^*(\mathcal{D})$.
Положим $X'=C_1 \ldots C_{r-1}\,\mathcal{D}\,C_{r+s+1} \ldots C_n$. Цепочке X' функция f^* сопоставляет цепочку категорий ξ'. Поскольку ξ' сокращается до φ_0 , цепочка X' - в силу индуктивного предположения - выводима в грамматике Г из Пр. Если мы теперь докажем, что цепочка $C_r C_{r+1} \ldots C_{r+s}$ выводима в Г из Д, то X окажется выводимой из Пр.,и тем самым доказательство теоремы будет завершено.

Цепочке $C_r C_{r+1} \ldots C_{r+s}$ функция f^* сопоставляет цепочку категорий $\Psi_r \Psi_{r+1} \ldots \Psi_{r+s-1} \Psi_{r+s} =$

$$= \theta_1[\Omega_2\backslash\theta_2][\Omega_3\backslash\theta_3] \ldots [\Omega_s\backslash\theta_s][\Omega\backslash\Delta],$$

где $\theta_1, \Omega_2, \theta_2, \Omega_3, \theta_3, \ldots, \Omega_s, \theta_s, \Omega$ - элементарные категории. Эта цепочка, как мы уже знаем, сокращается до Δ ; а это возможно только при условии, что

$$\theta_1=\Omega_2,\ \theta_2=\Omega_3, \ldots, \theta_{s-1}=\Omega_s,\ \theta_s=\Omega .$$

Таким образом, $\Psi_r \ldots \Psi_{r+s} = \theta_1[\theta_1\backslash\theta_2] \ldots [\theta_{s-1}\backslash\theta_s][\theta_s\backslash\Delta]$.
Поскольку $\Delta \in f^*(\mathcal{D})$ и $[\theta_s\backslash\Delta] \in f^*(C_{r+s})$, категория θ_s - в силу пункта в) определения функции f^* должна иметь вид $C^{\mathcal{D}}_{r+s}$. А отсюда - ввиду того,что $[\theta_{s-1}\backslash\theta_s] = [\theta_{s-1}\backslash C^{\mathcal{D}}_{r+s}]$ есть левая категория - вытекает, что $\theta_{s-1}=B^{\mathcal{D}}_{s-1}$, где B_{s-1} - некоторый вспомогательный символ. Точно таким же образом получаем, что

$\theta_{s-2}=B^{\mathcal{D}}_{s-2}, \ldots, \theta_1=B^{\mathcal{D}}_1$, где $B_{s-2}, \ldots, B_1 \in V_1$

Итак, функция f^* сопоставляет символам $C_r, C_{r+1}, \ldots, C_{r+s-2}, C_{r+s-1}$ соответственно категории $B^{\mathcal{D}}_1$, $[B^{\mathcal{D}}_1\backslash B^{\mathcal{D}}_2], \ldots, [B^{\mathcal{D}}_{s-2}\backslash B^{\mathcal{D}}_{s-1}], [B^{\mathcal{D}}_{s-1}\backslash C^{\mathcal{D}}_{r+s}]$.
По определению функции f^* /пункт δ) / отсюда вытекает наличие в схеме грамматики Г правил $\mathcal{D} \to C_r B_1$, $B_1 \to C_{r+1} B_2, \ldots, B_{s-2} \to C_{r+s-2} B_{s-1}$, $B_{s-1} \to C_{r+s-1} C_{r+s}$.

Поэтому следующая последовательность цепочек является выводом в Г:

$\mathcal{D}$
$C_r B_1$
$C_r C_{r+1} B_2$
$\ldots$
$C_r C_{r+1} \ldots C_{r+s-2} B_{s-1}$
$C_r C_{r+1} \ldots C_{r+s-2} C_{r+s-1} C_{r+s}$.

Мы показали, следовательно, что $\mathcal{D} \vdash C_r \ldots C_{r+s}(Г)$.
Теорема доказана.

Заметим, что из теорем 4.I и 4.2 вытекает

Следствие. Для каждой категориальной грамматики можно эффективно построить эквивалентную ей /т.е. определяющую тот же язык/ лм-грамматику сложности ≤ 2.

§ 28. Категориальные грамматики и системы составляющих

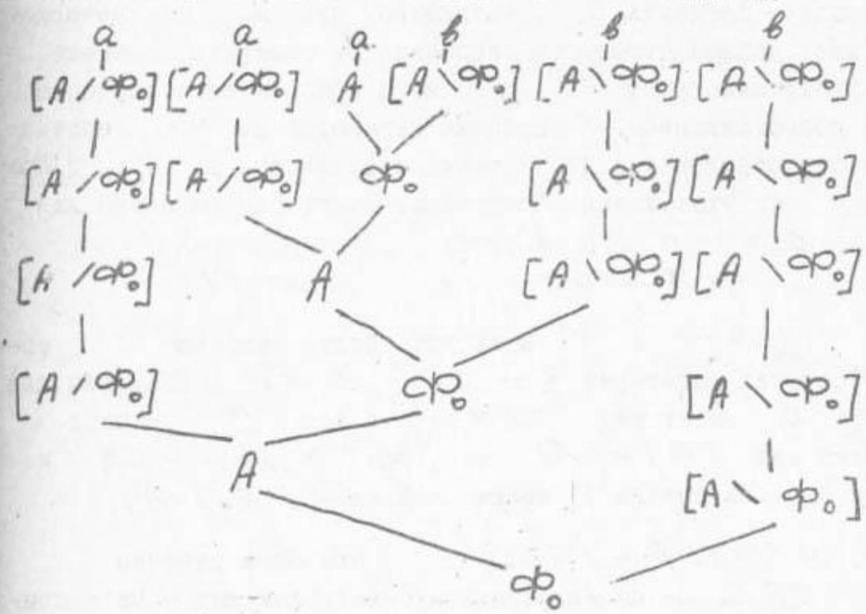
Из теоремы 4.I непосредственно следует, что категориальная грамматика позволяет естественным образом построить для каждой цепочки определяемого ею языка систему составляющих. Именно, достаточно построить по данной категориальной грамматике кс-грамматику с помощью конструкции, примененной в доказательстве теоремы, а затем по выводу цепочки в этой кс-грамматике построить систему составляющих, как описано в § II.

При фактическом построении по категориальной грамматике системы составляющих для цепочки X переходить к кс-грамматике не нужно. Следует просто сократить сопоставленную цепочке X цепочку категорий до φ_0 и взять в качестве составляющих те вхождения подцепочек в X, для которых соответствующие цепочки категорий будут на каких-нибудь шагах процесса сокращения превращаться в одноэлементные цепочки категорий. При этом каждой составляющей, для которой соответствующая цепочка категорий на каком-то шаге переходит в одну категорию Ψ, естественно приписать тип Ψ. /Заметим, что таким образом тип приписывается всем составляющим, включая атомные/. Тип Ψ может быть приписан составляющей только тогда, когда Ψ есть та самая категория, которая приписывается грамматикой /в смысле § 24/ этой составляющей /рассматриваемой как цепочке/.

В силу леммы 4.3 каждый процесс сокращения цепочки категорий до φ_0 сопоставляет цепочке над основным словарем только одну систему составляющих. Тем не менее категориальная грамматика может сопоставлять одной и той же цепочке разные системы составляющих - во-первых, за счет того, что одна и та же цепочка категорий может сокращаться до φ_0 разными способами /см. ниже пример 2/, и, во-вторых, за счет того, что одной цепочке над основным словарем могут сопоставляться разные цепочки категорий /см. ниже пример 4/.

Процесс сокращения цепочки категорий и возникновение связанной с ней системы составляющих можно изобразить графической схемой,

аналогичной схемам главы Ш /чертежи I2 и I3/. На черт. I5 приведена такая схема для сокращения цепочки категорий, соответствующей цепочке $a^3 b^3$, в грамматике G_1 примера 2 из § 25. Неатомными составляющими здесь будут: $(a^3b^3, 1)$, $(a^3b^2, 1)$, $(a^2b^2, 2)$, $(a^2b, 2)$, $(ab, 3)$. Составляющие, соответствующие цепочкам вида $a^i b^i$, имеют тип φ_0, соответствующие цепочкам вида $a^{i+1} b^i$ - тип A.



Черт. I5.

Рассмотрим еще несколько примеров.

Пример I. Возвращаясь к рассуждениям, с помощью которых мы доказывали, что всякая цепочка языка $L(G_3)$ примера 3 из § 25 есть ппф, мы без труда увидим теперь, что этими рассуждениями фактически доказано следующее: во всякой цепочке $x \in L(G_3)$ всякая составляющая типа φ_0 есть ппф, всякая составляющая типа φ_1 есть выражение вида (α), где α - ппф, и всякая составляющая типа $[\Phi_0/\Phi_1]$ есть либо $\neg$, либо выражение вида $(\alpha)\varkappa$, где α – ппф и $\varkappa$ – один из символов $\&$, $\vee$, $\supset$. Совершенно аналогично можно было бы доказать, что всякая составляющая типа $[K\setminus\Phi_1]$ есть выражение вида $\alpha)$, где α – ппф. Кроме того, из тех же рассуждений сразу следует, что неатомные составляющие типов, отличных от четырех перечисленных, невозможны.

Нетрудно, далее, показать, что если $x \in L(G_3)$ и x' – подцепочка цепочки x, являющаяся для x составляющей типа Φ_0 (для полной строгости следовало бы говорить, конечно, о некотором вхождении x' в x), то на некотором шаге процесса сокращения сопоставленной x цепочки категорий до Φ_0 цепочка категорий, сопоставленная x', также сокращается до Φ_0. При $l(x')=1$ это утверждение очевидно. Пусть оно доказано для случая $l(x')<n$, и пусть

$$x = a_1 \ldots a_p \in L(G_3) \quad (a_1,\ldots,a_p \in V),$$

$x' = a_{r+1} \ldots a_{r+n}$ и x' есть ппф. Пусть цепочке x сопоставлена цепочка категорий $\xi = \Omega_1 \ldots \Omega_p$. Возможны два случая: I) x' имеет вид $\neg(\alpha)$, где α – ппф; 2) x' имеет вид $(\alpha)\varkappa(\beta)$, где α, β – ппф и $\varkappa = \&, \vee, \supset$. В случае I) имеем $\Omega_{r+1} = [\Phi_0/\Phi_1]$,

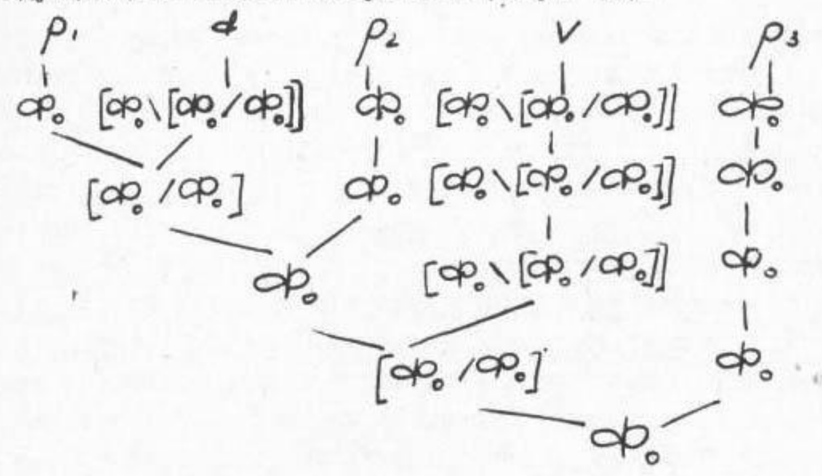
$\Omega_{r+2} = K$, $\Omega_{r+n} = [\Phi_0\setminus[K\setminus\Phi_1]]$. При этом цепочка $\Omega_{r+3}\ldots\Omega_{r+n-1}$ – по индуктивному предположению – на каком-то шаге процесса сокращения ξ сокращается до Φ_0. Следовательно, после этого шага "потомок" цепочки $\Omega_{r+1}\ldots\Omega_{r+n}$ будет иметь вид $\Omega_{r+1}\Omega_{r+2}\Phi_0\Omega_{r+n}$. $\Omega_{r+n}=[\Phi_0\setminus[K\setminus\Phi_1]]$ не может взаимодействовать с категорией, стоящей справа от неё, и поэтому должна взаимодействовать с выделенным вхождением Φ_0. После этого остается выражение $\Omega_{r+1}\Omega_{r+2}[K\setminus\Phi_1] =$

$= [\Phi_0/\Phi_1]K[K\setminus\Phi_1]$, которое, очевидно, не сможет взаимодействовать ни с какими категориями, стоящими слева или справа от него, пока не сократится само до Φ_0. Итак, на каком-то шаге процесса сокращения ξ отрезок $\Omega_{r+1}\ldots\Omega_{r+n}$ сокращается до Φ_0. В случае б) рассуждение совершенно аналогично; его проведение предоставляется читателю.

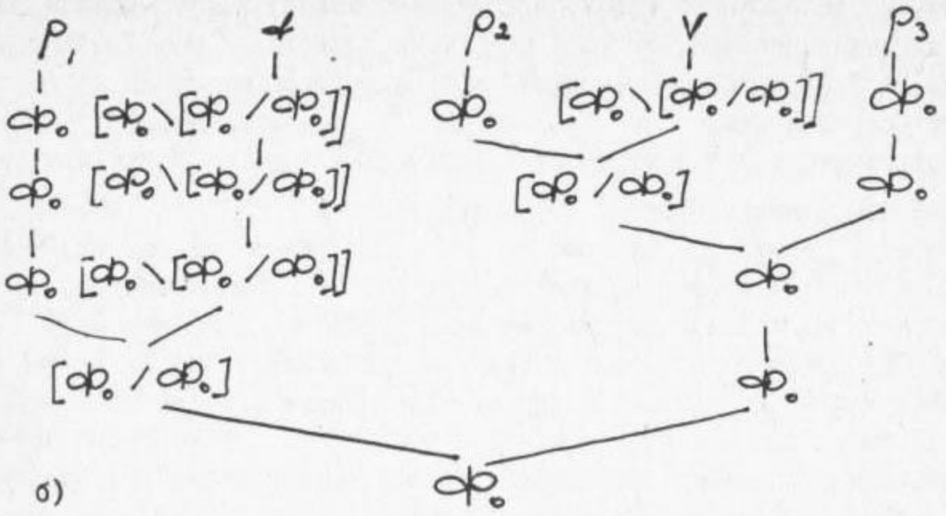
Из доказанного уже непосредственно следует, что всякая подцепочка цепочки $x \in L(G_3)$, имеющая вид (α), $(\alpha)\varkappa$ или $\alpha)$ /где α – всюду означает ппф и $\varkappa = \&, \vee, \supset$/, есть составляющая типа Φ_1, $[\Phi_0/\Phi_1]$ или $[K\setminus\Phi_1]$ соответственно. В самом деле, пусть $x = a_1 \ldots a_p$, соответствующая цепочка категорий пусть будет $\xi = \Omega_1 \ldots \Omega_p$, и пусть, например, $x' = a_r a_{r+1} \ldots a_{r+s}$ имеет вид (α), где α – ппф. Тогда отрезок $\Omega_r \ldots \Omega_{r+s}$ на каком-то шаге сокращения ξ до Φ_0 перейдет в $K\Phi_0[\Phi_0\setminus[K\setminus\Phi_1]]$, а этот последний отрезок не может взаимодействовать ни с чем слева или справа от него, пока не сократится до Φ_1.

Итак, для любой цепочки $x \in L(G_3)$ неатомными составляющими являются /при любом способе сокращения/, все её подцепочки вида α, (α), $(\alpha)\varkappa$ и $\alpha)$, и только они. Следовательно, грамматика G_3 сопоставляет каждой цепочке $x \in L(G_3)$ единственную систему составляющих.

Пример 2. Рассмотрим грамматику G_4 примера 4 из § 25. Для этой грамматики единственность системы составляющих не имеет места. Например, для квазиформулы $p_1 \& p_2 \vee p_3$ возможны следующие два способа сокращения /черт. I6/:



а)



б)

Черт. I6.

Первый способ дает систему составляющих $((((p_1 \&)p_2)\vee)p_3)$, второй – $((p_1 \&)((p_2 \vee)p_3))$. Содержательный смысл этого ясен: квазиформула $p_1 \& p_2 \vee p_3$ омонимична, её можно понимать либо как $(p_1 \& p_2) \vee p_3$, либо как $p_1 \& (p_2 \vee p_3)$.

В противоположность этому при способе записи формул, рассмотренном в предыдущем примере, омонимия отсутствует; именно поэтому разложение на составляющие там однозначно.

Пример 3. Грамматика G_6 примера 6 из § 25 сопоставляет, как легко проверить, предложению X = народ узнал колокольчик Пугачева и толпою бежал за нами следующие /неатомные/ составляющие: X , узнал колокольчик Пугачева и толпою бежал за нами, узнал колокольчик Пугачева и, узнал колокольчик Пугачева, колокольчик Пугачева, толпою бежал за нами, бежал за нами, за нами. Эта система не совпадает с той, которая сопоставляется тому же предложению грамматикой примера I из § I3. Система составляющих из § I3 не бинарна, а построенная сейчас – бинарна. /Очевидно, всякая система составляющих, строящаяся описанным нами способом по категориальной грамматике, является бинарной.ж)

ж) Можно было бы,впрочем,обобщить понятие категориальной грамматики так,чтобы в качестве категорий допускались и многоместные операторы; тогда соответствующие системы составляющих могли бы и не быть бинарными.

Пример 4. Пусть основной словарь некоторой категориальной грамматики состоит из русских словоформ и её приписывающая функция f такова, что множество f (дополняет) содержит категорию V , а каждое из множеств f (отчет) и f (сообщение) содержит категории $[\Phi_0/V^*]$, $[V^* \backslash \Phi_0]$, $[V \backslash V^*]$ и $[V^*/V]$.

Содержательный смысл категорий: Φ_0 – предложение; V^* – группа глагола /сказуемого/ в 3-м лице единственного числа настоящего времени; V – переходный глагол в 3-м лице единственного числа настоящего времени; $[\Phi_0/V^*]$ – подлежащее, стоящее перед сказуемым, т.е. оператор, действующий на группу сказуемого слева и превращающий её в предложение; $[V^* \backslash \Phi_0]$ – подлежащее, стоящее после сказуемого, т.е. аналогичный оператор, действующий справа; $[V \backslash V^*]$ – постпозитивное прямое дополнение, т.е. оператор , действующий справа на переходный глагол и превращающий его в группу сказуемого; $[V^*/V]$ – препозитивное прямое дополнение,т.е. аналогичный оператор, действующий слева.

Цепочке отчет дополняет сообщение такая грамматика сопоставит по меньшей мере две цепочки категорий, сокращающиеся до Φ_0 – $[\Phi_0/V^*]\,V[V \backslash V^*]$ и $[V^*/V]\,V[V^* \backslash \Phi_0]$.

Первая из этих цепочек категорий даст систему составляющих (отчет (дополняет сообщение)), вторая – ((отчет дополняет) сообщение).

Здесь, как и в примере 2, мы имеем дело с омонимией, но другого типа. В примере 2 элементарные символы, из которых состоит цепочка, неомонимичны – каждому из них сопоставлена только одна категория; омонимичен лишь способ упорядочения символов, т.е. омонимия появляется только на уровне предложения. Такую омонимию естественно назвать синтаксической. В настоящем же примере омонимия на уровне предложения возникает за счет омонимичности самих элементарных символов /слов/ – некоторым из них сопоставляется более одной категории. Такую омонимию естественно назвать категориальной.

Система составляющих, строящаяся с помощью категориальной грамматики, обладает одним важным достоинством – её можно естественным образом разметить. Именно, пусть цепочке x над основным словарем сопоставлена цепочка категорий ξ , способ сокращения ξ до φ_0 фиксирован, и пусть на некотором шаге данного процесса сокращения взаимодействуют категории ψ и θ , превращаясь в одну категорию φ . Это означает, что составляющие, соответствующие "предкам" ψ и θ , непосредственно вложены в составляющую, соответствующую "предку" φ . Одна из категорий ψ, θ при их взаимодействии играет роль оператора, а другая - роль аргумента. Составляющую, соответствующую категории, играющей роль аргумента, естественно при этом считать главной.

Например, если S - категория "существительное",то категорию $[S/S]$ естественно интерпретировать как "прилагательное" /ср.пример на стр. 111 /; когда на каком-то шаге процесса сокращения пара категорий $[S/S]\ S$ превращается в S , то естественно при этом считать, что существительное является главной составляющей, а прилагательное - не главной.

В § 4 мы видели, что по размеченной системе составляющих единственным образом строится дерево подчинения. Значит, категориальная грамматика позволяет естественным образом сопоставить каждой цепочке определяемого ею языка не только систему составляющих, но и дерево подчинения; в этом отношении категориальные грамматики обладают преимуществом перед нс-грамматиками.

Например, грамматика G_6 сопоставляет цепочке народ узнал колокольчик Пугачева и толпою бежал за нами следующее дерево подчинения /черт. I7/:

народ узнал колокольчик Пугачева и толпою бежал за нами

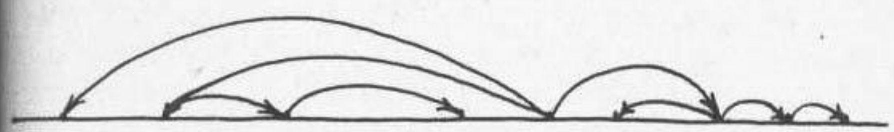
Черт. I7.

Несколько искусственный характер полученного дерева обусловлен в основном тем, что для предложений, содержащих сочинительные союзы, отношение подчинения само по себе не является достаточно адекватным средством представления синтаксической

структуры. Впрочем, более или менее приемлемое описание сочинительной конструкции с помощью отношения подчинения можно получить, если считать вершиной такой конструкции сочинительный союз. При построении отношения подчинения с помощью категориальной грамматики для этого нужно, в отступление от сформулированного выше правила, всегда считать главными составляющие, соответствующие категориям типа "сочинительный союз" и частям таких категорий. При этом понятие категории типа "сочинительный союз" может быть вполне строго определено: так естественно называть всякую категорию, имеющую вид $[\varphi \backslash [\varphi/\varphi]]$ или $[[\varphi \backslash \varphi]/\varphi]$.

Действительно, всякая такая категория есть двуместный оператор, который, будучи поставлен между двумя выражениями типа φ, дает одно выражение того же типа.

Применяя описанное видоизмененное правило построения отношения подчинения, мы получим для рассмотренной выше цепочки следующее дерево /черт. I7'/:



народ узнал колокольчик Пугачева и толпою бежал за нами

Черт. I7'.

Читателю рекомендуется самостоятельно построить деревья подчинения для цепочек, рассмотренных в примерах 2 и 4 настоящего параграфа. Для цепочки примера 2 можно воспользоваться как первоначальным правилом, так и видоизмененным, учитывающим сочинительные союзы /видоизмененное правило дает и в этом случае более естественные деревья/. Следует обратить внимание на то, что в примере 2 разные системы составляющих дают разные деревья(это верно при обоих способах построения последних), в то время как в примере 4 разные системы составляющих дают одно и то же дерево.

ГЛАВА У

НЕКОТОРЫЕ АЛГОРИТМИЧЕСКИЕ ВОПРОСЫ ТЕОРИИ ПОРОЖДАЮЩИХ ГРАММАТИК.

§ 29. Алгоритмическая природа языков, порождаемых произвольными грамматиками.

Задача настоящего параграфа - выяснить, каков об"ем класса языков, порождаемых произвольными грамматиками. Решение этой задачи естественно получается на языке теории алгоритмов.

Во всех вопросах, относящихся к алгоритмам, рекурсивным функциям, машинам Тьюринга и т.п., мы будем пользоваться терминологией и обозначениями книги А.И.Мальцева [I9], но с одним исключением - вместо употребляемых в [I9] терминов "алфавит" и "слово" будут, как всюду в настоящих лекциях, использоваться соответственно термины "словарь" и "цепочка". (В частности, мы будем говорить "внешний словарь машины Тьюринга", "машинная цепочка" и т.п.) . Кроме того, наряду с термином "словарное множество" мы будем пользоваться нашим обычным термином "язык", означающим то же самое. Поэтому ясен смысл таких словосочетаний, как"рекурсивно перечислимый язык", " примитивно рекурсивный язык" и т.п.

Установим прежде всего связь между машинами Тьюринга и грамматиками. Эта связь выражается следующей теоремой.

Теорема 5.I. Для каждой машины Тьюринга T, правильно вычисляющей одноместную частичную словарную функцию F, заданную в словаре $V = \{a_1, \dots, a_n\}$, может быть эффективно построена грамматика Γ такая, что $L(\Gamma)$ совпадает с множеством значений F.

Доказательство. Пусть внешний словарь машины T есть $\{a_0, a_1, \dots, a_n\}$ и функция F задана в редуцированном словаре $V = \{a_1, \dots, a_n\}$. Введем новые символы I_p, A, v. Пусть $q_0, q_1, \dots, q_s$ - внутренние состояния машины T (q_1 - начальное состояние, q_0 - заключительное). Каждой команде K машины T мы сопоставим систему правил $\Pi(K)$ следующим образом:

1/ Если $K = q_\alpha a_i \to q_\beta a_j$, то $\Pi(K)$ состоит из одного правила $q_\alpha a_i \to q_\beta a_j$.

2/ Если $K = q_\alpha a_i \to q_\beta L$, то $\Pi(K) =$

$$= \{ a_k q_\alpha a_i \to q_\beta a_k a_i \mid k = 0, 1, \dots n \}.$$

3/ Если $K = q_\alpha a_i \to q_\beta R$, то $\Pi(K) = \Pi_1(K) \cup \Pi_2(K)$, где $\Pi_1(K) = \{q_\alpha a_i a_k \to a_i q_\beta a_k \mid k = 0, 1, \dots, n\}$,

$$\Pi_2(K) = \{q_\alpha a_i v \to a_i q_\beta a_0 v\}.$$

Об"единение систем $\Pi(K)$ по всем командам машины T обозначим через S_0. Определим еще две системы правил:

$$S_1: \begin{cases} I_p \to A v \\ A \to A a_i, \quad i = 1, \dots n \\ A \to q_1 a_0 \end{cases}$$

$$S_2: \begin{cases} q_0 a_0 \to q_0 \\ q_0 a_i \to a_i q_0, \quad i = 1, \dots n \\ q_0 v \to \Lambda \end{cases}$$

Положим $V_1 = \{a_0, q_0, q_1, \dots, q_s, I_p, A, v\}$, и $S = S_0 \cup S_1 \cup S_2$. Грамматику $\langle V, V_1, I_p, S \rangle$ обозначим через Γ.

Покажем, что Γ и есть нужная грамматика.

Любое значение функции F выводится в Γ из I_p. Действительно, с помощью правил системы S_1 из I_p выводится любая цепочка вида $q_1 a_0 x v$, где x -произвольная цепочка над V. Далее, очевидно, что если μ и ν - машинные цепочки такие, что ν получается из μ применением некоторой команды K, то цепочка νv получается из μv применением одного из правил системы $\Pi(K)$. Поэтому для любой цепочки x, для которой $F(x)$ определена, из цепочки $q_1 a_0 x v$ выводима по правилам системы S_0 цепочка $q_0 a_0 x F(x) a_0^t v$, где t - некоторое неотрицательное целое число. А из этой цепочки, в свою очередь, по правилам системы S_2 можно вывести $F(x)$.

Покажем теперь, что всякая цепочка над V, выводимая в Γ из Π_p, является значением функции F. Ясно, прежде всего, что каждый вывод из Π_p должен начинаться применением правил системы S_1, причем сначала должно быть применено правило первой строки, затем сколько-то раз правила второй строки и, наконец, правило третьей строки; в результате получится цепочка вида $q_1 a_0 x v$, где x - цепочка над V. После этого должны применяться правила системы S_0; при этом ясно, что на каждом шаге к соответствующей промежуточной цепочке применимо только одно правило из S_0, отличное от $v \to a_0 v$. Индукцией по числу шагов легко доказать, что каждая цепочка, выводимая в Γ из $q_1 a_0 x v$, имеет вид $(q_1 a_0 x)^{(t)} a_0^k v$ (обозначение $y^{(t)}$ введено в [19], стр. 240), где $t = 1, 2, \ldots$ и $k = 0, 1, \ldots$. Далее, если окончательная цепочка вывода содержит только символы $a_1, \ldots, a_n$, то в выводе обязательно должны применяться правила системы S_2. Но применение этих правил возможно лишь после того, как получится цепочка, содержащая q_0; а такая цепочка может быть выведена из $q_1 a_0 x v$ только в том случае, когда она имеет вид $q_0 a_0 F(x) a_0^t v$ (в силу того, что Γ правильно вычисляет F). Наконец, из вида системы S_2 непосредственно ясно, что по правилам этой системы из цепочки $q_0 a_0 F(x) a_0^t v$ может быть выведена только одна цепочка над V, а именно $F(x)$ - действительно, символ q_0, прежде чем исчезнуть, должен пройти через всю цепочку до v, а при этом он уничтожит все вхождения a_0. Доказательство закончено.

Обратим внимание читателя на сходство конструкции, использованной в доказательстве теоремы 5.I, с конструкцией, доказывающей неразрешимость проблемы равенства слов в полугруппах([19], п.I3.I

В силу теоремы I пункта 4.2 книги [I9] всякий непустой рекурсивно перечислимый язык является множеством значений некоторой примитивно рекурсивной функции ж), а в силу теоремы 2 пункта I2.2 той же книги всякая частично рекурсивная функция - и тем более всякая примитивно рекурсивная функция - правильно вычислима на машине Тьюринга. Кроме того, пустой язык, очевидно, порождается грамматикой. Поэтому из теоремы 5.I вытекает

<u>Теорема 5.2.</u> Всякий рекурсивно перечислимый язык порождается некоторой грамматикой.

Имеет место и обратное утверждение:

<u>Теорема 5.3.</u> Всякий язык, порождаемый грамматикой, рекурсивно перечислим.

<u>Доказательство</u>. Мы воспользуемся теоремой о порожденных совокупностях из п.4.3. книги [I9]. Эта теорема, доказанная там для числовых множеств, верна, разумеется, и для языков.

Пусть $\Gamma = \langle V, V_1, \Pi_p, S \rangle$ - произвольная грамматика. Чтобы доказать рекурсивную перечислимость языка $L(\Gamma)$, достаточно установить, что рекурсивно перечислимо множество всех цепочек над $V \cup V_1$, выводимых в Γ из Π_p. (мы будем обозначать это множество $\Sigma(\Gamma)$). Действительно, $L(\Gamma) = \Sigma(\Gamma) \cap F(V)$; $F(V)$ (множество всех цепочек над V) очевидно, рекурсивно и тем более рекурсивно перечислимо, и пересечение двух рекурсивно перечислимых множеств рекурсивно перечислимо([I9], п.4.2, теорема 2).

Теперь нам нужно представить $\Sigma(\Gamma)$ как множество, порожденное рекурсивно перечислимой совокупностью W с помощью конечной системы примитивно рекурсивных словарных функций. В качестве W мы возьмем множество, состоящее из одного элемента Π_p. В качестве функций, с помощью которых происходит порождение, естественно взять операции подстановки в цепочку вместо вхождений

ж) Эта теорема доказана в [I9] для числовых множеств, но из определений словарного множества и словарной функции (там же, п.II.I) следует, что она верна и для словарных множеств (языков).

левых частей правил Γ вхождений правых частей тех же правил. Действительно, $\Sigma(\Gamma)$ есть в точности наименьшее множество, содержащее $\Pi p.$ и замкнутое относительно этих операций. Однако указанные операции многозначны, поскольку цепочка может содержать несколько вхождений левой части одного и того же правила; чтобы иметь возможность воспользоваться теоремой о порожденных совокупностях, нам нужно перейти к однозначным операциям. Это мы сделаем следующим образом. Сопоставим каждому символу $\alpha \in V \cup V_1$ новый символ $\bar{\alpha}$ и два новых правила $\alpha \to \bar{\alpha}$ и $\bar{\alpha} \to \alpha$. Грамматику, полученную из Γ присоединением к вспомогательному словарю всех новых символов и к схеме - всех новых правил, обозначим $\bar{\Gamma}$. Вывод в грамматике $\bar{\Gamma}$ назовем <u>левым</u>, если на каждом его шаге заменяется первое (т.е. самое левое) вхождение левой части применяемого на этом шаге правила. Множество всех цепочек, выводимых в $\bar{\Gamma}$ из $\Pi p.$ с помощью левых выводов, обозначим $\Sigma_\Lambda(\bar{\Gamma})$. Пусть теперь $\varphi_1 \to \psi_1, \ldots, \varphi_s \to \psi_s$ — все правила $\bar{\Gamma}$. Положим для произвольной цепочки ξ $F_i(\xi) = Sb(\xi, \varphi_i, \psi_i)$, где Sb - функция, определенная в п. 12.2 книги [19]. Ясно, что $\Sigma_\Lambda(\bar{\Gamma})$ есть множество, порожденное совокупностью $W = \{\Pi p\}$ с помощью системы функций $F_1, \ldots, F_s$. А поскольку функция Sb примитивно рекурсивна, все F_i также примитивно рекурсивны; стало быть, множество $\Sigma_\Lambda(\bar{\Gamma})$ рекурсивно перечислимо.

Покажем теперь, что

$$\Sigma(\Gamma) = \Sigma_\Lambda(\bar{\Gamma}) \cap F(V \cup V_1)$$

. Поскольку $F(V \cup V_1)$ рекурсивно перечислимо, это завершит доказательство.

Пусть $\xi \models \eta(\Gamma)$, т.е. $\xi = \omega\varphi\chi$, $\eta = \omega\psi\chi$, где $\varphi \to \psi$ - правило Γ, и пусть $\omega = \alpha_1 \ldots \alpha_k$ $(\alpha_1, \ldots, \alpha_k \in V \cup V_1)$. Тогда цепочка η может быть получена из ξ с помощью следующего левого вывода в $\bar{\Gamma}$:

$$(\xi = \alpha_1\alpha_2 \ldots \alpha_k\varphi\chi,\ \bar{\alpha}_1\alpha_2 \ldots \alpha_k\varphi\chi, \ldots, \bar{\alpha}_1\bar{\alpha}_2 \ldots \bar{\alpha}_k\varphi\chi,\ \bar{\alpha}_1\bar{\alpha}_2 \ldots \bar{\alpha}_k\psi\chi,\ \alpha_1\bar{\alpha}_2 \ldots \bar{\alpha}_k\psi\chi, \ldots, \alpha_1\alpha_2 \ldots \alpha_k\psi\chi).$$

Поэтому всякий вывод в Γ может быть преобразован в левый вывод в $\bar{\Gamma}$; отсюда $\Sigma(\Gamma) \subseteq \Sigma_\Lambda(\bar{\Gamma})$; но $\Sigma(\Gamma) \subseteq F(V \cup V_1)$, так что $\Sigma(\Gamma) \subseteq \Sigma_\Lambda(\bar{\Gamma}) \cap F(V \cup V_1)$.

С другой стороны, пусть $\alpha = (\Pi p = \omega_0, \omega_1, \ldots, \omega_n)$ -произвольный вывод в $\bar{\Gamma}$, заканчивающийся цепочкой над $V \cup V_1$, и пусть в этом выводе первое применение правила, не входящего в схему Γ, происходит на i-м шаге. Тогда i-й шаг состоит в надчеркивании некоторого символа, и этот символ не может участвовать ни в каких дальнейших преобразованиях до тех пор, пока на некотором шаге с номером j $(j > i)$ не потеряет черточку. Поэтому i-й и j-й шаги можно опустить. Так можно устранить все шаги, на которых применяются правила вида $\alpha \to \bar{\alpha}$ и $\bar{\alpha} \to \alpha$, и получить вывод в Γ из $\Pi p.$ с той же последней цепочкой. Отсюда $\Sigma(\bar{\Gamma}) \cap F(V \cup V_1) \subseteq \Sigma(\Gamma)$ и тем более $\Sigma_\Lambda(\bar{\Gamma}) \cap F(V \cup V_1) \subseteq \Sigma(\Gamma)$. Доказательство закончено.

Об"единяя теоремы 5.2 и 5.3, получаем следующее основное утверждение:

<u>Теорема 5.4</u>. Класс языков, порождаемых грамматиками, совпадает с классом рекурсивно перечислимых языков.

<u>Замечание</u>. Поскольку существуют рекурсивно перечислимые множества, дополнения которых не являются рекурсивно перечислимыми ([19], стр. 87), из теоремы 5.4 следует, что класс языков, порождаемых грамматиками, не замкнут относительно операции C (см. замечание I к теореме 3.II).

§ 30. Алгоритмическая природа нс-языков.

Из теоремы 5.4 можно сделать вывод, что класс естественных языков (точнее, класс множеств грамматически правильных фраз естественных языков) является весьма узким подклассом класса языков, порождаемых грамматиками. В самом деле, интуитивно представляется ясным, что для каждого естественного языка должен существовать алгоритм, позволяющий носителям языка распознавать по произвольной фразе, является ли она грамматически правильной. Иначе говоря, множество грамматически правильных фраз естественного языка должно быть рекурсивным. Но рекурсивные множества составляют собственный подкласс – и притом довольно узкий – класса всех рекурсивно перечислимых множеств.

Сверх сказанного, алгоритмы распознавания грамматической правильности должны быть более или менее простыми в том смысле, что они должны давать ответ в обозримые и даже коротк..е промежутки времени. Поэтому следует ожидать, что в классификациях рекурсивных множеств, основанных на сложности распознающих алгоритмов (о таких классификациях см., например, [20]), множества грамматически правильных фраз естественных языков будут занимать места где-то на низших ступенях иерархии.

С этой точки зрения интересно исследовать языки, порождаемые нс- грамматиками (нс-языки), поскольку нс-грамматики представляются удобными для описания естественных языков из других соображений, а именно потому, что они позволяют естественным образом приписывать порождаемым предложениям системы составляющих (см.§ II).

Прежде всего, нетрудно видеть, что для всякой нс-грамматики и даже для всякой неукорачивающей грамматики существует алгоритм, позволяющий по произвольной цепочке узнавать, выводима ли она в этой грамматике из начального символа. В самом деле, пусть $\Gamma = \langle V, V_1, \text{Пр}, S \rangle$ - неукорачивающая грамматика и ω - цепочка над $V \cup V_1$, выводимая в Γ из Пр. Тогда, очевидно, существует такой вывод в Γ $\mathfrak{A} = (\text{Пр} = \omega_0, \omega_1, \ldots \omega_n = \omega)$, в котором никакая цепочка не встречается дважды. Поскольку грамматика неукорачивающая, для каждого $i = 1, 2, \ldots n$ $l(\omega_i) \leq l(\omega)$; поэтому длина вывода $\mathfrak{A}$ не превосходит числа $C(\omega)$ всевозможных различных цепочек над $V \cup V_1$, длины которых меньше или равны $l(\omega)$. $C(\omega)$ легко может быть вычислено. Поэтому мы имеем следующий алгоритм: чтобы узнать, выводима ли цепочка ω из Пр в Γ, нужно вычислить $C(\omega)$ и выписать всевозможные выводы из Пр, длины которых не превосходят $C(\omega)$. Если среди этих выводов хотя бы один оканчивается цепочкой ω, и только в этом случае, ω выводима из Пр в Γ.

Для дальнейшего заметим, что построенный алгоритм не зависит от грамматики. В сущности, мы получили алгоритм, позволяющий для произвольной неукорачивающей грамматики Γ и произвольной цепочки узнать, выводима ли эта цепочка в грамматике Γ из ее начального символа.

Итак, мы доказали, что все языки, порождаемые неукорачивающими грамматиками - т.е. все нс-языки-рекурсивны. Этот результат можно усилить. Именно, более тонкий анализ только что проведенного рассуждения дает следующую теорему:

<u>Теорема 5.5.</u> Всякий нс-язык примитивно рекурсивен.

Для доказательства этой теоремы нам понадобятся некоторые новые понятия и леммы.

Пусть $\Gamma = \langle V, V_1, \text{Пр}, S \rangle$ - некоторая грамматика и ω - цепочка над $V \cup V_1$, выводимая в Γ из Пр (Сам символ Пр будем считать тоже выводимым из Пр). Обозначим через $\tau_{\Gamma,\omega}$ наименьшую возможную длину вывода цепочки ω из Пр в Γ. Через $\tau_\Gamma(n)$ будем обозначать максимум чисел $\tau_{\Gamma,\omega}$ по всем цепочкам ω длины $\leq n$, выводимым из Пр в Γ. Функция $\tau_\Gamma(n)$ называется <u>временной сигнализирующей функцией</u> грамматики Γ. Эта функция может рассматриваться как некоторая мера сложности грамматики.

Оказывается полезной и другая мера сложности грамматики. Именно, пусть $\mathfrak{A} = (\text{Пр} = \omega_0, \omega_1, \ldots, \omega_n)$ -вывод в Γ. Обозначим через $\sigma_{\Gamma,\mathfrak{A}}$ наибольшую из длин цепочек $\omega_0, \ldots, \omega_n$. Для произвольной цепочки ω, выводимой в Γ из Пр, будем обозначать через $\sigma_{\Gamma,\omega}$ минимум чисел $\sigma_{\Gamma,\mathfrak{A}}$ по всем выводам цепочки ω из Пр в Γ. Наконец, через $\sigma_\Gamma(n)$ будем обозначать максимум чисел $\sigma_{\Gamma,\omega}$ по всем цепочкам длины $\leq n$, выводимым из Пр в Γ. Функция $\sigma_\Gamma(n)$ называется <u>емкостной сигнализирующей функцией</u> грамматики Γ.

Наконец, нам понадобится еще одна сигнализирующая функция, которую мы будем обозначать $\tau'_\Gamma(n)$. Она определяется точно так же, как $\tau_\Gamma(n)$, но с тем изменением, что вместо всевозможных выводов берутся лишь всевозможные <u>левые</u> выводы (в том самом смысле, в котором этот термин употреблялся в предыдущем параграфе).

Ясно, что $\sigma_\Gamma(n), \tau_\Gamma(n), \tau'_\Gamma(n)$ – всюду определенные неубывающие функции.

Лемма 5.I. Для произвольной грамматики $\Gamma = \langle V, V_1, \Pi_p, S\rangle$ имеет место неравенство $\tau_\Gamma(n) < Q^{\sigma_\Gamma(n)+1}$, где Q – мощность словаря $V \cup V_1$.

Доказательство. В силу определения функции τ_Γ для каждого $n = 1, 2, \ldots$ существует такая цепочка ω_n длины $\le n$, что $\tau_{\Gamma,\omega_n} = \tau_\Gamma(n)$. Существует такой вывод $\mathfrak{A}$ цепочки ω_n из Π_p в Γ, что $\sigma_{\Gamma,\mathfrak{A}} = \sigma_{\Gamma,\omega_n}$. Пусть $\mathfrak{D}$ -кратчайший из выводов $\mathfrak{A}$, удовлетворяющих этому условию. В выводе $\mathfrak{D}$ никакая цепочка не встречается дважды (иначе он не был бы кратчайшим, т.к. из него можно было бы выбросить кусок между повторяющимися цепочками, и максимум длин цепочек при этом не увеличился бы). Поэтому, если t – длина вывода $\mathfrak{D}$, то t не превышает числа всевозможных различных цепочек над $V \cup V_1$, длины которых меньше или равны $\sigma_{\Gamma,\mathfrak{D}} = \sigma_{\Gamma,\omega_n}$. А это число равно

$$Q^{\sigma_{\Gamma,\omega_n}} + Q^{\sigma_{\Gamma,\omega_n}-1} + \dots + Q + 1 =$$

$$= \frac{1}{Q-1}\left(Q^{\sigma_{\Gamma,\omega_n}+1} - 1\right) < Q^{\sigma_{\Gamma,\omega_n}+1} \le Q^{\sigma_\Gamma(n)+1}$$

В то же время $t \ge \tau_{\Gamma,\omega_n} = \tau_\Gamma(n)$
Отсюда $\tau_\Gamma(n) < Q^{\sigma_\Gamma(n)+1}$.

Лемма 5.2. Если емкостная сигнализирующая функция грамматики Γ мажорируется примитивно рекурсивной функцией, то и язык $L(\Gamma)$ примитивно рекурсивен.

Доказательство. Пусть $\Gamma = \langle V, V_1, \Pi_p, S\rangle$. Построим, как в доказательстве теоремы 5.3, грамматику $\bar\Gamma$, для которой имеет место равенство $\Sigma(\Gamma) = \Sigma_\Lambda(\bar\Gamma) \cap F(V \cup V_1)$. Из этого равенства следует, что $L(\Gamma) = \Sigma_\Lambda(\bar\Gamma) \cap F(V)$. В силу примитивной рекурсивности множества $F(V)$ (вытекающей

хотя бы из теоремы I п.II.I книги [19]) и замкнутости класса примитивно рекурсивных множеств относительно пересечения (там же, п.4.1, теорема I) достаточно доказать примитивную рекурсивность множества $\Sigma_\Lambda(\bar\Gamma)$.

Нетрудно видеть, что функция $\tau'_{\bar\Gamma}(n)$ мажорируется примитивно рекурсивной функцией. В самом деле, пусть $\mathfrak{A} = (\Pi_p = \omega_0, \omega_1, \ldots, \omega_p = \omega)$ – произвольный вывод в $\bar\Gamma$, начинающийся символом Π_p и оканчивающийся цепочкой ω. Каждый шаг этого вывода, состоящий в переходе от ω_{i-1} к ω_i, можно, как в предыдущем параграфе, заменить некоторым левым выводом в $\bar\Gamma$, длина которого не превосходит $2\,l(\omega_i)$. Таким образом, вывод $\mathfrak{A}$ можно заменить левым выводом $\mathfrak{D} = (\Pi_p = \omega_0, \omega_1, \ldots, \omega_t = \omega)$, причем $t \le 2 \max\limits_{i=1,\ldots,n} l(\omega_{i-1}) \cdot n$. Но, очевидно, для каждого $i = 1, \ldots, n$ $l(\omega_i) \le i \cdot d$, где d – максимум разностей между длинами правых и левых частей правил $\bar\Gamma$. Поэтому $t \le 2n \cdot d \cdot n = 2dn^2$. Если $\mathfrak{A}$ – кратчайший вывод цепочки ω, то $n = \tau_{\bar\Gamma,\omega}$. В то же время $\tau'_{\bar\Gamma,\omega} \le t$ ($\tau'_{\bar\Gamma,\omega}$ -кратчайший левый вывод цепочки ω), так что $\tau'_{\bar\Gamma,\omega} \le 2d \cdot (\tau_{\bar\Gamma,\omega})^2$.
Если при этом ω – та цепочка длины $\le n$, для которой $\tau'_{\bar\Gamma,\omega} = \tau'_{\bar\Gamma}(n)$ (такая цепочка существует в силу определения $\tau'_{\bar\Gamma}(n)$), то $\tau'_{\bar\Gamma}(n) \le 2 \cdot d \cdot (\tau_{\bar\Gamma,\omega})^2 \le 2 \cdot d \cdot (\tau_{\bar\Gamma}(n))^2 \le$

$\le 2d \cdot Q^{2(\sigma_{\bar\Gamma}(n)+1)}$

(по лемме 5.I). Но из самой конструкции грамматики $\bar\Gamma$ ясно, что $\sigma_{\bar\Gamma}(n) = \sigma_\Gamma(n)$. Итак, $\tau'_{\bar\Gamma}(n) \le 2d \cdot Q^{2(\sigma_\Gamma(n)+1)} \le$

$\le 2d \cdot Q^{2(f(n)+1)}$,

где $f(n)$ – примитивно-рекурсивная функция, мажорирующая $\sigma_\Gamma(n)$.

Введем два новых символа α и β, не входящие ни в основной, ни во вспомогательный словарь грамматики $\bar\Gamma$. Мы будем определять некоторые словарные функции над словарем W, полученным из об"единения основного и вспомогательного словарей грамматики $\bar\Gamma$ присоединением к нему символов α и β.

Нам понадобятся также функции, у которых одни аргументы будут цепочками над W, а другие - натуральными числами; но мы будем отождествлять число n с цепочкой над W, для которой n является алфавитным номером (см. [I9], стр.222). Необходимый для получения алфавитной нумерации пересчет словаря W зададим произвольно.

Пусть $\varphi_1 \to \psi_1, \ldots, \varphi_s \to \psi_s$ — все правила грамматики Γ. Порядок, в котором они здесь записаны, произволен, но в дальнейшем считается фиксированным. Положим

$$G_1(\xi) = \alpha\, Sb(\xi, \varphi_1, \psi_1)\beta \ldots \alpha\, Sb(\xi, \varphi_s, \psi_s)\,\beta \quad *)$$

Очевидно, G_1 примитивно рекурсивна.

Положим, далее, для произвольного символа $\delta \in W$

$$H_\delta(\xi) = \mathcal{D}(\xi, \delta)\,\Delta\,(\xi \dot{-} \delta \cdot Sb_\delta(\xi, \Lambda)).$$

Очевидно, $H_\delta(\xi) = \Lambda$ тогда и только тогда, когда цепочка ξ имеет вид $\xi'\delta$, где ξ' не содержит вхождений δ. Ясно также, что H_δ примитивно рекурсивна.

Положим

$$\mathcal{J}_\delta(\xi) = H_\delta(\xi^\sim).$$

Ясно, что $\mathcal{J}_\delta$ примитивно рекурсивна и $\mathcal{J}_\delta(\xi) = \Lambda$ тогда и только тогда, когда $\xi = \delta\,\xi'$, где ξ' не содержит δ.

Пусть теперь

$$G_2(\xi) = H_\beta(\xi)\,\mathcal{J}_\alpha(\xi \dot{-} \beta).$$

Очевидно, G_2 примитивно рекурсивна и $G_2(\xi) = \Lambda$ тогда и только тогда, когда $\xi = \alpha\,\xi'\beta$, где ξ' не содержит ни α, ни β.

---

*) Определения функций Sb, $\mathcal{D}$, $\dot{-}$, $^\sim$ и т.д. см [I9], п.II.2.

Далее, пусть

$$G_3(\xi, \eta) = G_2(\eta)\,\mathcal{D}(\xi, \eta).$$

Очевидно, $G_3(\xi, \eta) = \Lambda$ тогда и только тогда, когда $\xi = \xi'\eta$ и η имеет вид $\alpha\,\eta'\beta$, где η' не содержит ни α, ни β. Функция G_3 примитивно рекурсивна.

Положим

$$G_4(\xi, \eta) = \begin{cases} \Lambda & \text{, если } \Delta(\xi, \eta) = \Lambda \\ G_3(\xi, \eta) & \text{, если } \Delta(\xi, \eta) \neq \Lambda. \end{cases}$$

Очевидно, $G_4(\xi, \eta) = W_1(\Lambda, G_3(\xi, \eta), \Delta(\xi, \eta))$, так что G_4 примитивно рекурсивна. При этом уравнение $G_4(\xi, \eta) = \Lambda = 0$ (вспомним, что мы отождествили цепочки над W с натуральными числами) для любого ξ имеет хотя бы одно решение (равное ξ).

Пусть теперь

$$G_5(\xi) = \mu_\eta\,(G_4(\xi, \eta) = \Lambda)$$

(μ берется в смысле естественного порядка натуральных чисел).

Очевидно, если ξ имеет вид $\xi = \xi'\alpha\,\eta'\beta$, где η' не содержит ни α, ни β, то $G_5(\xi) = \alpha\,\eta'\beta$; в противном случае $G_5(\xi) = \xi$. Поскольку при $\xi = \xi'\alpha\,\eta'\beta$ имеем $\alpha\,\eta'\beta \leq \xi$, для любого ξ верно $G_5(\xi) \leq \xi$; а отсюда по теореме о мажорируемых неявных функциях ([I9], стр.56) вытекает, что G_5 примитивно рекурсивна.

Положим

$$F(\xi) = ((\xi \dot{-} \beta)^\sim \dot{-} \alpha)^\sim.$$

Очевидно, если $\xi = \alpha\,\eta\,\beta$, то $F(\xi) = \eta$.

Определим теперь функцию $G_6(\xi)$ следующим образом:

$$G_6(\Lambda) = \Lambda$$

$$G_6(\xi\delta) = G_6(\xi), \text{ если } \delta \in W \text{ и } \delta \neq \beta$$

$$G_6(\xi\beta) = G_6(\xi)\, G_1(F(G_5(\xi)))$$

Функция G_6 примитивно рекурсивна.

Очевидно, если $\xi = \alpha\eta_1\beta \dots \alpha\eta_k\beta$, где $\eta_1, \dots, \eta_k$ не содержат ни α, ни β, то $G_6(\xi) = G_1(\eta_1) \dots G_1(\eta_k)$

Далее, определим примитивно рекурсивную функцию $G_7(n)$ так:

$$G_7(0) = \alpha\, \text{Пр.}\, \beta$$

$$G_7(n+1) = G_7(n)\, G_6(G_7(n))$$

Для каждого n цепочка $G_7(n)$ имеет вид $\alpha\xi_1\beta \dots \alpha\xi_t\beta$, где последовательность $\xi_1, \dots, \xi_k$ состоит в точности из всех цепочек, являющихся результатами таких левых выводов из Пр. в $\bar{\Gamma}$, длины которых равны $0, 1, \dots, n$. (В этой последовательности возможны, конечно, повторения).

Определим функцию $G_8(\xi, \eta)$ следующим образом:

$$G_8(\xi, \eta) = E(\xi, \eta)\, G_2(\eta)$$

Очевидно, G_8 примитивно рекурсивна и при $\xi \neq \Lambda$ $G_8(\xi, \eta) = \Lambda$ тогда и только тогда, когда η есть подцепочка ξ, имеющая вид $\alpha\eta'\beta$, где η' не содержит ни α, ни β.

Пусть $g(n)$ - примитивно рекурсивная функция, мажорирующая $\tau'_{\bar{\Gamma}}(n)$. Положим $h(\xi) = g(\ell(\xi))$. Поскольку $\ell(\xi)$ - примитивно рекурсивная функция *), $h(\xi)$ также примитивно рекурсивна.

Определим функцию $G_9(n, \eta)$ так:

$$G_9(n,\eta) = \begin{cases} G_8(G_7(n), \alpha\eta\beta), & \text{если } n \leq h(\eta) \\ \Lambda, & \text{если } n > h(\eta) \end{cases}$$

Примитивная рекурсивность этой функции очевидна.

При $n \leq h(\eta)$ $G_9(n, \eta) = \Lambda$ тогда и только тогда, когда η не содержит ни α, ни β и $\alpha\eta\beta$ есть подцепочка цепочки $G_7(n)$; а это имеет место тогда и только тогда, когда цепочка η является результатом некоторого левого вывода из Пр. в $\bar{\Gamma}$, длина которого не превосходит n. Но, поскольку $\tau'_{\bar{\Gamma}}(\ell(\eta)) \leq h(\eta)$, произвольная цепочка η принадлежит $\tilde{L}_\Lambda(\bar{\Gamma})$ тогда и только тогда, когда η является результатом какого-либо левого вывода из Пр. в Γ, длина которого не превосходит $h(\eta)$. Следовательно, $\eta \in \tilde{L}_\Lambda(\bar{\Gamma})$ тогда и только тогда, когда наименьшее n, для которого $G_9(n, h) = \Lambda = 0$, не превосходит $h(\eta)$. Иначе говоря, функция

$$G_{10}(\eta) = \overline{sg}((h(\eta)+1) \dot{-} \mu_n(G_9(n, \eta) = 0))$$

является характеристической функцией множества $\tilde{L}_\Lambda(\bar{\Gamma})$.

А эта функция в силу теоремы о мажорируемых неявных функциях примитивно рекурсивна. (В самом деле, уравнение $G_9(n, \eta) = 0$ для любого η имеет хотя бы одно решение, не превосходящее $h(\eta) + 1$.)

Лемма доказана.

Из доказанной леммы немедленно вытекает теорема 5.5, поскольку для любой нс-грамматики и вообще для любой неукорачивающей грамматики Γ имеем, очевидно, $\sigma_\Gamma(n) \leq n$.

*) Она определяется схемой: $\ell(\Lambda) = 0$

$$\ell(\xi\delta) = \ell(\xi) + 1 \quad (\delta \in W).$$

Обращение теоремы 5.5 неверно– не всякий примитивно рекурсивный язык является нс-языком. Мы покажем это с помощью примера.

Обозначим через $\mathcal{O}'$ множество всевозможных неукорачивающих грамматик Γ', удовлетворяющих следующим условиям: 1/ основной словарь каждой грамматики Γ' есть $V=\{a,b,c,d\}$. 2/ вспомогательный словарь каждой грамматики Γ' содержится в счетном множестве $Z'=\{C_1,C_2,\ldots\}$. 3/ начальный символ каждой грамматики Γ' есть C_1. 4/ схема каждой грамматики Γ' содержит правила $C_2\to a$, $C_3\to b$, $C_4\to c$, $C_5\to d$ (эти правила мы будем называть особыми), а все остальные правила Γ' не содержат вхождений основных символов.

Для каждой грамматики $\Gamma'\in\mathcal{O}'$ построим грамматику Γ такую, что $\widetilde{L}_\lambda(\Gamma)\cap F(V)=L(\Gamma')$. При этом воспользуемся способом, несколько отличным от применявшегося ранее. Именно, новые символы и новые правила будем сопоставлять теперь только вспомогательным символом грамматики Γ'. (Таким образом, каждому символу $C_i\in Z'$ будет сопоставлен новый символ $\overline{C_i}$). Равенство $\widetilde{L}_\lambda(\Gamma)\cap F(V)=L(\Gamma')$ будет доказываться так же, как в доказательстве теоремы 5.3 устанавливалось равенство $\widetilde{L}_\lambda(\Gamma)\cap F(V\cup\overline{V}_1)=\widetilde{L}(\Gamma)$ — нужно только заметить, что основные символы встречаются лишь в правых частях особых правил, которые можно применять в самом конце вывода и притом слева направо.

Множество всех полученных таким образом грамматик мы обозначим $\mathcal{O}$. Вспомогательные словари всех грамматик из $\mathcal{O}$ содержатся в множестве $Z=\{A_1,A_2,\ldots\}$, где A_{2i-1} означает C_i и A_{2i} — $\overline{C_i}$.

Ясно, что любой нс-язык над словарем $V=\{a,b,c,d\}$ порождается некоторой грамматикой из $\mathcal{O}'$; следовательно, для любого языка L над V существует такая грамматика $\Gamma\in\mathcal{O}$, что $L=\widetilde{L}_\lambda(\Gamma)\cap F(V)$.

Пусть $\Gamma\in\mathcal{O}$ и $\Pi=A_{i_1}\ldots A_{i_p}\to A_{j_1}\ldots A_{j_q}$ -некоторое неособое правило Γ. Назовем кодом правила Π цепочку $ba^{i_1}b\ldots ba^{i_p}bcba^{j_1}b\ldots ba^{j_q}b$.

Кодом грамматики Γ будем называть цепочку $dx_1d\ldots dx_sd$, где $x_1,\ldots,x_s$ – коды всех неособых правил Γ, причем среди цепочек $x_1,\ldots,x_s$ могут быть и повторяющиеся. Разумеется, каждая грамматика из $\mathcal{O}$ имеет бесконечно много кодов, но цепочка x над V может быть кодом не более чем одной грамматики; эту грамматику, если она существует, мы будем обозначать Γ_x. Назовем теперь цепочку x над V самопорождающим кодом, соответственно несамопорождающим кодом, если Γ_x существует и $x\in\widetilde{L}_\lambda(\Gamma_x)$ (соответственно $x\notin\widetilde{L}_\lambda(\Gamma_x)$).

Обозначим через L^* множество всех цепочек над V, не являющихся самопорождающими кодами. Легко видеть, что L^* не есть нс-язык. В самом деле, если бы L^* был нс – языком, существовала бы такая грамматика $\Gamma\in\mathcal{O}$, что $L^*=\widetilde{L}_\lambda(\Gamma)\cap F(V)$, и нашлась бы такая цепочка z над V, что $\Gamma=\Gamma_z$, т.е. $L^*=\widetilde{L}_\lambda(\Gamma_z)\cap F(V)$. Допустим, что $z\in L^*$. Тогда $z\in\widetilde{L}_\lambda(\Gamma_z)$, т.е. z – самопорождающий код, откуда следует $z\notin L^*$. Следовательно, $z\in L^*$ невозможно. Если же $z\notin L^*$, то (поскольку $z\in F(V)$) $z\notin\widetilde{L}_\lambda(\Gamma_z)$, т.е. z – несамопорождающий код, откуда $z\in L^*$. Таким образом, $z\notin L^*$ также невозможно. Полученное противоречие доказывает, что ни для какой цепочки z не может иметь места равенство $L^*=\widetilde{L}_\lambda(\Gamma_z)\cap F(V)$; значит, L^* – не нс- язык.

Покажем теперь, что L^* примитивно рекурсивен. Для этого поступим следующим образом. Рассмотрим новый словарь $V_1=\{\Pi p.,A,B\}$. Произвольную грамматику $\Gamma\in\mathcal{O}$ преобразуем так: а/ заменим ее вспомогательный словарь словарем V_1; б/ в каждом неособом правиле Γ заменим каждое вхождение каждого символа A_i $(i=1,2,\ldots)$ вхождением цепочки BA^iB; в/ особые правила заменим правилами $BA^3B\to a$, $BA^5B\to b$, $BA^7B\to c$, $BA^9B\to d$; г/ добавим к схеме еще одно правило $\Pi p.\to BAB$. Полученную таким преобразованием грамматику обозначим $\hat{\Gamma}$; ясно, что $\widetilde{L}_\lambda(\hat{\Gamma})\cap F(V)=\widetilde{L}_\lambda(\Gamma)\cap F(V)$.

Множество всех грамматик $\hat{\Gamma}$ будем обозначать $\hat{\mathfrak{G}}$.

Введем еще два новых символа α, β и положим $W = V \cup V_1 \cup \{\alpha, \beta\}$. Нашей задачей будет определить на множестве $F(W)$ примитивно рекурсивную функцию $K(\eta, \zeta)$, обладающую следующим свойством: Если y, z – цепочки над V, то $K(y, z) = \Lambda$ (или, если отождествить цепочки над W указанным выше способом с натуральными числами, $K(y, z) = 0$) тогда и только тогда, когда z есть код грамматики Γ_z из $\mathfrak{G}$ и $y \in L_\alpha(\Gamma_z)$. Когда такая функция будет построена, останется лишь заметить, что характеристическая функция множества L^* равна $\overline{sg}\, K(x,x) + f(x)$, где $f(x)$ – характеристическая функция множества $F(V)$, являющаяся, очевидно, примитивно рекурсивной *).

Перейдем к определению функции К.

Пусть $K_1(\xi)$ означает цепочку, полученную из ξ подстановкой A вместо каждого вхождения a, B – вместо каждого вхождения b и Λ – вместо каждого вхождения c или d. Очевидно,

$$K_1(\xi) = Sb_d(Sb_c(Sb_b(Sb_a(\xi, A), B), \Lambda), \Lambda),$$

так что K_1 примитивно рекурсивна.

Определим теперь функции $G_5'(\xi)$ и $G_5''(\xi)$ точно таким же способом, как определялась функция $G_5(\xi)$, но с той разницей, что та роль, которая принадлежала в определении G_5 символам α и β, в определении G_5' будет принадлежать соответственно символам c и d, а в определении G_5'' и роль α, и роль β передается одному символу d.

Положим

$$L(\xi) = K_1(\xi \dot{-} G_5'(\xi))$$

$$R(\xi) = K_1(G_5'(\xi))$$

---

*) То, что мы вместо функции на $F(V)$ рассматриваем функцию на $F(W)$, не влияет на результат в силу теоремы I п. II.I из [19].

Если цепочка x является кодом некоторого правила некоторой грамматики $\Gamma \in \mathfrak{G}$, то, очевидно, $L(dxd)$ есть левая часть соответствующего правила грамматики $\hat{\Gamma}$, а $R(dxd)$ – правая часть того же правила.

Далее, определим функцию $K_2(\xi, \zeta)$ следующим образом:

$$K_2(\xi, \Lambda) = \Lambda$$

$$K_2(\xi, \zeta\delta) = K_2(\xi, \zeta), \text{ если } \delta \in W \text{ и } \delta \neq d$$

$$K_2(\xi, \zeta d) = K_2(\xi, \zeta)\,\alpha\, Sb(\xi, L(G_5''(\zeta d)), R(G_5''(\zeta d)))\,\beta.$$

Функция K_2 примитивно рекурсивна.

Очевидно, если $\zeta = dx_1dx_2d \ldots dx_sd$, где $x_1, \ldots, x_s$ – коды правил над z (иначе говоря, если ζ – код грамматики), то $K_2(\xi, \zeta) = \alpha\xi_1\beta\alpha\xi_2\beta \ldots \alpha\xi_s\beta$, где ξ_i означает результат замены в ξ первого вхождения левой части правила над V_1, соответствующего тому правилу, кодом которого является x_i, вхождением правой части того же правила над V_1.

Положим теперь

$$K_3(\eta, \zeta) = K_2(\eta, \zeta)\,\alpha\, Sb(\eta, BA^3B, a)\,\beta$$
$$\alpha\, Sb(\eta, BA^5B, b)\,\beta\,\alpha\, Sb(\eta, BA^3B, c)\,\beta\,\alpha\, Sb(\eta, BA^3B, d)\,\beta.$$

Если z – код грамматики Γ, то $K_3(\eta, z)$ есть цепочка вида $\alpha\eta_1\beta \ldots \alpha\eta_\tau\beta$, где $\eta_1, \ldots, \eta_\tau$ – все цепочки, которые можно получить из η применением какого-либо правила грамматики $\hat{\Gamma}$ к первому вхождению его левой части.

Повторим теперь слово в слово все построения, выполненные при определении функции $G_{10}(\eta)$ в доказательстве леммы 5.2, с тем изменением, что в определении функции G_6 роль функции $G_1(F(G_5(\xi)))$ будет передана теперь функции $K_3(\xi,\eta)$, а в определении функции G_{10} роль $h(\eta)$ будет передана функции $2l(\xi)^{2l(\eta)+3}$. Полученную таким образом примитивно рекурсивную функцию, которая будет зависеть не только от η, но и от ξ, обозначим $G'_{10}(\eta,\xi)$.

Если z - код некоторой грамматики $\Gamma_z \in \mathcal{G}$, то $G'_{10}(\eta,z)$ обращается в нуль для тех и только для тех η, которые являются результатами таких левых выводов из $Пр.$ в соответствующей грамматике Γ, длины которых не превосходят $2l(z)^{2l(\eta)+3}$.

Для каждой грамматики Γ_z мы имеем — как, по существу, показано выше - неравенство $\tau'_{\Gamma_z}(n) \le 2d\cdot Q^{2(\sigma_{\Gamma'_z}(n)+1)}$, где Γ'_z - та грамматика из $\mathcal{G}'$, по которой построена Γ_z, и числа Q и d соответствуют грамматике Γ'_z. Поскольку z - код грамматики Γ_z, легко сообразить, что каждое из чисел Q и d меньше $l(z)$ (при этом мы считаем, конечно, что каждый вспомогательный символ грамматики Γ'_z встречается хотя бы в одном ее правиле). Кроме того, поскольку грамматика Γ'_z — неукорачивающая, имеем $\sigma_{\Gamma'_z}(n)=n$. Итак, $\tau'_{\Gamma_z}(n) \le 2l(z)\cdot l(z)^{2(n+1)} = 2l(z)^{2n+3}$

Но легко видеть, что $\tau'_{\tilde\Gamma_z}(n) = \tau'_{\Gamma_z}(n)$. Следовательно, $\tilde L_\lambda(\tilde\Gamma_z)$ совпадает с множеством цепочек, являющихся результатами таких левых выводов из Пр. в $\tilde\Gamma_z$, длины которых не превосходят $2l(z)^{2l(\eta)+3}$. Следовательно, в случае, когда z есть код грамматики, $G'_{10}(\eta,z)$ обращается в нуль тогда и только тогда, когда $\eta \in \tilde L_\lambda(\tilde\Gamma_z)$.

Пусть, наконец, $K_4(z)$ - функция, обращающаяся в нуль тогда и только тогда, когда z есть код грамматики из $\mathcal{G}$. Эта функция примитивно рекурсивна (хотя бы потому, что множество кодов грамматик из $\mathcal{G}$ является нс-языком и

даже НС-языком *)). Функция $K(y,z) = G'_{10}(y,z) + K_4(z)$ и есть та, которая нам нужна. Доказательство закончено.

§ 31. Проблемы распознавания свойств языков, порождаемых грамматиками.

В этом и двух следующих параграфах мы займемся вопросом о существовании алгоритмов, позволяющих по заданной грамматике узнавать, обладает ли порождаемый ею язык тем или иным свойством. Некоторые относящиеся сюда факты нам уже известны; так, в § 18 мы построили алгоритмы, позволяющие по заданной НС-грамматике узнавать, является ли порождаемый ею язык конечным и является ли он пустым. Сейчас, однако, нас будут интересовать главным образом те случаи, когда соответствующих алгоритмов нет, и нашей целью будет именно доказательство несуществования алгоритмов. Такие результаты хотя и не имеют непосредственной лингвистической интерпретации, но представляются существенными для лингвистических приложений - хотя бы потому, что предостерегают от поисков общих алгоритмов в тех случаях, когда этих алгоритмов заведомо нет.

Вопрос о наличии алгоритма, распознающего по грамматике Γ, обладает ли язык $L(\Gamma)$ некоторым свойством, может решаться по -разному в зависимости от того, для какого класса грамматик

---

*) Оно порождается НС-грамматикой со следующей схемой:

$$\left\{\begin{array}{l} Пр. \to Пр.\mathcal{D}d \\ Пр. \to d\mathcal{D}d \\ \mathcal{D} \to bAb\mathcal{D}bAb \\ \mathcal{D} \to C \\ C \to CbAb \\ C \to c \\ A \to Aa \\ A \to a \end{array}\right.$$

этот вопрос поставлен. Так, например, не существует алгоритма, позволяющего по любой нс-грамматике узнать, является ли порождаемый ею язык конечным (это будет доказано в следующем параграфе) ; в то же время для кс- грамматик такой алгоритм имеется. В связи с этим мы введем следующее определение :

Пусть $\mathcal{K}$ - класс грамматик и Θ - свойство языков. Мы будем говорить, что свойство Θ распознаваемо в классе $\mathcal{K}$, если существует алгоритм, позволяющий по любой грамматике класса $\mathcal{K}$ узнать, обладает ли она свойством Θ.

Если K_1, K_2 - классы грамматик и $K_2 \subseteq K_1$, то, очевидно, всякое свойство языков, распознаваемое в классе K_1, будет распознаваемо и в классе K_2. В частности, всякое свойство, распознаваемое в классе нс-грамматик, распознаваемо и в классе кс-грамматик, а всякое свойство, распознаваемое в классе кс- грамматик, распознаваемо и в классе α- грамматик. Обратные утверждения, как мы увидим ниже, неверны.

Исходным пунктом всех конструкций этого и следующих параграфов будет теорема Райса ([19] , стр.147):

Если $\mathfrak{F}$ - непустое семейство одноместных частично рекурсивных функций, отличное от совокупности всех таких функций, то множество клиниевских номеров функций, принадлежащих $\mathfrak{F}$, не может быть рекурсивным.

Для приложения к грамматикам нам будет удобно пользоваться другой формулировкой этой теоремы - в терминах машин Тьюринга. Чтобы получить эту формулировку, введем следующее определение: класс машин Тьюринга М называется инвариантным, если, какова бы ни была машина Т, принадлежащая М и вычисляющая одноместную (словарную) функцию F_T ж), любая другая машина, вычисляющая ту же функцию F_T, также принадлежит М.

Теперь мы можем сформулировать теорему Райса так:
Если М - непустой инвариантный класс машин Тьюринга, внешние

ж) Очевидно, всякая машина Тьюринга вычисляет какую-то одноместную словарную функцию.

словари которых содержат символы a_0 и a_1, и М не совпадает с совокупностью всех таких машин, то не существует алгоритма, позволяющего по любой машине Тьюринга ж) с внешним словарем, содержащим a_0 и a_1, узнать, принадлежит ли эта машина классу М.

Эта формулировка получается из предыдущей следующим образом. Пусть T - универсальная машина Тьюринга с внешним словарем $\{a_0, a_1\}$, вычисляющая функцию Клини $K(x,y)$ ([19] , стр. 271).
По программе машины T и произвольному натуральному числу n без труда строится программа машины T_n, вычисляющей одноместную функцию $K(n,x)$. Допустим теперь, что алгоритм, указанный в формулировке теоремы, существует. Применяя его к программе машины T_n, мы можем узнать, принадлежит ли машина T_n классу $\mathcal{M}$ - иначе говоря, принадлежит ли функция $K(n,x)$ семейству $\mathfrak{F}_M$, состоящему из всевозможных одноместных функций, вычисляемых машинами класса М. Таким образом, мы получаем алгоритм, позволяющий по клиниевскому номеру частично рекурсивной функции узнать, принадлежит ли эта функция семейству $\mathfrak{F}_M$. А это означает, что множество клиниевских номеров функций из $\mathfrak{F}_M$ рекурсивно, что невозможно.

В дальнейшем, ссылаясь на теорему Райса, мы всегда будем иметь в виду вторую формулировку.

Пусть $\mathcal{L}$ - класс языков и Θ - свойство языков. Скажем, что Θ нетривиально в классе $\mathcal{L}$, если $\mathcal{L}$ содержит как языки, обладающие свойством Θ, так и языки, им не обладающие.

Теорема 5.6. Ни одно свойство языков, нетривиальное в классе всех языков, порождаемых грамматиками, не является распознаваемым в классе всех грамматик.

ж) Или, что то же самое, по любой программе машины Тьюринга.

Доказательство. Пусть T - произвольная машина Тьюринга с внешним словарем $\{a_0, a_1, \ldots, a_n\}$. Одноместную функцию в редуцированном словаре $\{a_1, \ldots, a_n\}$, вычисляемую машиной T, мы будем обозначать f_T, а множество значений этой функции - E_T.

По программе машины T нетрудно эффективно построить программу другой машины Тьюринга T_1, правильно вычисляющей ту же функцию f_T. Идея построения машины T_1 состоит в том, что новая машина должна работать так же, как старая, с той только разницей, что : а/ всякий раз, когда старая машина надстраивает ячейку слева, новая вместо этого сдвигает все содержимое ленты на одну ячейку вправо (для этого нужно в начале работы поставить в крайней слева ячейке ленты некоторую "метку" и выполнять сдвиг каждый раз, когда по старой программе головка должна идти из помеченной ячейки влево); б/ после окончания вычисления по старой программе новая машина "прогоняет" все нули, кроме того, который стоит в крайней слева ячейке, в правый конец ленты (чтобы иметь возможность это сделать, нужно в начале работы поставить еще одну "метку" в правом конце ленты, а потом по мере надстраивания ячеек сдвигать ее вправо), а затем головка переходит в крайнюю слева ячейку, после чего машина останавливается ; в/ в случаях, когда старая машина останавливается, не перейдя в заключительное состояние, новая зацикливается (например, оставляя головку на одном месте и периодически меняя состояния). Детальное построение программы машины T_1 предоставляется читателю.

Далее, по программе машины T_1 в силу теоремы 5.I можно эффективно построить грамматику, порождающую язык $E_{T_1} = E_T$. Итак, существует алгоритм, позволяющий по программе произвольной машины Тьюринга T построить грамматику Γ_T, порождающую язык E_T.

Допустим теперь, что для некоторого свойства языков Θ, нетривиального в классе всех языков, порождаемых грамматиками (или - что по теореме 5.4, означает то же самое - в классе множеств значений всевозможных одноместных (словарных) функций, вычисляемых машинами Тьюринга), существует алгоритм, позволяющий

по любой грамматике Γ распознать, обладает ли язык $L(\Gamma)$ свойством Θ . Обозначим через $\mathcal{M}(\Theta)$ класс, состоящий из всех тех машин Тьюринга T, для которых соответствующие множества E_T обладают свойством Θ . Тогда для произвольной машины T мы можем построить грамматику Γ_T и затем узнать, обладает ли язык $L(\Gamma_T) = E_T$ свойством Θ , т.е. принадлежит ли T классу $\mathcal{M}(\Theta)$. Таким образом, мы получаем алгоритм, позволяющий для любой машины T узнавать, принадлежит ли она классу $\mathcal{M}(\Theta)$; а это противоречит теореме Райса.

§ 32. Проблемы распознавания свойств нс-языков.

Только что мы показали, что в классе всех грамматик каждое нетривиальное свойство нераспознаваемо. Уже в классе нс-грамматик дело обстоит иначе. Действительно, пусть x - некоторая цепочка и Θx есть свойство языка содержать цепочку x . В начале § 30 был построен простой алгоритм, позволяющий по любой неукорачивающей грамматике и тем более по любой нс-грамматике распознать, обладает ли порождаемый ею язык свойством Θx .

Для очень широкого класса свойств оказывается, однако, возможным доказать их нераспознаваемость в классе нс-грамматик. Чтобы сделать это, предварительно докажем несколько лемм.

Лемма 5.3. Для произвольной машины Тьюринга T с внешним словарем $\{a_0, a_1, \ldots, a_n\}$ и внутренним словарем $\{q_0, q_1, \ldots, q_s\}$ (где q_1 - начальное состояние и q_0 - заключительное) можно эффективно построить такие нс-грамматики Γ_T' и Γ_T'', что: а/ основной словарь каждой из них состоит из одного символа b; б/ если функция f_T определена для цепочки, состоящей из одного символа a_1, то $L(\Gamma_T') = \{b^n \mid n \geq n_T\}$ и $L(\Gamma_T'') = \{b^n \mid n \leq m_T\}$; в/ если f_T не определена для a_1, то $L(\Gamma_T') = \emptyset$ и $L(\Gamma_T'') = \{b^n \mid n = 1, 2, \ldots\}$. Здесь n_T и m_T - натуральные числа, зависящие от машины T .

Доказательство. В силу теоремы 3.3. достаточно построить по машине T неукорачивающие грамматики Δ'_T и Δ''_T, удовлетворяющие условиям а/, б/, в/ .

Грамматику Δ'_T построим следующим образом. Ее основной словарь будет состоять из одного символа b, не входящего ни во внешний, ни во внутренний словари машины T. Вспомогательный словарь будет представлять собой об"единение внешнего и внутреннего словарей машины T с добавлением двух новых символов $Пр.$ и v. Начальным символом будет $Пр$. Схема грамматики Δ'_T будет представлять собой об"единение систем правил S_0, S_1 и S'_2, где : S_0 строится в точности так же, как одноименная система правил в доказательстве теоремы 5.I ; S_1 состоит из одного правила

$$Пр. \to q_1 a_0 a_1 v \ ;$$

S'_2 состоит из правил

$$\begin{cases} q_0 \to b \\ b a_i \to bb \\ b v \to bb \\ b \to bb \end{cases} \qquad i = 0, 1, \ldots, n$$

Рассуждениями, совершенно аналогичными примененным в доказательстве теоремы 5.I, устанавливается, что цепочка, содержащая q_0, выводима в Δ'_T из $Пр$ тогда и только тогда, когда f_T определена для a_1. При этом, если такая цепочка выводима из $Пр.$, то она имеет вид $q_0 y v$, где y - цепочка над $\{a_0, a_1, \cdots, a_n\}$; а из этой цепочки по правилам системы S'_2 выводима любая цепочка вида b^n, где $n \geqslant \ell(q_0 y v)$ (и не выводима ни одна цепочка вида b^m, где $m < \ell(q_0 y v)$) . Если же из $Пр.$ не выводима никакая цепочка, содержащая q_0, то не выводима и никакая цепочка вида b^n.

Прежде чем строить грамматику Δ''_T, заметим, что по машине T нетрудно построить другую машину Тьюринга T_1, правильно вычисляющую ту же самую функцию f_T и такую, что если машина T, начиная работу с какой-то цепочки, не останавливается, то в процессе ее работы появляются сколь угодно длинные машинные цепочки. Идея построения машины T_1 состоит в следующем. Новая машина работает так же, как старая, с той только разницей, что в начале работы она ставит "метку" в крайней справа ячейке ленты, а затем перед каждым шагом, который нужно сделать по старой программе, головка идет вправо до метки, надстраивает еще одну ячейку (с записанным в ней "пустым" символом a_0), передвигает туда метку и возращается на прежнее место (которое тоже должно быть помечено, чтобы его можно было найти), после чего машина делает нужный шаг. Детальное построение программы машины T_1 предоставляется читателю.

Грамматика Δ''_T строится теперь так. Ее словари и начальный символ определяются так же, как для Δ'_T (но по словарям машины T_1, а не T). Схема представляет собой об"единение $S_0 \cup S_1 \cup S''_2 \cup S''_3$, где S_0 и S_1 строятся так же, как для Δ'_T (но опять-таки по программе машины T_1), а S''_2 и S''_3 имеют следующий вид:

$$S''_2 : \begin{cases} a_i \to b \\ q_\alpha \to b \\ v \to b \end{cases} \qquad \begin{array}{l} i = 0, 1, \ldots, n \\ \alpha = 0, 1, \ldots, s \end{array}$$

$$S''_3 : \begin{cases} Пр. \to b \\ Пр. \to bb \\ Пр. \to bbb \end{cases}$$

Пусть функция $f_T = f_{T_1}$ определена для a_1, т.е. процесс работы машины T_1, начинающийся с машинной цепочки $q_1 a_0 a_1$, заканчивается. Если m_T - длина последней машинной цепочки этого процесса, то из $Пр.$ по правилам систем S_1, S_0 и S_2'' выводима в Δ_T'' любая цепочка вида b^n, где $4 \le n \le m_T$, и никакая цепочка длины $> m_T$ из $Пр.$ не выводима. Если же машина T_1 не останавливается, начиная работу с a_1, то из $Пр.$ можно вывести по правилам систем S_1' и S_0 цепочку над $\{a_0, \dots, a_n\} \cup \cup \{q_0, \dots, q_s\} \cup \{\upsilon\}$ любой длины $\geqslant 4$; следовательно, в этом случае из $Пр.$ выводима любая цепочка вида b^n. Доказательство закончено.

Лемма 5.4. Пусть для произвольных языков L и L' $M(L, L')$ означает

$$\{x \mid x \in L \,\&\, \exists y (y \in L' \,\&\, \ell(y) = \ell(x)\}.$$

Тогда для любых двух нс-грамматик Γ и Γ' можно эффективно построить нс-грамматику $\widetilde{\Gamma}$, порождающую язык

$$M(L(\Gamma), L(\Gamma')).$$

Доказательство. Пусть $\Gamma = \langle V, V_1, Пр., S \rangle$,

$$\Gamma' = \langle V', V_1', Пр.', S' \rangle$$

В силу теоремы 3.4 достаточно построить грамматику без существенного укорачивания Γ^*, порождающую $M(L(\Gamma), L(\Gamma'))$.

Мы можем без ограничения общности считать, что

$$(V \cup V_1) \cap (V' \cup V_1') = \varnothing$$

Сопоставим каждому символу $a \in V$ новый символ $\overline{a}$. Множество всех таких новых символов обозначим $\overline{V}$. Обозначим, кроме того, через $\overline{S}$ систему правил, получаемую из S заменой в каждом правиле каждого вхождения символа $a \in V$ вхождением символа $\overline{a}$. Введем еще один новый символ $Пр.^*$

и положим $\Gamma^* = \langle V, V_1^*, Пр.^*, S^* \rangle$,

где $V_1^* = \overline{V} \cup V_1 \cup V_1' \cup \{Пр.^*\} \cup V'$,

а S^* получается из $\overline{S} \cup S'$ добавлением новых правил:

$$Пр.^* \longrightarrow Пр.\,Пр.'$$

$$\overline{a} b \longrightarrow b \overline{a}, \qquad a \in V,\ b \in V'$$

$$b \overline{a} \longrightarrow a, \qquad a \in V,\ b \in V'.$$

Построенная грамматика, очевидно, является грамматикой без существенного укорачивания, т.к. в ее схеме укорачивающими являются только правила вида $b\overline{a} \to a$, а они- заключительные.

Легко видеть, что цепочка x над V выводима из $Пр.$ в Γ^* тогда и только тогда, когда $x \in L(\Gamma)$ и язык $L(\Gamma')$ содержит цепочку, длина которой равна длине x. Лемма доказана.

Для произвольного языка L и произвольного натурального числа n будем полагать

$$L^{(n)} = \{x \mid x \in L \,\&\, \ell(x) \le n\},$$

$$\overline{L}^{(n)} = \{x \mid x \in L \,\&\, \ell(x) \geqslant n\}$$

Лемма 5.5. Если L - нс-язык и K -класс языков, содержащий пустой язык и не содержащий ни одного из языков $\overline{L}^{(n)}$, то свойство принадлежать классу K нераспознаваемо в классе нс-грамматик.

Лемма 5.6. Если L -нс-язык и K -класс языков, содержащий L и не содержащий ни одного из языков $L^{(n)}$, то свойство принадлежать классу K нераспознаваемо в классе нс-грамматик.

Доказательство леммы 5.5.

Пусть T - произвольная машина Тьюринга и f_T-вычисляемая ею одноместная функция. Как в доказательстве теоремы 5.5, по машине T эффективно строится машина T_1, правильно вычисляющая ту же функцию f_T. Для машины T_1 построим грамматику Γ'_{T_1} по лемме 5.3. Обозначим через Γ нс- грамматику, порождающую язык L, и построим по лемме 5.4 нс-грамматику $\widetilde{\Gamma}_T$, порождающую язык $M(L(\Gamma), L(\Gamma'_{T_1}))$.

Если функция $f_T = f_{T_1}$ определена для a_1, то $L(\Gamma'_{T_1}) = \{b^n \mid n \geq n_T\}$ и, следовательно, $L(\widetilde{\Gamma}_T) = \overline{L}^{(n_T)}$. В противном случае язык $L(\Gamma'_{T_1})$ пуст, так что пуст и язык $L(\widetilde{\Gamma}_T)$.

Таким образом, язык $L(\widetilde{\Gamma}_T)$ принадлежит классу K тогда и только тогда, когда функция f_T не определена для a_1.

Допустим теперь, что существует алгоритм, позволяющий по любой нс-грамматике распознать, принадлежит ли порождаемый ею язык классу K. Тогда мы могли бы для произвольной машины Тьюринга T, внешний словарь которой содержит a_0 и a_1, построить грамматику $\widetilde{\Gamma}_T$ и затем, воспользовавшись упомянутым алгоритмом, узнать, принадлежит ли язык $L(\widetilde{\Gamma}_T)$ классу K. Тем самым мы распознали бы, определена ли функция f_T для a_1. Таким образом, у нас был бы алгоритм, позволяющий для любой машины Тьюринга T с внешним словарем, содержащим a_0 и a_1, узнать, принадлежит ли a_1 области определенности функции f_T. Но это противоречит теореме Райса.

Лемма 5.6 доказывается совершенно аналогично с заменой грамматики Γ'_{T_1} на Γ''_{T_1}. Детальное проведение доказательства предоставляется читателю.

Теперь мы можем доказать основную теорему этого параграфа.

Будем говорить, что языки L_1 и L_2 <u>почти совпадают</u>,

если они отличаются друг от друга лишь конечным числом элементов, т.е. если язык $(L_1 \setminus L_2) \cup (L_2 \setminus L_1)$ конечен.

Отношение почти совпадения является, очевидно, эквивалентностью. Классы, на которые эта эквивалентность разбивает множество всех нс-языков, мы будем называть <u>пучками</u>.

<u>Теорема 5.7</u>. Пусть K -непустой класс нс-языков, не пересекающийся хотя бы с одним пучком. Тогда свойство принадлежать классу K нераспознаваемо в классе нс-грамматик.

<u>Доказательство</u>. Пусть сначала K не содержит ни одного конечного языка. Тогда существует бесконечный язык L, принадлежащий K. В то же время ни один язык $L^{(n)}$ классу K не принадлежит. По лемме 5.6 отсюда следует нераспознаваемость свойства принадлежать классу K.

Пусть теперь K содержит конечный язык L_0. Обозначим через K' класс языков, определяемый следующим образом: $L \in K'$ тогда и только тогда, когда $L \cup L_0 \in K$. Класс K', очевидно, содержит пустой язык. Далее, существует пучок, не пересекающийся с классом K, а, значит, и с классом K'. Если L - произвольный язык из этого пучка, то любой язык $\overline{L}^{(n)}$ принадлежит тому же пучку и, следовательно, не принадлежит K. По лемме 5.5. свойство принадлежать K' нераспознаваемо. Но для любой нс-грамматики Γ можно (по теореме 3.II) эффективно построить нс-грамматику, порождающую язык $L(\Gamma) \cup L_0$; этот язык тогда и только тогда принадлежит K, когда $L(\Gamma) \in K'$. Поэтому распознаваемость свойства принадлежать K влекла бы за собой распознаваемость свойства принадлежать K', так что свойство принадлежать K также нераспознаваемо.

В качестве примеров применения теоремы 5.7 выведем из нее некоторые следствия.

<u>Следствие I.</u> Пусть Θ - свойство языков, нетривиальное в классе нс-языков. В следующих случаях свойство Θ нераспознаваемо в классе нс-грамматик:

а/ Если свойством Θ обладает лишь конечное число языков.
б/ Если свойством Θ обладают все конечные языки.
в/ Если свойством Θ обладают только кс-языки.

Доказательство. а/ сразу следует из того, что число пучков бесконечно; б/ - из того, что конечные языки образуют пучок (и из того, что отрицание нераспознаваемого свойства нераспознаваемо). Чтобы доказать в/, достаточно заметить, что всякий язык, почти совпадающий с кс-языком, сам является кс-языком (это следует из теорем 3.II и 3.I2 , поскольку всякий язык с конечным дополнением - автоматный), и поэтому существуют пучки, не содержащие кс-языков.

В частности, нераспознаваемы в классе нс-грамматик следующие свойства языков: быть пустым (по пункту а/ или в/), иметь пустое дополнение (по пункту а/или в/), быть конечным (по пункту б/или в/), иметь конечное дополнение (по пункту в/), быть автоматным (по пункту б/или в/), быть контекстно-свободным (по пункту б/или в/), иметь контекстно-свободное дополнение (по пункту б/).

Следствие 2. Ни для одной нс-грамматики Γ_0 не существует алгоритма, позволяющего по любой нс-грамматике Γ узнавать, эквивалентна ли она грамматике Γ_0 .

Доказательство. Обозначим через Θ_0 свойство языка совпадать с языком $L(\Gamma_0)$. Поскольку этим свойством обладает только один язык, оно нераспознаваемо по пункту а) следствия I.

Следствие 3. Если язык L_0 (не обязательно нс-) имеет бесконечное дополнение, то не существует алгоритма, позволяющего по нс-грамматике Γ узнавать, содержится ли язык $L(\Gamma)$ в языке L_0 .

Доказательство. Обозначим через Θ_0' свойство языка содержаться в L_0 . Всякий язык, обладающий свойством Θ_0' , имеет бесконечное дополнение, но языки с конечными дополнениями образуют пучок.

Следствие 4. Если язык L_0 бесконечен, то не существует: а/ алгоритма, позволяющего по нс-грамматике Γ узнавать, содержится ли язык L_0 в языке $L(\Gamma)$; б/ алгоритма, позволяющего по нс-грамматике Γ узнавать, является ли пересечение $L_0 \cap L(\Gamma)$ пустым.

Доказательство. а/ Обозначим через Θ_0'' свойство языка содержать L_0 и через $\overline{\Theta_0''}$ отрицание свойства Θ_0'' .

Всякий язык, обладающий свойством $\overline{\Theta_0''}$, бесконечен; поэтому все конечные языки обладают свойством $\overline{\Theta_0''}$, и по пункту б/ следствия I оно- а вместе с ним и Θ_0'' -нераспознаваемо. б/ вытекает из следствия 3, поскольку $L_0 \cap L(\Gamma) = \varnothing$ равносильно $L(\Gamma) \subseteq CL_0$.

Замечание. Нетрудно видеть, что для конечного языка L_0 алгоритмы, указанные в формулировке следствия 4, существуют. Это вытекает из замечания, сделанного в начале настоящего параграфа. Точно так же, если язык L_0 имеет конечное дополнение, то существует алгоритм, указанный в формулировке следствия 3.

§ 33. Проблемы распознавания свойств кс-языков.

Для кс-языков класс распознаваемых свойств шире, чем для нс-языков - пустота, конечность и некоторые другие свойства, нераспознаваемые в классе нс- языков, в классе кс-языков распознаваемы. Однако многие важные свойства остаются нераспознаваемыми и здесь. Чтобы установить это, введем одно новое понятие, относящееся к машинам Тьюринга.

Пусть T -машина Тьюринга с внешним словарем $\{a_0, a_1\}$ и внутренним словарем $\{q_0, q_1, \dots, q_s\}$ (q_1 - начальное состояние , q_0 - заключительное). Введем два новых символа q, d . Для произвольной машинной цепочки ω назовем ее образом цепочку, полученную из ω заменой вхождения символа q_α вхождением подцепочки $q^{\alpha+1}$. Будем называть протоколом машины T всякую цепочку вида

$$d\omega_0 d d\omega_1 d \dots d\omega_t d$$

,где:

1/ ω_0 - образ машинной цепочки ω , содержащей q_1 ;
2/ для всякого $i=1,\dots,t$ ω_i есть образ цепочки $\omega^{(i)}$;
3/ ω_t является образом цепочки, содержащей q_0 . Множество всех протоколов машины T будем обозначать Π_T . Положим $W=\{a_0, a_1, q, d\}$ и $\Pi_T' = F'(W) \setminus \Pi_T$ ($F'(W)$ означает множество всех непустых цепочек над W) .

Основную роль в конструкциях этого параграфа будет играть следующая лемма:

Лемма 5.7. Для любой машины Тьюринга T с внешним словарем

$\{a_0 a_1\}$ множество Π'_t является кс-языком. При этом кс-грамматика, порождающая Π'_T, строится по T эффективно.

<u>Доказательство</u>. Назовем <u>квазипротоколом</u> всякую цепочку вида $d\omega_0 d d\omega_1 d \ldots d\omega_t d$, где $\omega_0, \ldots, \omega_t$ - образы машинных цепочек, причем ω_0 - образ цепочки, содержащей q_1, а ω_t - образ цепочки, содержащей q_0. Множество всех квазипротоколов обозначим K. Положим также $K' = F'(W) \setminus K$.

Покажем, что K является а-языком.

Пусть $\mathcal{M}$ - множество всех цепочек вида $d\omega d$, где ω - образ машинной цепочки, и $\mathcal{M}_\alpha$ - множество всех цепочек вида $d\omega d$, где ω - образ машинной цепочки, содержащей q_α $(\alpha = 0, 1, 2, \ldots)$.

Языки $\mathcal{M}$ и $\mathcal{M}_\alpha$, очевидно, автоматные (построение соответствующих a - грамматик предоставляется читателю).

Положим $N_1 = \mathcal{M}^*$ (звездочка - знак итерации, см. стр. 103), и $N_2 = N_1 \cup \{\Lambda\}$. Очевидно, $K = \mathcal{M}_1 \times N_2 \times \mathcal{M}_0$, откуда по теореме 3.II следует, что K есть а - язык и его а - грамматику можно эффективно построить по а- грамматикам языков $\mathcal{M}$, $\mathcal{M}_0$ и $\mathcal{M}_1$.

Пусть теперь P_T означает множество всевозможных цепочек вида $d\omega_1 d d\omega_2 d$, где ω_1 и ω_2 - соответственно образы машинных цепочек φ_1 и φ_2, причем $\varphi_2 \neq \varphi_1^{(1)}$.

Имеем следующее очевидное тождество:

$$\Pi'_T = K' \cup [K \cap (N_2 \times P_T \times N_2)].$$

Из этого тождества в силу теорем 3.II и 3.I2 вытекает, что если P_T - кс-язык, то и Π'_T - кс-язык, причем кс-грамматика, порождающая Π'_T, строится эффективно по кс-грамматике, порождающей P_T, и а-грамматикам, порождающим N_2 и K.

Итак, для завершения доказательства леммы достаточно показать, что по машине T эффективно строится кс-грамматика, порождающая P_T.

Обозначим через $P_T^{i\alpha}$ $(i = 0, 1;\ \alpha = 0, 1, 2, \ldots, s)$ множество всех цепочек вида $d\omega_1 d d\omega_2 d$, где ω_1 - образ машинной цепочки, содержащей вхождение подцепочки $q_\alpha a_i$ и ω_2 - образ машинной цепочки, <u>не</u> получающейся из ω_1 применением команды с левой частью $q_\alpha a_i$. (Эту команду, если она существует, мы будем обозначать $K_{i\alpha}$).

Очевидно,
$$P_T = \bigcup_{\substack{i=0,1 \\ \alpha=0,\ldots,s}} P_T^{i\alpha}$$

Поэтому нам достаточно уметь для каждой пары i, α $(i = 0, 1;\ \alpha = 0, \ldots, s)$ строить кс-грамматику, порождающую $P_T^{i\alpha}$. Мы покажем, что такую кс-грамматику действительно можно эффективно построить.

Обозначим через $M_{i\alpha}$ множество всех цепочек вида $d\omega d$, где ω - образ машинной цепочки, содержащей вхождение подцепочки $q_\alpha a_i$. Положим $Q_{i\alpha} = M_{i\alpha} \times M$. Легко построить а-грамматику, порождающую $M_{i\alpha}$ (это предоставляется читателю). Поэтому и $Q_{i\alpha}$ - а - язык.

Будем различать четыре случая:

I/. Команда $K_{i\alpha}$ не существует. В этом случае $P_T^{i\alpha} = Q_{i\alpha}$, так что $P_T^{i\alpha}$ порождается а-грамматикой, которую легко выписать, зная i и α.

2/. Команда $K_{i\alpha}$ имеет вид $q_\alpha a_i \to q_\beta a_j$.

Для произвольной цепочки ω под словарем $\{a_0, a_1, q\}$ будем обозначать через $l'(\omega)$ общее число вхождений в ω символов a_0 и a_1. Обозначим через R, R_1, R_2 множества всевозможных цепочек вида $d\omega_1 d d\omega_2 d$, где ω_1 и ω_2 - цепочки над $\{a_0, a_1, q\}$ такие, что:

$l'(\omega_1) \neq l'(\omega_2)$ в случае R; $l'(\omega_1) < l'(\omega_2)$ в случае R_1; $l'(\omega_1) > l'(\omega_2)$ в случае R_2.

R_1 порождается, как легко проверить, кс-грамматикой со следующей схемой :

$$
\begin{aligned}
&\text{Пр} \longrightarrow dAd \\
&A \longrightarrow A a_i, \quad i = 0,1 \\
&A \longrightarrow A q \\
&A \longrightarrow B a_i, \quad i = 0,1 \\
&B \longrightarrow a_i B a_j, \quad i,j = 0,1 \\
&B \longrightarrow B q \\
&B \longrightarrow q B \\
&B \longrightarrow d d
\end{aligned}
$$

Совершенно аналогично строится кс- грамматика, порождающая R_2. Отсюда, поскольку $R = R_1 \cup R_2$, следует, что и R порождается кс- грамматикой.

Пусть теперь $\omega_1 = \alpha_u \alpha_{u-1} \cdots \alpha_1$, $\omega_2 = \beta_v \beta_{v-1} \cdots \beta_1$,

где каждое α_k и β_m есть либо один из символов a_0, a_1, либо степень символа q, и $u \le v$. Мы будем говорить, что цепочка ω_2 нарушает согласование с цепочкой ω_1 на к-м месте $(k = 1, 2, \ldots, u-1)$, если выполняется одно из двух: 1/ $\alpha_{k+1}\alpha_k \ne q^{\alpha+1} a_i$ и $\beta_{k+1}\beta_k \ne \alpha_{k+1}\alpha_k$;

2/ $\alpha_{k+1}\alpha_k = q^{\alpha+1} a_i$ и

$\beta_{k+1}\beta_k \ne q^{\beta+1} a_j$.

Множество всех цепочек вида $d\omega_1 dd\omega_2 d$, где ω_1, ω_2 - цепочки над $\{a_0, a_1, q\}$ такие, что ω_2 на некотором месте нарушает согласование с ω_1, мы обозначим $S^{i\alpha}_{j\beta}$.

Ясно, что цепочка $d\omega_1 dd\omega_2 d$ принадлежит $P_T^{i\alpha}$ тогда и только тогда, когда она, во-первых, принадлежит $Q_{i\alpha}$ и, во-вторых, либо $l'(\omega_1) \ne l'(\omega_2)$, либо цепочка ω_2 на каком-то месте нарушает согласование с ω_1.

Иначе говоря, имеем $P_T^{i\alpha} = Q_{i\alpha} \cap (R \cup S^{i\alpha}_{j\beta})$.

Поэтому достаточно построить кс-грамматику, порождающую $S^{i\alpha}_{j\beta}$. Но нетрудно непосредственно проверить, что $S^{i\alpha}_{j\beta}$ порождается кс- грамматикой со следующей схемой:

$$
\begin{aligned}
&\text{Пр.} \longrightarrow dAd \\
&A \longrightarrow \alpha A, \quad (\alpha = a_0, a_1, q, q^2, \ldots, q^{s+1}) \\
&A \longrightarrow \gamma\delta B_{\gamma\delta} \quad (\gamma, \delta = a_0, a_1, q, \ldots, q^{s+1}) \\
&B_{\gamma\delta} \longrightarrow \alpha B_{\gamma\delta} \beta \quad (\alpha, \beta, \gamma, \delta = a_0, a_1, q, \ldots, q^{s+1}) \\
&B_{\gamma\delta} \longrightarrow C_{\gamma'\delta'} \begin{cases} \gamma, \delta, \gamma', \delta' = a_0, a_1, q, \ldots, q^{s+1} \\ \text{если } \gamma\delta \ne q^{\alpha+1} a_i, \text{ то } \gamma'\delta' \ne \gamma\delta \\ \text{если } \gamma\delta = q^{\alpha+1} a_i, \text{ то } \gamma'\delta' \ne q^{\beta+1} a_j \end{cases} \\
&C \longrightarrow C\alpha \quad (\alpha = a_0, a_1, q, \ldots, q^{s+1}) \\
&C \longrightarrow dd.
\end{aligned}
$$

3/ Команда $K_{i\alpha}$ имеет вид $q_\alpha a_i \longrightarrow q_\beta R$.

В этом случае $P_T^{i\alpha} = \bar{P}_T^{i\alpha} \cup \bar{\bar{P}}_T^{i\alpha}$, где $\bar{P}_T^{i\alpha}$ состоит из тех цепочек $d\omega_1 dd\omega_2 d \in P_T^{i\alpha}$, которые не содержат вхождений подцепочки $q^{\alpha+1} a_i dd$, а $\bar{\bar{P}}_T^{i\alpha}$ - из всех остальных цепочек $\in P_T^{i\alpha}$. (Иначе говоря, $\bar{P}_T^{i\alpha}$ содержит цепочки вида $d\omega_1 dd\omega_2 d$, для которых переход от машинной цепочки, образом которой является

w_1, к цепочке, получаемой из нее с помощью команды $K_{i\alpha}$, не требует надстраивания ячейки справа, а $\overline{\overline{P}}{}_T^{i\alpha}$ содержит те цепочки, в которых при указанном переходе происходит надстраивание).

Достаточно построить кс- грамматики для $\overline{P}{}_T^{i\alpha}$ и $\overline{\overline{P}}{}_T^{i\alpha}$ по отдельности.

а/ Пусть $w_1 = \alpha_u \alpha_{u-1} \dots \alpha_1$, $w_2 = \beta_v \beta_{v-1} \dots \beta_1$,

где каждое α_k и β_m есть либо один из символов a_0, a_1, либо степень символа q, и $u \le v$.
Будем говорить, что w_2 <u>нарушает согласование с w_1 на k-м месте</u> ($k = 0, 1, \dots, u-1$), если выполняется одно из двух:

1/ $\alpha_{k+1}\alpha_k \ne q^{\alpha+1} a_i$ и $\beta_{k+1}\beta_k \ne \alpha_{k+1}\alpha_k$;

2/ $\alpha_{k+1}\alpha_k = q^{\alpha+1} a_i$ и $\beta_{k+1}\beta_k \ne a_i q^{\beta+1}$.

Множество всех цепочек вида dw_1ddw_2d, где w_1, w_2 - цепочки над $\{a_0, a_1, q\}$ такие, что w_1 не кончается подцепочкой $q^\alpha a_i$ и w_2 на каком-нибудь месте нарушает согласование с w_1, мы обозначим $\overline{T}{}_\beta^{i\alpha}$.

Легко видеть, что

$$\overline{P}{}_T^{i\alpha} = Q_{i\alpha} \cap \left(R \cup \overline{T}{}_\beta^{i\alpha}\right),$$

где R означает то же, что в предыдущем случае. Поэтому достаточно построить кс-грамматику для $\overline{T}{}_\beta^{i\alpha}$. Эта грамматика строится совершенно аналогично тому, как в предыдущем случае строилась грамматика для $S_{\gamma\beta}^{i\alpha}$; поэтому мы опускаем ее построение.

б/ Обозначим через R' множество всевозможных цепочек вида dw_1ddw_2d, где w_1 и w_2 - цепочки над $\{a_0, a_1, q\}$ такие, что $l'(w_2) \ne l'(w_1)+1$. Язык R' порождается кс-грамматикой, сходной с построенной

выше грамматикой для R; построение этой грамматики предоставляется читателю.

Далее, обозначим через $\overline{\overline{T}}{}_\beta^{i\alpha}$ множество всех цепочек вида $dw_1ddw_2a_0d$, где w_1, w_2 - цепочки над $\{a_0, a_1, q\}$ такие, что w_1 кончается подцепочкой $q^\alpha a_i$ и w_2 на каком-нибудь месте нарушает согласование с w_1 (в смысле предыдущего пункта).

Легко видеть, что

$$\overline{\overline{P}}{}_T^{i\alpha} = Q_{i\alpha} \cap \left(R' \cup \overline{\overline{T}}{}_\beta^{i\alpha}\right),$$

так что и для $\overline{\overline{P}}{}_T^{i\alpha}$ эффективно строится кс-грамматика.

4/ Команда $K_{i\alpha}$ имеет вид $q_\alpha a_i \to q_\beta L$.

Доказательство для этого случая совершенно аналогично предыдущему, и мы его опускаем.

Итак, лемма доказана.

Докажем теперь еще одну лемму.

<u>Лемма 5.8.</u> Если машина Тьюринга T такова, что каждый ее протокол содержит не менее трех образов машинных цепочек, и при этом область определенности функции f_T бесконечна, то Π_T не есть кс-язык.

<u>Доказательство</u>. Допустим, что для некоторой машины T, обладающей указанными свойствами, множество Π_T есть кс-язык. Пусть x - цепочка, принадлежащая области определенности f_T и такая, что $l(x) \ge 2\max(r, s)$, где r, s - числа из теоремы 3.8, соответствующие языку Π_T.
Язык Π_T содержит цепочку $\varphi = dw_0ddw_1d \dots dw_td$, где $w_0 = q^2 a_0 x$ и для каждого $i = 1, \dots, t$ w_i есть образ цепочки $(q_1 a_0 x)^{(i)}$.
Очевидно, для каждого $i = 0, 1, \dots, t$, $l'(w_i) > 2\max(r, s)$.

По теореме 3.8 цепочку φ можно представить в виде $\varphi = z_1 y_1 x y_2 z_2$, где хотя бы одна из цепочек y_1, y_2 не пуста, $l(y_1 x y_2) \le s$ и при любом $n = 1, 2, \ldots$ цепочка $\varphi_n = z_1 y_1^n x y_2^n z_2$ принадлежит Π_T. Очевидно, $y_1 x y_2$ не может содержать целиком ни одного из отрезков $d w_i d$ и поэтому содержится либо в одном таком отрезке, либо в двух соседних.

Заметим прежде всего, что ни y_1, ни y_2 не могут содержать d. В самом деле, пусть, например, y_1 содержит d. Тогда возможен один из четырех случаев : I/ $y_1 = d$; 2/ $y_1 = d w'$, где w' – непустое начало некоторого отрезка w_i ; 3/ $y_1 = w'' d$, где w'' – непустой конец некоторого отрезка w_i ; 4/ $y_1 = w'' d d w'$, где w'' – конец отрезка w_{i-1} и w' – начало отрезка w_i (w'' и w' могут быть и пустыми). В случае I/ цепочка φ_3 содержит подцепочку $y_1^3 = d^3$ и поэтому не может быть протоколом. В случае 2/ цепочка φ_2 содержит подцепочку $y_1^2 = d w' d w'$, но никакой протокол не содержит подцепочек вида $\varphi d \psi$, где φ и ψ –непустые цепочки над $\{a_0, a_1, q\}$. Случай 3/ аналогичен 2/. Наконец, в случае 4/ цепочка φ_2 содержит подцепочку

$$d w_{i-1} d d w' y_1 = d w_{i-1} d d w' w'' d d w' ;$$

при этом $\ell'(w_i) > 2 \max(r, s)$ и $\ell'(w' w'') \leqslant$

$\leqslant \ell(w' w'') < 2 \max(r, s)$.

Но во всяком протоколе любая подцепочка вида $d w_{j-1} d d w_j d$, где w_{j-1}, w_j – цепочки над $\{a_0, a_1, q\}$, должна быть такова, что либо $\ell'(w_j) = \ell'(w_{j-1})$, либо $\ell'(w_j) = \ell'(w_{j-1}) + 1$. Поэтому φ_2 не может быть протоколом. Итак, y_1 не может содержать d. Аналогично для y_2.

Теперь остаются две возможности: либо $y_1 x y_2$ содержится целиком в некотором отрезке w_i, либо y_1 содержится в w_{i-1}, а y_2 – в w_i.
При этом ни y_1, ни y_2 не может содержать q – в самом

деле, в противном случае при любом n цепочка φ_n содержала бы вхождение подцепочки вида $d \omega d$, где ω –цепочка над $\{a_0, a_1, q\}$, содержащая не менее n вхождений q ; но в любом протоколе машины T число вхождений q между двумя d не может быть больше увеличенного на единицу числа состояний машины.

Пусть для определенности цепочка y_1 непуста. Рассмотрим два случая : I/ y_1 содержится в ω_i, где $i > 0$: $\omega_i = \omega_i' y_1 \omega_i''$. Тогда цепочка φ_3 будет содержать подцепочку $d \omega_{i-1} d d \omega_i' y_1^3 \omega_i'' d$; но такая подцепочка не может содержаться в протоколе, т.к. $\ell'(\omega_i' y_1^3 \omega_i'') \geqslant \geqslant \ell'(\omega_i) + 2 \geqslant \ell'(\omega_{i-1}) + 2$.

2/ y_1 содержится в ω_0 : $\omega_0 = \omega_0' y_1 \omega_0''$. Тогда цепочка φ_4 будет содержать подцепочку $d \omega_0' y_1^4 \omega_0'' d$; а такая подцепочка не может содержаться в протоколе, т.к.

$$\ell(\omega_2) \leqslant \ell(\omega_0) + 2 < \ell(\omega_0' y_1^4 \omega_0'') .$$

Доказательство закончено.

Рассмотрим теперь следующие свойства языков: $\theta_{K\mathcal{D}}$ – иметь конечное дополнение (здесь и далее подразумевается дополнение до $F'(W)$) ; $\theta_{\mathcal{D}}^{(n)}$ – иметь дополнение, состоящее точно из n элементов $(n = 0, 1, \ldots)$; в частности, $\theta_{\mathcal{D}}^{(0)}$ – совпадение с $F'(W)$) .

$\theta_{KC\mathcal{D}}$ – иметь контекстно-свободное дополнение ;

θ_A – быть а - языком.

<u>Теорема 5.8.</u> Ни одно из свойств $\theta_{K\mathcal{D}}$, $\theta_{\mathcal{D}}^{(n)}$ (при любом $n = 0, 1, \ldots\ldots$), $\theta_{KC\mathcal{D}}$, θ_A не распознаваемо в классе кс-языков.

<u>Доказательство</u>. а/ Очевидно, язык Π_T тогда и только тогда конечен, когда функция f_T имеет конечную область определенности. Если бы существовал алгоритм, распознающий по

кс-грамматике конечность дополнения к порождаемому ею языку, то мы могли бы для произвольной машины Тьюринга T построить по лемме 5.7 кс-грамматику, порождающую Π_T, а затем распознать с помощью указанного алгоритма, является ли Π_T конечным множеством; тем самым мы узнали бы, является ли область определенности f_T конечной. Следовательно, мы получили бы алгоритм, позволяющий по машине T распознавать конечность области определенности функции f_T; а существование такого алгоритма противоречит теореме Райса.

б/ Язык Π_T состоит из n элементов тогда и только тогда, когда f_T определена точно для n цепочек. Поэтому, если бы свойство $\theta_{\mathcal{D}}^{(n)}$ было распознаваемо в классе кс-грамматик, то мы получили бы-аналогично предыдущему пункту - алгоритм, позволяющий по машине T распознавать, состоит ли область определенности функции f_T точно из n элементов.

в/ Прежде чем доказывать нераспознаваемость свойств $\theta_{KC\mathcal{D}}$ и θ_A, мы заметим, что для всякой машины Тьюринга T легко построить другую машину T', вычисляющую ту же одноместную функцию f_T и обладающую тем свойством, что каждый ее протокол содержит не менее трех образов машинных цепочек. Такая машина получится, например, если присоединить к внутреннему словарю машины T новое состояние q_0', об"явив его заключительным вместо q_0, а к программе добавить команду $q_0 a_0 \to q_0' a_0$.

Допустим теперь, что хотя бы одно из свойств $\theta_{KC\mathcal{D}}$ и θ_A распознаваемо в классе кс-грамматик. Рассмотрим произвольную машину Тьюринга T и построим для нее машину T' только что указанным способом. Если функция $f_{T'} = f_T$ имеет бесконечную область определенности, то по лемме 5.8 язык $C\Pi'_{T'} = \Pi_T$ не является кс-языком и, следовательно, $\Pi'_{T'}$ не является а-языком. Если же область определенности функции f_T конечна, то язык Π_T также конечен и, следовательно, является a-языком; язык $\Pi'_{T'}$ в этом случае, очевидно, автоматный. Поэтому, если бы мы умели распознавать по кс-грамматике, является ли порождаемый ею язык а-языком или является ли его дополнение кс-языком, то мы смогли бы распознавать по машине Тьюринга T, имеет ли функция f_T конечную область определенности. Доказательство закончено.

Следствие. Не существует алгоритмов, позволяющих по любой паре кс-грамматик Γ_1 и Γ_2 узнавать, являются ли эти грамматики эквивалентными и содержится ли язык $L(\Gamma_1)$ в языке $L(\Gamma_2)$.

Доказательство. Пусть Γ_0 - а-грамматика, порождающая язык $F'(W)$. Тогда для любой кс-грамматики Γ соотношения $L(\Gamma_0) = L(\Gamma)_{(0)}$ и $L(\Gamma_0) \subseteq L(\Gamma)$ равносильны свойству $\theta_{\mathcal{D}}$. Поэтому не существует алгоритмов, позволяющих по любой кс-грамматике Γ узнавать, эквивалентна ли грамматика Γ грамматике Γ_0 и содержится ли язык $L(\Gamma_0)$ в языке $L(\Gamma)$. Тем более не существует общих алгоритмов, указанных в формулировке следствия.

Заметим, что для а- грамматик оба указанных алгоритма существуют. Свойства $\theta_{K\mathcal{D}}$ и $\theta_{\mathcal{D}}^{(n)}$ (для любого $n = 0, 1, \ldots$) также распознаваемы в классе а-грамматик. Соответствующие алгоритмы легко получаются с помощью теоремы 3.II.

ПРИЛОЖЕНИЕ: ВАРИАНТЫ И ПЕРЕВОДЫ ТЕРМИНОВ

Ниже приводятся английские эквиваленты некоторых наиболее важных терминов (часть из них в нескольких вариантах), а также встречающиеся в литературе параллельные русские термины.

Автоматная грамматика - finite state grammar, грамматика с конечным числом состояний.

Автоматный язык (а-язык) - finite state language, regular set, regular event, язык с конечным числом состояний, регулярное множество, регулярное событие.

Взаимозамещаемость - E - эквивалентность, синтаксическая эквивалентность.

Вспомогательный символ - nonterminal symbol, нетерминальный символ.

Вспомогательный словарь - nonterminal vocabulary, нетерминальный словарь.

Вывод - derivation, деривация.

Грамматика непосредственно составляющих - phrase - structure grammar, context - dependent grammar, context - sensitive grammar, фразово-структурная грамматика, грамматика фразовых структур.

Категориальная грамматика - categorial grammar.

Контекстно-свободная грамматика - context - free grammar, simple phrase - structure grammar, бесконтекстная грамматика.

Контекстно-свободный язык (кс- язык) - context-free language, simple phrase - structure language, definable set, бесконтекстный язык.

Нс- язык - phrase - structure language.

Основной символ- terminal symbol, терминальный символ.

Основной словарь - terminal vocabulary, терминальный словарь.

Подчинение - непосредственная доминация, управление.

Правило грамматики - rule of grammar, rewriting rule, правило переписывания.

Система составляющих - phrase - structure system.

Словарь - vocabulary, alphabet, алфавит.

Составляющая - phrase

Цепочка - string, фраза.

Циклический символ- recursive symbol, рекурсивный символ.

Элементарная категория - primitive category.

ЛИТЕРАТУРА

К главе I.

Все необходимые сведения из теории графов имеются в книге:

[1] К.Берж, Теория графов и ее применения, М.,1962.

Неформальное, с лингвистической интерпретацией, изложение понятий, содержащихся в гл.I, имеется в статье:

[2] Е.В.Падучева, О способах представления синтаксической структуры предложения. Вопросы языкознания, 1964, № 2.

О значении понятий проективности и сильной проективности для исследования естественных языков см.

[3] Л.Н.Иорданская, О некоторых свойствах правильной синтаксической структуры. Вопросы языкознания,1963, № 4.

Отношениям подчинения посвящены работы:

[4] М.И.Белецкий,В.М.Григорян, И.Д.Заславский, Аксиоматическое описание порядка и управления слов в некоторых типах предложений. В сборнике "Математические вопросы кибернетики и вычислительной техники", № I, Ереван,1963.

[5] С.Я.Фитиалов, О моделировании синтаксиса в структурной лингвистике. В сборнике "Проблемы структурной лингвистики", М.1962.

Связь между отношениями подчинения и системами составляющих рассматривается в работе:

[6] C. Gaifman, Dependency systems and phrase-structure systems. – Inform. and Control, 1965, 8, N 3, pp. 304-337.

Пример непроективного фрагмента естественного языка указан в работе:

[7] P. M. Postal, On the limitations of context-free phrase-structure description. – Quarterly progress report, N 64, Jan. 1962, MIT, Research Lab. of Electronics, Cambridge (Mass.), 231-238.

К главе II.

Все необходимые сведения из алгебры имеются в книге:

[8] А.Г.Курош, Лекции по общей алгебре, М.,1962.

Основные понятия теории замещаемости были впервые введены в работе:

[9] О.С.Кулагина, Об одном способе определения грамматических понятий на базе теории множеств. - Проблемы кибернетики, вып.I.,М.,1958.

Понятие конфигурации рассматривается с точки зрения лингвистической интерпретации в статье:

[10] Е.В.Падучева, О понятии конфигурации.- Вопросы языкознания,1965, № I.

Приложения теории замещаемости к изучению естественных языков подробно рассмотрены в книге:

[11] И.И.Ревзин, Метод моделирования и типология славянских языков (в печати).

К главе III.

Изложение теории порождающих грамматик имеется в работе:

[12] N. Chomsky, Formal properties of grammars. – В книге: R. R. Bush, E. H. Galanter, R. Luce (eds), Handbook of Math. Psychology, vol. 2, New York, 1963.

(В"Кибернетическом сборнике" должен появиться русский перевод).

Лингвистические аспекты теории порождающих грамматик рассматриваются в работе:

[13] Н.Хомский, Синтаксические структуры.- В сборнике "Новое в лингвистике", вып.2,М.,1962.

Контекстно-свободным грамматикам посвящены работы:

[14] Y. Bar-Hillel, M. Perles, E. Shamir, On formal properties of simple phrase structure grammars. – Z. für Phonetik, Sprachwissenschaft und Kommunikationsforschung, 1961, 14, 2, S. 143-172.

[15] N. Chomsky, M.-P. Schützenberger, The algebraic theory of context-free languages. - В книге: P. Braffort, D. Hirschberg (eds), Computer programming and formal systems, Amsterdam, 1963
(В "Кибернетическом сборнике" должен появиться русский перевод).

Автоматным грамматикам посвящена работа:

[16] Н.Хомский, Д.Миллер, Языки с конечным числом состояний.- Кибернетический сборник, вып.4, М.,1962.

К главе IУ.

Основная теорема о связи между категориальными и контекстно-свободными грамматиками доказана в статье:

[17] Y. Bar-Hillel, C. Gaifman, E. Shamir, On categorial and phrase-structure grammars. - Bull. of the Research Council of Israel, 1960, 9F, pp. 1-16.

Различные математические и лингвистические аспекты теории категориальных грамматик рассматриваются в работе

[18] И.Ламбек, Математическое исследование структуры предложения.- В сборнике "Математическая лингвистика", М.,1964.

К главе У.

Все необходимые сведения из теории алгоритмов имеются в книге

[19] А.И.Мальцев, Алгоритмы и рекурсивные функции,М.,1965.

Вопрос о классификации примитивно рекурсивных функций по сложности вычисления рассматривается в статье

[20] R.W. Ritchie, Classes of predictably computable functions. - Trans. Amer. Math. Soc., 1963, 106, pp. 109-163.

Различные алгоритмические проблемы для грамматик рассматриваются, в частности, в работах:

[21] P.S. Landweber, Decision problems of phrase-structure grammars. - IEEE Trans. on Electronic Computers, 1964, EC-13, 4, pp. 354-362.

[22] А.В.Гладкий, Алгоритмическая природа инвариантных свойств грамматик непосредственно составляющих.- Алгебра и Логика. Семинар, 1964,3,2.

[23] А.В.Гладкий, Некоторые алгоритмические проблемы для контекстно-свободных грамматик.- Алгебра и Логика. Семинар,1965, 4,1.

[24] А.В.Гладкий, Алгоритмическая нераспознаваемость существенной неопределенности контекстно-свободных языков.- Алгебра и Логика. Семинар,1965,4,4.

Алгоритмические проблемы теории грамматик рассматриваются также в [12], [14] и [15].

ОГЛАВЛЕНИЕ

стр.

Введение 1

Глава I

Представление синтаксической структуры предложения

§ I. Цепочки, подцепочки, вхождения 8

§ 2. Системы составляющих 10

§ 3. Отношения подчинения 14

§ 4. Связь между системами составляющих и отношениями подчинения 22

Глава П.

Замещаемость

§ 5. Два способа описания языка 29

§ 6. Понятия замещаемости, взаимозамещаемости и конфигурации 29

§ 7. Конфигурационные характеристики..................... 32

§ 8. Укрупнения эквивалентностей 34

§ 9. Окрестности, классы и типы 38

§ I0. Пример 42

Глава Ш.

Порождающие грамматики

§ II. Основные определения 49

§ I2. Абстрактные примеры 55

§ I3. Лингвистические примеры 63

§ I4. Обобщенные нс-грамматики 67

§ I5. Неукорачивающие грамматики........................ 72

§ I6. Грамматики без существенного укорачивания......... 74

§ I7. Некоторые вспомогательные утверждения о контекстно-свободных грамматиках.................................. 80

§ I8. Распознавание пустоты и конечности кс-языка....... 85

§ I9. Необходимое условие контекстной свободности языка.. 89

§ 20. Конечно характеризуемые языки и кс-языки.......... 92

§ 2I. Автоматные грамматики 94

§ 22. Теорема о конгруентностях конечного индекса....... 99

§ 23. Операции над языками..............................103

Глава IУ.

Категориальные грамматики

§ 24. Основные определения 110

§ 25. Примеры 113

§ 26. Односторонние монотонные грамматики 120

§ 27. Связь между категориальными и контекстно-свободными грамматиками 123

§ 28. Категориальные грамматики и системы составляющих...132

Глава У.

Некоторые алгоритмические вопросы теории порождающих грамматик

§ 29. Алгоритмическая природа языков, порождаемых произвольными грамматиками...............................140

§ 30. Алгоритмическая природа нс-языков 145

§ 31. Проблемы распознавания свойств языков, порождаемых грамматиками.......................................159

§ 32. Проблемы распознавания свойств нс-языков..........163

§ 33. Проблемы распознавания свойств кс-языков 171

Приложение: варианты и переводы терминов............182

Литература 184

ГЛАДКИЙ АЛЕКСЕЙ ВСЕВОЛОДОВИЧ

Лекции по математической лингвистике

Ответственный за выпуск

Т.И. Шедько.

Подписано в печать ______ МИ 04202

Формат бумаги 60 x 84 I/16 Объем II,8 п.л.

Заказ № 400 Тираж 350.

Цена 36 к.

Ротапринт Новосибирский госуниверситет.