М. И. Шлезингер

МАТЕМАТИЧЕСКИЕ СРЕДСТВА ОБРАБОТКИ ИЗОБРАЖЕНИЙ

УДК 519.72

Математические средства обработки изображений / Шлезингер М. И. Отв. ред. Скурихин В. И.; АН УССР. Ин-т кибернетики.— Киев: Наук. думка, 1989.— 200 с.— ISBN 5-12-000546-2.

В монографии рассмотрены математические средства, предназначенные для описания, экономного хранения и обработки множеств изображений. Этими средствами являются двумерные грамматики — формализм, подобный известным одномерным грамматикам, но учитывающий двумерный характер порождаемых объектов.

На основании представления изображений двумерными грамматиками предложена единая формулировка для таких задач обработки и распознавания изображений, которые ранее представлялись существенно различными. Исследована вычислительная сложность сформулированной задачи в ее общей постановке.

Для специалистов, занимающихся вопросами теоретической кибернетики, обработки изображений, а также пользователей систем обработки изображений.

Ил. 39. Табл. 1. Библиогр.: с. 190—197 (140 назв).

Ответственный редактор *В. И. Скурихин*

Утверждено к печати ученым советом
Института кибернетики им. В. М. Глушкова АН УССР

Редакция физико-математической литературы
Редактор *В. П. Егорова*

Ш $\frac{1402010000-89}{М221(04)-133}$ 169-89

ISBN 5-12-000546-2

ПРЕДИСЛОВИЕ

Кто-то сказал, что легче иметь дело с человеком, который не читает книг вообще, чем с человеком, который прочитал одну книгу. Сейчас мало кто помнит (автор данной монографии уж точно не помнит), кто написал первую книгу по распознаванию образов, кто — вторую и т. д., но, конечно, труднее всего пришлось автору второй книги, потому что именно он вынужден был доказывать, зачем нужна еще одна книга по распознаванию. Сейчас, когда только на русском языке опубликованы десятки работ по распознаванию, вопрос о целесообразности еще одной даже не возникает, так как стало ясно, что распознавание образов — это океан нашего незнания, и даже десятки монографий представляются разбросанными в этом океане немногочисленными островками.

Как правило, каждый такой островок знаний вырастает из опыта решения определенного класса прикладных задач, которые оказывают достаточно сильное влияние на характер полученных знаний. В данной монографии описываются знания о распознавании, приобретенные автором при решении прикладных задач распознавания чертежно-графических и текстовых изображений. Описание этих задач в их прикладном аспекте приведено в приложении. В основное же содержание книги включены лишь те результаты, которые удалось представить математически, т. е. в виде теорем и решения формально поставленных задач. Взаимосвязь описываемой теории и прикладных задач не следует, однако, воспринимать слишком прямолинейно. Некоторые результаты, изложенные в монографии, действительно получены потому, что они были необходимы для достижения ясно осознанных прикладных целей, и эти цели упорно не достигались без их теоретического осмысления. В таком режиме разработаны алгоритмы синтеза решающих правил, названные по традиции обучением, алгоритмы вычеркивания и двумерного программирования. Другие результаты получены в менее форсированном режиме, когда работоспособные алгоритмы уже применялись на практике, а теоретический их анализ и формализация решаемых задач носили апостериорный характер. Таков в основном весь материал гл. 2, в частности исследования методов экономного кодирования графических изображений.

В монографии описаны и те исследования, выполнение которых не стимулировалось какими-либо ближними прикладными целями. Известно, что абстрактные математические понятия, первоначально введенные для формализации реальных объектов, обретают затем способность жить своей собственной жизнью, и исследование таких чисто формальных свойств введенных понятий только на первый взгляд не имеет отношения к практике. Иногда (правда, реже, чем хотелось бы) полученные таким путем результаты допускают деформализацию, т. е. обратное проецирование на прикладную область, где они приводят к совершенно неожиданным рекомендациям. К этому классу результатов относятся исследования по общей двумерно-грамматической задаче и степени сложности графов (описанные в конце гл. 3) и почти вся теория самообучения (гл. 4).

И, наконец, о последнем классе полученных результатов. Читатель, несомненно, заметит, что некоторые рекомендации автора являются далеко не простыми. В этой ситуации неизбежно возникает вопрос: а нельзя ли проще? Этот вопрос неоднократно ставил перед собой и автор, тем более, что в процессе работы слишком часто открывались пути, на первый взгляд более легкие, а в действительности тупиковые. Они приводили к множеству отрицательных результатов, и некоторые из них включены в книгу, в частности критический анализ локально-конъюнктивных предикатов (гл. 1). Кроме того, получение любых новых результатов невозможно без определенного пересмотра общепринятых представлений и их критической переоценки. Этот пересмотр, проведенный во введении в намеренно резкой форме, следует понимать не более чем приглашение к спору.

Само собой разумеется, что рекомендации, которые при теоретическом рассмотрении даже очень убедительны, требуют чрезвычайно больших усилий и изобретательности при их практическом использовании. Поэтому автор искренне благодарен своим товарищам по работе В. М. Кийко, В. В. Мацелло, Ю. Л. Провалову и В. М. Шарыпанову, сумевшим отдельные результаты этой монографии довести до уровня практически действующих комплексов по обработке чертежно-графических изображений. Конечно, в этих комплексах реализованы и их собственные плодотворные идеи, далеко выходящие за рамки материала монографии. Автор признателен также замечательному коллективу разработчиков читающих автоматов СКБ математических машин и систем во главе с А. Г. Семеновским и А. Ф. Возияновым. В этом коллективе идеи обучения были доведены до конкретных технологий машинного синтеза и настройки распознающих устройств. Автор благодарен также Т. Н. Барабанюк, чья дружеская помощь была необходима на протяжении всего периода работы над монографией.

ВВЕДЕНИЕ

Заслуживает особого внимания тот факт, что распознавание образов, зарождение которого стимулировалось необходимостью решать задачи зрительного анализа, сейчас большей частью не имеет прямого отношения к этим задачам. Первоначальные публикации таких широко известных методов распознавания, как метод обобщенного портрета [19], потенциальных функций [5], алгоритмы М. Бонгарда [17] и Ф. Розенблатта [89] более или менее явно были ориентированы именно на задачи зрительного анализа. Сейчас эти алгоритмы успешно применяются для решения задач медицинской диагностики [29], предсказания долговечности приборов [77], различения нефтеносных и водоносных пластов [33]. Область применения этих алгоритмов, по-видимому, включает и многие другие практически важные задачи, однако распознавание изображений находится вне этой области. Ощутимые практические достижения как в советских [3, 14, 16, 68], так и в зарубежных [28, 83] системах распознавания изображений обусловлены в значительно меньшей степени рекомендациями общей теории, чем учетом специфических особенностей каждой отдельной прикладной задачи. Это свидетельствует об определенном разрыве между теорией распознавания образов в том виде, в каком она сформировалась к настоящему времени, и существующей практикой распознавания изображений.

Мы укажем на основные ключевые понятия современной теории распознавания образов, которые обусловливают ее несоответствие требованиям распознавания изображений. Фундаментальные результаты теории распознавания образов основаны на использовании следующих формальных понятий. Прежде всего вводится в рассмотрение конечное множество T, называемое множеством признаков, и множество V значений признаков. Множество V^T функций вида $T \to V$ называется множеством $\mathcal{V}$ распознаваемых объектов, а одна отдельная функция $v: T \to V$ — распознаваемым объектом. Чаще всего, хотя и не всегда, предполагается, что V — это множество вещественных чисел либо множество $\{0, 1\}$. В этих случаях множество $\mathcal{V}$ является n-мерным пространством R^n, где $n = |T|$, либо множеством вершин n-мерного единичного куба. На множестве $\mathcal{V}$ предполагается существование функции $k: \mathcal{V} \to \mathcal{K}$, где $\mathcal{K}$ — множество, состоящее из небольшого количества элементов, в частности двух элементов. Множество $\mathcal{K}$ называется множеством классов распознаваемых объектов, а для любого $k^* \in \mathcal{K}$ множество $\{v : v \in \mathcal{V},\ k(v) = k^*\}$ называется классом объектов с именем k^*. Функция $k: \mathcal{V} \to \mathcal{K}$, как правило, неизвестна, однако предполагается известным множество $\bar{k}$ функций, содержащее k, а также сужение k на некоторое конечное подмножество $\hat{v} \subset \mathcal{V}$ объектов, называемое обучающим подмножеством. Обозначим это сужение через $k(\hat{v})$, т. е. $k(\hat{v})$ — это функция $\hat{v} \to \mathcal{K}$, такая что $k(\hat{v})(v) = k(v)$ для всех $v \in \hat{v}$, и не определенная для $v \notin \hat{v}$.

Основное содержание распознавания образов заключается в том, чтобы по известному множеству $\bar{k}$, обучающему множеству $\hat{v}$ и функции $k(\hat{v})$ принять разумное в определенном смысле решение о значении $k(v)$ функции k для некоторого объекта v, не принадлежащего $\hat{v}$.

Все разнообразие известных методов и направлений в современном распознавании образов заключается в той или иной формализации указанного неформального требования. Наиболее глубокие и вместе с тем обширные исследования здесь выполнены в школах, руководимых Ю. И. Журавлевым [37—43, 70, 71, 90], В. Н. Вапником [7, 8, 19—21], Ш. Ю. Раудисом [84, 88], Н. Г. Загоруйко [44—47, 66], В. А. Якубовичем [123, 124], и в более ранних работах М. А. Айзермана, Э. М. Бравермана, Л. И. Розоноэра [6], А. Г. Ивахненко [51], Ф. Розенблатта [89], Я. З. Цыпкина [104], К. С. Фу [100] и др. Совокупность указанных работ сформировала современную теорию обучения распознавания образов, плодотворность которой уже прошла длительную и многостороннюю проверку на большом количестве прикладных задач медицины, геологии, социологии и других нематематизированных наук.

Основная мысль, которая прослеживается в данной монографии, заключается в том, что несоответствие теории распознавания образов и практики распознавания изображений обусловлено не каким-либо изъяном в формализации процесса обучения, а определяется неадекватностью более глубоких

понятий, принимаемых как само собой разумеющееся еще до этой формализации. Рассмотрим отдельные стороны этого несоответствия более подробно.

1. Представление распознаваемого объекта в виде многомерного вектора. Изображение в своем первичном представлении является функцией ρ, определенной на прямоугольной области P в плоскости (x, y) и принимающей значения из множества вещественных чисел. Вторичное представление изображения получают с помощью операции дискретизации. Это значит, что прямоугольник P разбивают на некоторые участки — клетки и для каждой клетки измеряют среднее значение функции ρ в этой клетке. Множество T клеток называется полем зрения, а количество $|T|$ клеток — размером поля зрения. В результате дискретизации изображение представлено функцией $v: T \to R$, а множество изображений отождествляется с $|T|$-мерным пространством $R^{|T|}$. Каждому классу изображений соответствует определенное подмножество в $R^{|T|}$ (рис. В.1). В случае когда этих классов два, задача распознавания может пониматься [6, 79, 93] как нахождение уравнения границы, разделяющей эти два подмножества, т. е. отыскание функции $f: R^{|T|} \to R$, такой, что $f(v) = 0$ есть уравнение границы, $f(v) > 0$ — условие принадлежности изображения v к одному классу, а $f(v) < 0$ — к другому. Задача распознавания в такой формализации, казалось бы, допускает достаточно наглядную интерпретацию.

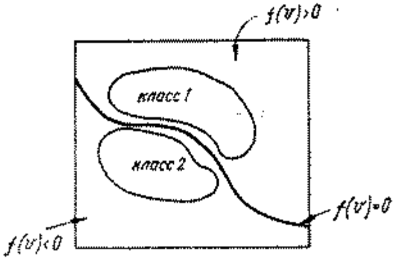


Рис. В.1. Геометрическая интерпретация задачи распознавания

Потребовалось довольно много усилий и времени, чтобы убедиться в существенном изъяне такой формализации, который заключается в том, что интуитивное понимание сложности того или иного класса изображений совершенно не совпадает со сложностью представления соответствующего ему подмножества в $R^{|T|}$. Таким простейшим зрительным понятиям, как, например, квадрат, окружность или треугольник соответствуют вовсе непростые подмножества в $R^{|T|}$, каковыми считаются гиперсфера, подпространство, пересечение полупространств и т. п. (см., например, убедительные доводы по этому поводу В. А. Ковалевского [61]). Поэтому использование многомерных геометрических аналогий в распознавании изображений не оказалось столь плодотворным, как, скажем, в линейном программировании, где основному понятию — системе линейных неравенств — соответствует очень наглядный геометрический объект — выпуклый многогранник. Кроме того, и части этого объекта (грани, ребра, вершина) также допускают правильную интерпретацию. Понимание чисто геометрических отношений между этими частями (например, что из любой вершины можно попасть в любую другую вершину, перемещаясь только по ребрам) облегчает правдоподобные рассуждения исследователя, т. е. позволяет убеждаться в правильности той или иной теоремы еще до того, как получено ее формальное доказательство. В обработке же изображений подобные геометрические аналогии оказались не только неплодотворными, а прямо-таки дезориентирующими.

Известно достаточно резкое высказывание М. Минского и С. Пэйперта [76], что ничто не нанесло такого большого вреда обработке изображений, как эксплуатация многомерных геометрических аналогий. Хотя такая резкость была вполне уместной в пору всеобщего увлечения этими аналогиями, сейчас не хотелось бы безоговорочно соглашаться с данным высказыванием, потому что ясное понимание неприемлемости геометрического подхода для обработки изображений должно сочетаться с не менее ясным пониманием его плодотворности для ряда незрительных задач. Тот факт, что геометрический подход оказался плодотворным для очень многих задач, но только не для зрительных, свидетельствует об исключительном характере изображения как объекта распознавания. Эта исключительность заключается в том, что множество T в зрительных задачах является не просто абстрактным конечным множеством, а множеством клеток с естественной и вполне определенной структурой, которая позволяет вводить такие отношения на этом множестве, как соседство, близость и т. п. Основные зрительные понятия, такие как связность, симметрия, размер, ориентация, просто немыслимы вне этих свойств. Важность структуры множества T для задач зрительного анализа является настолько очевидной, что можно только удивляться тому оптимизму, который позволял надеяться на получение содержательных результатов с помощью формализмов, не учитывающих эту структуру. Что же касается задач незрительного распознавания, то здесь игнорирование структуры множества T признаков (именно признаков, а не их числовых значений!), по-видимому, не имело столь деструктивного характера.

Из сказанного следует, что одно из несоответствий теории распознавания образов и практики распознавания изображений заключается в упрощенном представлении множества T признаков, при котором не учитываются свойства этого множества, существенные для правильного понимания задач распознавания изображений.

2. Классификация изображений. Представление о том, что результатом распознавания какого-либо объекта, явления или изображения является отнесение его к тому или иному классу, следует считать предельным упрощением того, что должно пониматься под распознаванием. Такое упрощенное рассмотрение задачи распознавания является лишь предварительным этапом при решении прикладной задачи, хотя и важным, но не исчерпывающим всей задачи.

Допустим, например, что распознаваемым объектом является исходный текст программы на некотором формальном языке, а задача распознавания заключается в определении синтаксической правильности этого текста. Допустим также, что существует алгоритм, безошибочно решающий эту задачу, т. е. безошибочно осуществляющий классификацию всех текстов на два класса. Если сейчас достаточно ясно представить, что этот алгоритм всего лишь фиксирует неправильность исходного текста, пусть безошибочно, но без каких-либо указаний, характеризующих эту неправильность, то становится очевидной практическая бесполезность такого алгоритма.

Приведенный пример не является типичным для распознавания образов, так как автоматический анализ программ, насколько известно, не входит в сферу применимости распознавания образов. Однако задачи технической или медицинской диагностики, анализ геологоразведочных данных и т. п. являются уже традиционной сферой применения распознавания образов, и можно допустить, что в этих областях классификация распознаваемых объектов является также лишь паллиативом по отношению к действительной прикладной задаче, в которой требуется определить значение скрытых, непосредственно не измеряемых параметров, характеризующих объект и обусловливающих те или иные значения измеряемых параметров. Может быть это и спорно, так как речь идет о незрительных объектах. Однако опыт автоматического распознавания изображений, т. е. космических снимков [28, 78], треков микрочастиц [96], чертежно-графической информации [12, 13] и машинописных текстов [103] позволяет утверждать, что ни одна из перечисленных задач не сводилась к отображению множества распознаваемых объектов в множество, состоящее из небольшого количества классов. В зависимости от характера прикладной задачи при распознавании изображения должна быть получена либо некоторая его числовая характеристика, либо набор таких характеристик, либо наименование его частей с указанием тех или иных соотношений между ними, либо, наконец, просто другое изображение. Процесс получения такого рода результатов трудно и неуместно интерпретировать как классификацию изображения, хотя отдельные этапы этого процесса и сходны с классификацией.

Таким образом, второе из несоответствий теории распознавания образов и практики распознавания изображений заключается в упрощенном представлении множества $\mathcal{K}$ результатов распознавания, при котором невозможна адекватная формализация прикладных задач.

3. О классе алгоритмов распознавания и роли обучения. Распознающий алгоритм предназначен для реализации отображения $k : \mathcal{V} \to \mathcal{K}$, где $\mathcal{V}$ — множество распознаваемых объектов, а $\mathcal{K}$ — множество результатов распознавания. Из предыдущего рассмотрения следует, что для распознавания изображений оба эти множества должны быть формализованы значительно более тщательно, чем это сделано к настоящему времени.

Не менее важной является формализация множества $\bar{k}$ функций k, которые уместно считать распознаванием изображений. Это множество должно быть задано хотя бы потому, что оно необходимо при любой разумной формулировке задачи обучения, но не только поэтому.

Своеобразие задач зрительного распознавания заключается в том, что даже если функция $k : \mathcal{V} \to \mathcal{K}$ вполне однозначно и недвусмысленно задана, построение алгоритма для ее вычисления требует решения далеко не тривиальной задачи. В распознавании незрительных объектов подобная ситуация не возникает, т. е. обычно рассматриваются лишь такие функции $k : \mathcal{V} \to \mathcal{K}$, при которых проблемы их вычисления просто не возникают. Основные результаты здесь относятся к анализу алгоритмов обучения, т. е. алгоритмов нахождения этой априори неизвестной функции, что способствовало распространению представления будто проблемой обучения исчерпывается вся трудность распознавания. Такое представление, несмотря на свою распространенность, глубоко ошибочно и следует из непонимания фундаментального различия между однозначным заданием функции, отношения, множества и алгоритмом вычисления функции, проверки отношения или принадлежности к множеству.

Распознавание принадлежности цепочки к тому или иному языку, определение противоречивости отношений, проверка эквивалентности отношений — все это типичные для математических исследований и далеко не тривиальные задачи. Они формулируются именно как распознавание принадлежности вполне определенного математического объекта недвусмысленно заданному множеству таких объектов. Задачи распознавания изображений принадлежат именно к таким математическим задачам, и их трудность заключается не в том, что какие-то множества или функции неизвестны, а в том, что эти функции или множества, уместные в распознавании изображений, не входят ни в какой хорошо изученный к настоящему времени класс математических объектов. Поэтому каждую отдельную задачу распознавания изображений решают как бы заново, начиная с нуля, а не на основе знания общего вида этих задач.

Потребность в автоматическом распознавании изображений, которую испытывают сейчас практически все отрасли народного хозяйства, имеет настолько разнообразный и массовый характер, что удовлетворить ее невозможно при указанном разрыве между теорией и практикой. Поэтому если в период зарождения распознавания образов актуальным являлось исследование наиболее общих характеристик распознающих алгоритмов, не зависящих от свойств множеств $\mathcal{K}$ классов и признаков T, и построение алгоритмов, работоспособных при как можно меньших априорных сведениях, то сейчас существует настоятельная необходимость развития этой теории в направлении, допускающем такое формальное задание всех априорных сведений о классе распознаваемых объектов, чтобы и алгоритм распознавания формально следовал из этого задания.

Описанный в монографии аппарат двумерных грамматик является математическим средством, позволяющим формально задавать множества распознаваемых изображений и на основании этого задания формулировать и решать задачи распознавания.

ОГЛАВЛЕНИЕ

Глава 1

ДВУМЕРНЫЕ ГРАММАТИКИ

Конечно, прав был П. Халмош [102], остроумно отметивший, что наилучшим стимулом для выполнения хорошей работы является выполненная кем-то другим не совсем хорошая работа. В настоящее время распознавание образов не испытывает недостатка в такого рода исследованиях, которые уже оказали или еще окажут на него мощное стимулирующее воздействие в том смысле, каком это имел в виду П. Халмош. Экстремальное положение среди таких работ, вне всякого сомнения, занимает персептрон Ф. Розенблатта. Предполагаемые свойства персептрона были настолько привлекательными и настолько не совпадали с его действительными возможностями, что не могло не вызвать потока работ в данном направлении.

Персептрон впервые был подвергнут критическому анализу в работах В. М. Глушкова [31, 32]. В них исследованы алгоритмы обучения и самообучения в персептроне и показано, что многие положительные свойства персептрона оказались таковыми только на словах. Однако, как это не так уж редко бывает, понимание недостатков конкретных алгоритмов настройки персептрона не уменьшило, а увеличило интерес к этим алгоритмам, и идеи Ф. Розенблатта, несмотря на свою ошибочность, послужили толчком для зарождения целого плодотворного направления в распознавании образов, развитого в школах М. А. Айзермана [6] и В. А. Якубовича [123, 124].

Несколько лет спустя оказалось, что драматическая история персептрона не завершилась исследованием ошибок, заложенных в его алгоритмах обучения. Новое развитие этой истории дала монография М. Минского и С. Пэйперта [76], которая вскрыла более глубинные недостатки, относящиеся уже не к алгоритмам обучения, а к структуре персептрона. Было показано, что персептроны, реализуя только так называемые локально-конъюнктивные предикаты, не могут быть

настроены на решение некоторых задач, и эта ситуация не изменится ни при каком совершенствовании алгоритмов обучения. Понимание недостатков локально-конъюнктивных предикатов стимулирует разработку более совершенных формальных схем для анализа изображений. Одна из возможных таких схем — двумерная грамматика — и является основным объектом исследования в данной книге.

В этой главе прежде всего будет внесена полная ясность в вопрос о том, что можно выражать локально-конъюнктивными предикатами, поднятый М. Минским и С. Пэйпертом, т. е. будут сформулированы необходимые и достаточные условия, которым должны удовлетворять множества изображений, чтобы их можно было выразить предикатами такого типа. Будет показано, что контрпримерами для локально-конъюнктивных предикатов служат не какие-то экзотические, специально придуманные множества изображений, а самые обычные, в некотором смысле типичные зрительные понятия. Кроме того, будут определены дополнительные формальные средства, существенно расширяющие возможности задания множеств, но сохраняющие привлекательные свойства локально-конъюнктивных предикатов, состоящие в том, что ограничения на изображение в целом, т. е. глобальные ограничения, выражаются сугубо локальными средствами.

1.1. ЛОКАЛЬНО-КОНЪЮНКТИВНЫЕ ПРЕДИКАТЫ

Как в работе [76], дадим следующее определение локально-конъюнктивного предиката.

Пусть V — конечный алфавит сигналов, Tv — конечное множество, называемое полем зрения, элементы которого называются клетками. Изображением называется функция $v : Tv \to V$, фрагментом поля зрения — любое подмножество $T' \subset Tv$, структурой $\mathcal{T}$ поля зрения — любое подмножество фрагментов $T' \subset Tv$, $T' \neq Tv$, не равных полю зрения. Порядком структуры $\mathcal{T}$ называется число $\max\limits_{T' \in \mathcal{T}} | T' |$. Для любого изображения v и фрагмента T' через $v\,(T')$ обозначим функцию $T' \to V$, являющуюся сужением v на $T' \subset T$ и называемую фрагментом изображения v на фрагменте T'. Для любой клетки $t \in T$ через $v\,(t)$ обозначим значение изображения v для клетки t. Таким образом, $v\,(t) = v\,(\{t\})$, а $v = v\,(Tv)$.

Пусть $\{0, 1\}$ — множество, для которого определены как логические функции $\bigwedge$ и $\bigvee$, так и арифметические функции min, max и отношения $>$ и $<$.

Для любого фрагмента $T' \subset Tv$ через $F(T')$ обозначим функцию $V^{T'} \to \{0, 1\}$, называемую локальным предикатом допустимости фрагментов изображений на T'. Значение этой функции для фрагмента $v(T')$ обозначим через $F(T', v(T'))$. Функция $F(Tv)$ будет называться предикатом допустимости изображений.

Предикат $F(Tv)$ называется локально-конъюнктивным на структуре $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$, если существуют такие локальные предикаты $F(T')$, $T' \in \mathcal{T}$, что для любого изображения v справедливо равенство $F(Tv, v) = \bigwedge_{T' \in \mathcal{T}} F(T', v(T'))$.

Порядком предиката называется порядок структуры, на которой он является локально-конъюнктивным. Таким образом, предикат какого-нибудь, скажем, k-го, порядка является предикатом и l-го порядка для любого $l > k$.

Локально-конъюнктивные предикаты формализуют такое представление о классе распознаваемых объектов, при котором исходят из предположения, что существенные характеристики распознаваемых объектов выражаются функциями, зависящими от небольшого количества признаков из фиксированного, заранее заданного множества признаков. Такое представление находит свое крайнее выражение в работах (например, [46, 99]), где предполагается, что сам предикат допустимости есть функция небольшого множества признаков. Известны попытки обосновать такого рода предположения [46, 48], относящиеся к теории обнаружения закономерностей — одного из направлений распознавания. Мы рассмотрим два примера, которые могут служить оправданием для данного предположения (первый из примеров взят из монографии [76]). Затем выполним формальный анализ локально-конъюнктивных предикатов и убедимся, что возможность выражать с их помощью зрительные понятия является скорее исключением, чем правилом.

Примеры. 1. Чтобы не перегружать пример ненужными сейчас подробностями, рассмотрим в качестве поля зрения Tv не конечное множество, а множество точек на участке плоскости, ограниченной двумя парами параллельных прямых. Изображением будем считать функцию v, заданную на этом участке и принимающую два значения: 0 и 1 (белое и черное). Допустимыми будем считать только изображения выпуклых фигур, т. е. изображения, для которых множество $\{t : t \in Tv,\ v(t) = 1\}$ является выпуклым. Непосредственно из определения выпуклости следует, что предикат «выпуклость» локально-конъюнктивный третьего порядка. Действительно, для того чтобы проверить, является ли изображение в поле зрения выпуклой фигурой, достаточно просмотреть все

тройки точек поля зрения, расположенных на одной прямой, и проверить значение черноты в этих точках. Если во всех тройках, в которых две крайние точки черные, третья точка также является черной, то в поле зрения представлено изображение выпуклой фигуры. Если хотя бы в одной тройке две крайние точки черные, а третья белая, то фигура в поле зрения невыпуклая.

2. Для заданных целых чисел m и n полем зрения Tv будем считать множество $\{(i, j) : 1 \leqslant i \leqslant m,\ 1 \leqslant j \leqslant n\}$ пар чисел i и j. Изображением прямоугольника с координатами i_1, i_2, j_1 и j_2 будем считать функцию $Tv \to \{0, 1\}$, принимающую значение 1 на подмножестве клеток $\{(i, j) : (i, j) \in Tv,\ i_1 \leqslant i \leqslant i_2,\ j_1 \leqslant j \leqslant j_2\}$ и значение 0 за пределами этого множества. Допустимым будем считать любое изображение, на котором представлена совокупность прямоугольников, попарно не соприкасающихся (рис. 1.1). Очевидно, что такое множество допустимых изображений выражается предикатом, локально-конъюнктивным на структуре четвертого порядка, содержащей все четверки клеток вида $((i, j), (i + 1, j), (i, j + 1), (i + 1, j + 1))$. Менее очевидно, но все же верно, что в действительности этот предикат является предикатом третьего порядка. Описание примеров закончено.

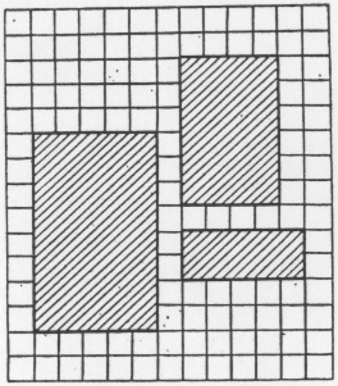


Рис. 1.1. Пример допустимого изображения прямоугольников

Для заданного поля зрения Tv и алфавита сигналов V количество различных локально-конъюнктивных предикатов порядка m не превышает величины $2^{C_n^m |V|^m}$, где $n = |Tv|$, а количество всех возможных предикатов допустимости изображений равно $2^{|V|^n}$. Отсюда непосредственно следует, что с помощью предикатов малого порядка нельзя выразить любой предикат допустимости, так как при малых значениях m справедливо неравенство $C_n^m |V| < |V|^n$. При больших значениях m, в частности при $m = n - 1$ и при $n > |V|$, выполняется неравенство $C_n^m |V|^m > |V|^n$, и возможность или невозможность представления любого предиката как локально-конъюнктивного хотя бы большого порядка не столь очевидна. Определенную ясность в этот вопрос внесли М. Минский и С. Пэйперт [76], доказав, что предикат, определяющий четность или нечетность количества черных клеток изображения, не является локально-конъюнктивным ни на какой структуре. Полученная в результате этого ясность, конечно, недостаточна для решения о пригодности или непригодности локально-конъюнктивных предикатов для распознавания изображений, так как предикат четности не типичен для обработки изображений. Такой контрпример менее убедителен, чем, в частности, положительные примеры, приведенные выше. Здесь требуется более тщательный анализ множеств, которые могут быть выражены локально-конъюнктивным предикатом.

В данном параграфе мы прежде всего сформулируем необходимые условия локальной конъюнктивности предиката.

Будем считать, что предикат $F(Tv)$ изолирует изображение v_0 на структуре $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$, если $F(Tv, v_0) = 0$, и для любого $i = 1, 2, \ldots, m$ существует изображение v_i, такое что $F(Tv, v_i) = 1$, а $v_i(T_i) = v_0(T_i)$.

Теорема 1. *Если для некоторого предиката $F(Tv)$ и структуры $\mathcal{T}$ существует изолированное изображение, то этот предикат не является локально-конъюнктивным на этой структуре.*

Доказательство. Изолированное изображение, существование которого предполагается по условию теоремы, обозначим через v_0. Для него справедливо равенство

$$F(Tv, v_0) = 0. \tag{1.2}$$

Допустим, что теорема неверна и что предикат $F(Tv)$ является локально-конъюнктивным на $\mathcal{T}$. Значит существуют такие локальные предикаты $F(T')$, $T' \in \mathcal{T}$, что

$$F(Tv, v_0) = \bigwedge_{T' \in \mathcal{T}} F(T', v_0(T')). \tag{1.3}$$

Из (1.2) и (1.3) следует существование такого фрагмента $T^* \in \mathcal{T}$, что

$$F(T^*, v_0(T^*)) = 0. \tag{1.4}$$

Из условия, что изображение v_0 изолировано, следует существование изображения v^*, такого, что

$$v^*(T^*) = v_0(T^*), \tag{1.5}$$

$$F(Tv, v^*) = 1. \tag{1.6}$$

Из (1.6) и локальной конъюнктивности $F(Tv)$ следует, что

$$F(T^*, v^*(T^*)) = 1. \tag{1.7}$$

Из (1.7) и (1.5) следует $F(T^*, v_0(T^*)) = 1$, что противоречит (1.4). Следовательно предположение о локальной конъюнктивности предиката $F(Tv)$ неверно. Теорема доказана.

Пример. Как и в рассмотренном ранее примере, полем зрения Tv будем считать множество пар (i, j), $i = 1, 2, \ldots, n$ $j = 1, 2, \ldots, m$, где m и n — достаточно большие целые числа.

Зафиксируем некоторое целое число k и определим для каждого $(i, j) \in Tv$ фрагмент $Tk(i, j)$ поля зрения: $Tk(i, j) = \{(i', j') : (i', j') \in Tv,\ i \leqslant i' \leqslant i + k - 1,\ j \leqslant j' \leqslant j + k - 1\}$. Структуру $\mathcal{T}k$ определим как $\mathcal{T}k = \{Tk(i, j) : (i, j) \in Tv\}$. Иными словами, структура $\mathcal{T}k$ — это множества всех квадратных частей поля зрения со стороной, равной k.

Изображением прямоугольника с координатами i_1, i_2, j_1, j_2 будем называть изображение v_{i_1, i_2, j_1, j_2}, при котором

$$v_{i_1, i_2, j_1, j_2}(i, j) = \begin{cases} 1, \text{ если } j = j_1,\ i_1 \leqslant i \leqslant i_2; \\ 1, \text{ если } j = j_2,\ i_1 \leqslant i \leqslant i_2; \\ 1, \text{ если } i = i_1,\ j_1 \leqslant j \leqslant j_2; \\ 1, \text{ если } i = i_2,\ j_1 \leqslant j \leqslant j_2; \\ 0 \text{ во всех прочих случаях.} \end{cases}$$

Множеством допустимых изображений будем считать множество

$$\{v_{i_1 i_2 j_1 j_2} : (i_1, j_1) \in Tv,\ (i_2, j_2) \in Tv\}.$$

На рис. 1.2 слева представлены некоторые изображения прямоугольников, т. е. допустимые изображения, а справа — недопустимые изображения.

Введенное множество изображений прямоугольников отличается от множества прямоугольников, рассмотренного ранее, хотя это различие и не очень существенно с точки зрения содержания. И тем не менее эти два примера, содержательно достаточно близкие, существенно различаются с точки зрения возможности их представления локально-конъюнктивными предикатами. А именно в рассмотренном ранее примере предикат допустимости изображения был локально-конъюнктивным на структуре $\mathcal{T}k$ при $k = 2$, а в данном примере он не является таковым не только при $k = 2$, но и при других k, бóльших 2. Действительно, если соотношение между размерами поля зрения и значением k таково, что возможно изображение, представленное на рис. 1.3, причем с указанными на нем размерами, то это изображение изолированно. Изолированность этого изображения означает, что для любого квадратного участка поля зрения со стороной, равной k, можно указать изображение прямоугольника, в частности представленным на рисунке штриховыми линиями, которое на данном квадратном участке будет совпадать с изображением непрямоугольника. Иными словами, при введенной структуре поля зрения рассматриваемое множество допустимых изображений не может быть выражено локально-конъюнктивным

2 8—2045

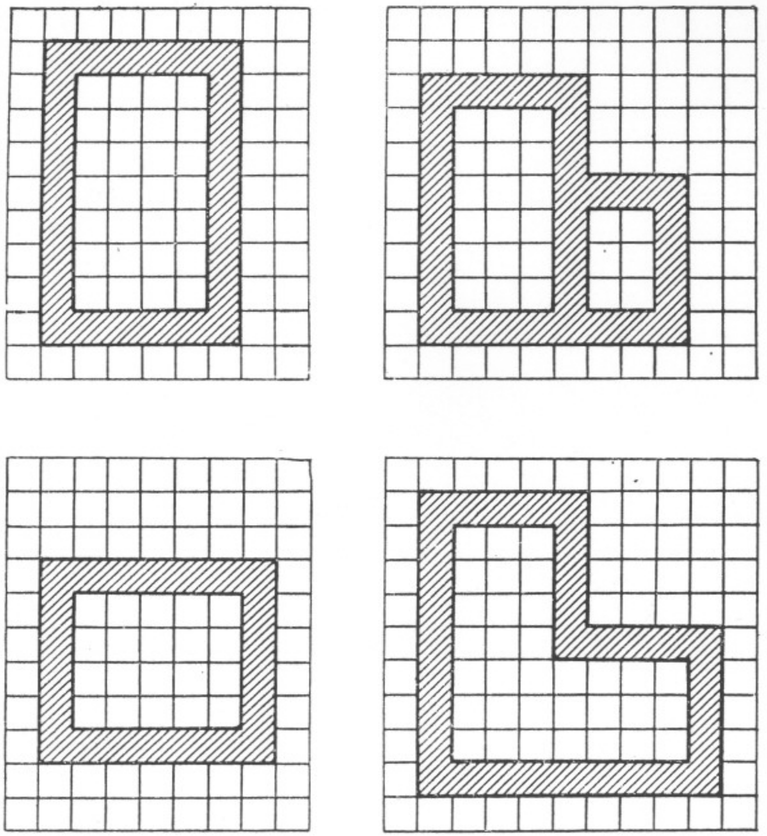


Рис. 1.2. Примеры изображений прямоугольников и непрямоугольников

предикатом, так как существует недопустимое изображение, все фрагменты которого допустимы.

Заканчивая описание примера, отметим, что его рассмотрение выполнено без каких-либо обращений к многомерным геометрическим аналогиям. Хотелось бы надеяться, что при этом обнаружилась простота, которая была бы тщательно скрыта при рассмотрении данного вопроса в терминах многомерных пространств и разделяющих гиперплоскостей. Описание примера закончено.

Доказанное в теореме 1 свойство локально-конъюнктивных предикатов само по себе не является их кардинальным недостатком. Действительно, теорема лишь утверждает, что для любой наперед заданной структуры могут быть подобраны контрпримеры множеств, не выражающихся локально-

конъюнктивно на этой структуре. Однако при решении той или иной конкретной задачи наперед задано именно множество изображений, а не структура, выбор которой полностью находится в руках конструктора. Поэтому можно было бы надеяться, что, задавшись целью реализовать в распознающей системе один-единственный предикат, конструктор сможет подобрать структуру поля зрения так, чтобы предикат выражался локально-конъюнктивно. Оказывается, это не всегда возможно, так как существуют достаточно простые множества, которые не выражаются локально-конъюнктивно ни на какой структуре. Свойства таких множеств определены в следующих двух теоремах.

Теорема 2. *Любой предикат, не являющийся локально-конъюнктивным на структуре* $\mathscr{T}_0 = \{Tv \setminus \{t\} : t \in Tv\}$, *не является локально-конъюнктивным ни на какой другой структуре.*

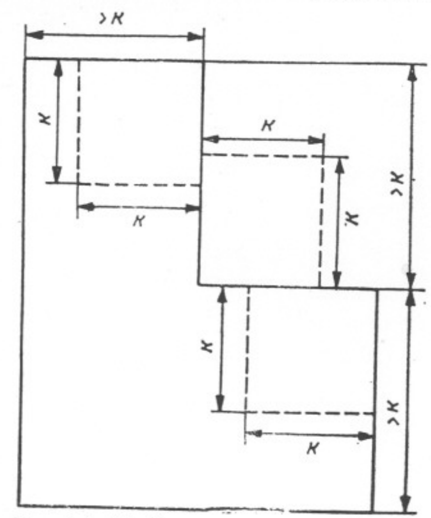


Рис. 1.3. Изображение, изолированное на структуре $\mathscr{T}_k$

Доказательство. Пусть $\mathscr{T}_1$ — любая структура. Для любого фрагмента T_1 из $\mathscr{T}_1$ существует фрагмент T_0 из $\mathscr{T}_0$, такой что $T_1 \subset T_0$. Для любого фрагмента T_0 из $\mathscr{T}_0$ через $\mathscr{T}_1(T_0)$ обозначим множество фрагментов T_1 из $\mathscr{T}_1$, являющихся подмножествами в T_0. Тогда справедливо равенство

$$\mathscr{T}_1 = \bigcup_{T_0 \in \mathscr{T}_0} \mathscr{T}_1(T_0). \tag{1.8}$$

Пусть $F_1(Tv)$ — любой предикат, локально-конъюнктивный на $\mathscr{T}_1$, т. е. существуют такие предикаты $F_1(T_1)$, $T_1 \in \mathscr{T}_1$, что для любого изображения v справедливо равенство

$$F_1(Tv;\ v) = \bigwedge_{T_1 \in \mathscr{T}_1} F_1(T_1,\ v(T_1)). \tag{1.9}$$

Докажем, что предикат $F_1(Tv)$ будет локально-конъюнктивным и на $\mathscr{T}_0$.

Для любого фрагмента T_0 из $\mathscr{T}_0$ определим предикат $F_0(T_0)$, такой, что для любого изображения v справедливо равенство

$$F_0(T_0,\ v(T_0)) = \bigwedge_{T_1 \in \mathscr{T}_1(T_0)} F_1(T_1,\ v(T_1)), \tag{1.10}$$

и предикат $F_0(Tv)$, такой, что для любого изображения справедливо равенство

$$F_0(Tv, v) = \bigwedge_{T_0 \in \mathcal{T}_0} F_0(T_0, v(T_0)). \qquad (1.11)$$

Предикат $F_0(Tv)$ —локально-конъюнктивен вследствие (1.11) и равен $F_1(Tv)$ в результате следующей цепочки равенств:

$$F_0(Tv, v) = \bigwedge_{T_0 \in \mathcal{T}_0} F_0(T_0, v(T_0)) = \bigwedge_{T_0 \in \mathcal{T}_0} \bigwedge_{T_1 \in \mathcal{T}_1(T_0)} F_1(T_1, v(T_1)) =$$

$$= \bigwedge_{T_1 \in \mathcal{T}_1} F_1(T_1, v(T_1)) = F_1(Tv, v).$$

Первое равенство в этой цепочке справедливо в результате (1.11), а второе, третье и последнее — в силу (1.10), (1.8) и (1.9) соответственно. Теорема доказана.

Некоторые изображения назовем сильно изолированными. Будем считать, что предикат $F(Tv)$ сильно изолирует изображение v_0, если $F(Tv, v_0) = 0$, и для любого $t \in Tv$ существует изображение v_t, такое, что $v_0(Tv \setminus \{t\}) = v_t(Tv \,/\, \{t\})$, а $F(Tv, v_t) = 1$.

Теорема 3. *Если для предиката $F(Tv)$ существует по крайней мере одно сильно изолированное изображение, то этот предикат не является локально-конъюнктивным ни на какой структуре.*

Доказательство. Предикат $F(Tv)$, для которого существует хотя бы одно сильно изолированное изображение, вследствие теоремы 1 не является локально-конъюнктивным на структуре $\mathcal{T}_0 = \{Tv \setminus \{t\} : t \in Tv\}$. Следовательно, согласно теореме 2 он не может быть локально-конъюнктивным и на любой другой структуре. Теорема доказана.

Примеры. Из справедливости теоремы 3 становится достаточно очевидной упоминавшаяся ранее теорема М. Минского и С. Пэйперта, что предикат четности не является локально-конъюнктивным ни при какой структуре. Действительно, любое изображение, содержащее нечетное количество черных клеток, сильно изолированно, так как оно недопустимо, а изменение черноты в любой клетке превращает его в допустимое изображение.

Подобным образом можно доказать, что не может быть локально-конъюнктивным предикат, допускающий все изображения, кроме тех, в которых суммарное количество черных клеток равно некоторому фиксированному числу. Не может быть локально-конъюнктивным и любой предикат, допускающий все изображения, кроме некоторого одного. Количество примеров можно было бы продолжить. Описание примеров закончено.

Из доказанных теорем вытекает, что локально-конъюнктивные предикаты обладают ограниченными возможностями.

Неуниверсальность локально-конъюнктивных предикатов, т. е. невозможность с их помощью задать некоторые подмножества, конечно же, является важным их свойством. Вместе с тем это свойство с чисто прагматической точки зрения не следует считать серьезным недостатком, равно как универсальность какого-либо другого средства — серьезным достоинством.

Недостаток локально-конъюнктивных предикатов не в том, что определенные множества не могут быть заданы с их помощью, а в том, что некоторые из этих множеств являются простыми и уместными с точки зрения обработки изображений. Аппарат двумерных грамматик, которому посвящена данная работа, обладая достоинствами локально-коньюнктивных предикатов (простота, наглядность) не имеет указанных недостатков.

Для достижения определенной завершенности анализа локально-конъюнктивных предикатов докажем, что сформулированные выше необходимые условия локальной конъюнктивности предиката являются и достаточными. Для этой цели вводится и исследуется понятие локально-конъюнктивного замыкания предиката.

Предикат $F_1(Tv)$ считается не бо́льшим, чем предикат $F_2(Tv)$, $F_1(Tv) \leqslant F_2(Tv)$, если для любого изображения v, выполняется неравенство $F_1(Tv, v) \leqslant F_2(Tv, v)$.

Если при $F_1(Tv) \leqslant F_2(Tv)$ существует изображение v, такое, что $F_1(Tv, v) < F_2(Tv, v)$, то будем писать $F_1(Tv) < F_2(Tv)$.

Предикат $F^*(Tv)$ называется локально-конъюнктивным замыканием предиката $F(Tv)$ на структуре $\mathcal{T}$, если:

1) $F^*(Tv)$ — локально-конъюнктивный предикат на $\mathcal{T}$;
2) $F^*(Tv) \geqslant F(Tv)$;
3) не существует предикат $F'(Tv)$, локально-конъюнктивный на $\mathcal{T}$, такой, что $F(Tv) \leqslant F'(Tv) < F^*(Tv)$.

Очевидно, что необходимым и достаточным условием локальной конъюнктивности предиката является совпадение его со своим локально-конъюнктивным замыканием.

Для заданного изображения v и фрагмента T' через $\tilde{v}(v, T')$ обозначено множество тех изображений v', для которых $v'(T') = v(T')$.

Теорема 4. *Для любой структуры $\mathcal{T}$ любой предикат $F(Tv)$ имеет единственное локально-конъюнктивное замыка-*

ние $F^*(Tv)$, причем

$$\forall v((F^*(Tv, v)) = \bigwedge_{T' \in \mathcal{T}} \bigvee_{v' \in \tilde{v}(v, T')} F(Tv, v')). \qquad (1.12)$$

Доказательство. Выражение под знаком конъюнкции в (1.12) обозначим через $F(T', v(T'))$. Таким образом,

$$F(T', v(T')) = \bigvee_{v' \in \tilde{v}(v, T')} F(Tv, v'). \qquad (1.13)$$

Из самой записи (1.12) следует, что предикат $F^*(Tv)$ является локально-конъюнктивным.

Докажем, что $F^*(Tv) \geqslant F(Tv)$. Если $F(Tv, v) = 0$ для всех v, то доказываемое неравенство очевидно. Пусть для некоторого v выполняется равенство $F(Tv, v) = 1$. В таком случае согласно (1.13) $F(T', v(T')) = 1$ для всех $T' \in \mathcal{T}$. Следовательно, в силу (1.12) $F^*(Tv, v) = 1$. Неравенство $F^*(Tv) \geqslant F(Tv)$ доказано.

Докажем теперь, что любой локально-конъюнктивный предикат $F^{**}(Tv)$, такой, что $F^{**}(Tv) \geqslant F(Tv)$, будет таким, что $F^{**}(Tv) \geqslant F^*(Tv)$. Допустим, это не так, т. е. существует локально-конъюнктивный предикат $F^{**}(Tv)$, такой, что выполняется неравенство $F^{**}(Tv) \geqslant F(Tv)$, а неравенство $F^{**}(Tv) \geqslant F^*(Tv)$ не выполняется.

Так как предикат $F^{**}(Tv)$ является локально-конъюнктивным на $\mathcal{T}$, существуют такие локальные предикаты $F^{**}(T')$, $T' \in \mathcal{T}$, при которых справедливо равенство

$$\forall v(F^{**}(Tv, v) = \bigwedge_{T' \in \mathcal{T}} F^{**}(T', v(T')). \qquad (1.14)$$

В результате того что $F^{**}(Tv) \geqslant F(Tv)$, справедливо соотношение

$$\forall v(F^{**}(Tv, v) \geqslant F(Tv, v)). \qquad (1.15)$$

Поскольку неравенство $F^{**}(Tv) \geqslant F^*(Tv)$ не выполняется,

$$\exists v(F^{**}(Tv, v) < F^*(Tv, v)). \qquad (1.16)$$

Покажем, что условия (1.12) — (1.16) противоречивы. Изображение, для которого выполняется условие под знаком существования в (1.16), обозначим через v^*. Следовательно, $F^{**}(Tv, v^*) = 0$, а $F^*(Tv, v^*) = 1$. Из первого равенства с учетом (1.14) следует существование $T' \in \mathcal{T}$, такого, что

$$F^{**}(T', v^*(T')) = 0. \qquad (1.17)$$

Из равенства $F^*(Tv, v^*) = 1$ и равенств (1.12) и (1.13) следует, что для этого же фрагмента T' выполняется равенство

$$F(T', v^*(T')) = 1, \qquad (1.18)$$

из которого в силу (1.13) следует существование такого изображения v^{**}, при котором

$$v^{**}(T') = v^*(T'), \qquad (1.19)$$

$$F(Tv, v^{**}) = 1. \qquad (1.20)$$

Из двух последних равенств следует, что для изображения v^{**} справедливо равенство $F^{**}(T', v^{**}(T')) = 0$, из которого с учетом (1.14) следует

$$F^{**}(Tv, v^{**}) = 0. \qquad (1.21)$$

Равенства (1.20) и (1.21) противоречат условию (1.15).

Таким образом, оказалось неверным предположение о существовании локально-конъюнктивного предиката $F^{**}(Tv)$, такого, что выполняется неравенство $F^{**}(Tv) \geqslant F(Tv)$ и не выполняется неравенство $F^{**}(Tv) \geqslant F^*(Tv)$. Теорема доказана.

Теорема 5. *Если для предиката $F(Tv)$ не существует изображения, изолированного на структуре $\mathcal{T}$, то этот предикат локально-конъюнктивный на $\mathcal{T}$.*

Доказательство. Локально-конъюнктивное замыкание предиката $F(Tv)$ на структуре $\mathcal{T}$ согласно теореме 4 выражается формулой

$$\forall v(F^*(Tv, v) = \bigwedge_{T' \in \mathcal{T}} \bigvee_{v' \in \tilde{v}(v, T')} F(Tv, v')). \qquad (1.22)$$

Допустим, что $F(Tv)$ не является локально-конъюнктивным. Это значит, что $F(Tv) \neq F^*(Tv)$, но $F(Tv) \leqslant F^*(Tv)$. Следовательно, $F(Tv) < F^*(Tv)$, и существует изображение v_0, такое что

$$F(Tv, v_0) = 0, \qquad (1.23)$$

$$F^*(Tv, v_0) = 1. \qquad (1.24)$$

Из (1.22) и (1.24) следует, что для любого $T' \in \mathcal{T}$ существует такое изображение $v_{T'}$, что $v_{T'}(T') = v_0(T')$, а $F(Tv, v_{T'}) = 1$. А это значит, что изображение v_0 изолированно на структуре $\mathcal{T}$, а это противоречит условию теоремы.

Следовательно, предположение о том, что предикат $F(Tv)$ не является локально конъюнктивным, неверно. Теорема доказана.

Теорема 6. *Если для предиката $F(Tv)$ не существует сильно изолированного изображения, то предикат $F(Tv)$ локально-конъюнктивный на некоторой структуре.*

Доказательство. Согласно теореме 5 предикат $F(Tv)$ локально-конъюнктивный по крайней мере на структуре $\{Tv \setminus \{t\} : t \in Tv\}$. Теорема доказана.

Пример. Пусть $Tv = \{(i, j) : 1 \leqslant i \leqslant n,\ 1 \leqslant j \leqslant m\}$, где n и m — целые числа, больше двух. Изображением будем считать функцию $Tv \to \{0, 1\}$. Две клетки поля зрения (i_1, j_1) и (i_2, j_2) будем считать соседними, если $|i_1 - i_2| + |j_1 - j_2| = 1$. Изображение v будем считать допустимым, если для любых двух клеток $t_0 \in Tv$ и $t_k \in Tv$, таких, что $v(t_0) = 1$ и $v(t_k) = 1$, существует последовательность $t_0, t_1, \ldots, t_k$ клеток, таких, что для всех $s = 1, 2, \ldots, k$ справедливо равенство $v(t_s) = 1$, а клетки t_s, t_{s-1} — соседние. Приведенное определение допустимого изображения есть определение дискретного аналога связности.

Почему-то распространилось мнение, что понятие связности в приведенной формулировке не является локально-конъюнктивным. Зачастую это представление связывается с известной монографией М. Минского и С. Пэйперта [76], хотя в ней и не содержится явного ответа на этот вопрос. В этом смысле интересен тот факт, что предикат связности все-таки локально-конъюнктивный, так как в противном случае существовало бы несвязное изображение, которое изменением черноты в любой клетке превращалось бы в связное, а такого изображения не существует. Описание примера закончено.

Интересные примеры, иллюстрирующие понятие локальной конъюнктивности, представляют собой так называемые невозможные изображения, наиболее известные по рисункам голландского графика М. Эшера. Фактически любое невозможное изображение входит в локально-конъюнктивное замыкание того или иного множества возможных изображений. Наиболее впечатляющие (и наиболее известные) примеры невозможных изображений построены так, что их невозможность не обнаруживается даже на очень больших частях изображения, а обнаруживается только при наблюдении всего изображения целиком. Именно это иллюстрирует тот факт, что множество изображений, воспринимаемых человеком, как возможные, не выражается локально-конъюнктивно даже на очень сильных структурах. Однако человек все-таки обнаруживает несуразность невозможных изображений, и это может служить косвенным доводом в пользу того, сколь далеко за пределами локальной конъюнктивности находятся действительные механизмы восприятия и понимания изображений.

1.2. ДВУМЕРНЫЕ ГРАММАТИКИ ВТОРОГО ПОРЯДКА

По-прежнему через Tv будем обозначать конечное множество, называемое полем зрения, а через V — конечный алфавит сигналов. Изображение — это функция $Tv \to V$, имеющая областью определения поле зрения, а областью значений — алфавит сигналов. Из элементов алфавита сигналов непосредственно составляется изображение, и в этом смысле он основной. В какой-то мере алфавит сигналов аналогичен алфавиту терминальных символов в обычных формальных грамматиках.

Кроме алфавита сигналов, будем считать заданным алфавит K, называемый алфавитом структурных элементов. Структурный алфавит является вспомогательным и служит для того, чтобы с его помощью определять множества изображений, т. е. строить допустимые изображения. Структурные элементы не входят в изображение, в этом смысле они аналогичны нетерминальным символам в обычных формальных грамматиках.

Из структурных элементов составляются так называемое описание изображения, которое определяется как функция $Tk \to K$, имеющая областью определения конечное множество Tk, называемое полем описания, и областью значений алфавит K. В простейших случаях поле описания изоморфно полю зрения, однако так бывает не всегда.

Объединение поля зрения и поля описания называется просто полем и обозначено T. Элементы множества T называются клетками. Через S обозначено объединение $V \cup K$ алфавитов сигналов и структурных элементов. Вариантом называется функция $T \to S$, сужение которой на поле зрения есть изображение, а сужение на поле описания — описание. Иными словами, вариант есть пара «изображение-описание».

Пусть $\mathcal{T}$ — структура второго порядка на поле T, т. е. некоторое подмножество пар клеток из T. Пара клеток, входящих в T, будет называться парой соседних клеток. Пусть для каждой пары $T' \in \mathcal{T}$ соседних клеток задан предикат $F(T') : S^{T'} \to \{0, 1\}$, называемый локальным предикатом допустимости фрагментов вариантов на паре клеток T'. Совокупность этих предикатов обозначим через Ω; $\Omega = \{F(T') : T' \in \mathcal{T}\}$.

Двумерная грамматика есть совокупность перечисленных средств, т. е. шестерка $\langle V, K, Tv, Tk, \mathcal{T}, \Omega \rangle$, где $\Omega = \{F(T') : T' \in \mathcal{T}\}$.

Для любого варианта $s : T \to S$ и любого подмножества $T' \subset T$ через $s(T')$ обозначим сужение функции s на T', а для любой клетки $t \in T$ через $s(t)$ — значение функции для клетки t.

Вариант $s : T \to S$ называется допустимым вариантом в грамматике $G = \langle V, K, Tv, Tk, \mathcal{T}, \{F(T') : T' \in \mathcal{T}\} \rangle$, если для любого $T' \in \mathcal{T}$ справедливо равенство $F(T', s(T')) = 1$.

Изображение $v : Tv \to V$ называется допустимым изображением в грамматике G, если существует вариант s, допустимый в этой грамматике, такой, что $v = s\,(Tv)$.

Множество изображений, допустимых в грамматике G, обозначим через $\mathscr{L}\,(G)$, а множество допустимых вариантов — через $S\,(G)$.

Непосредственно из определений двумерной грамматики и локально-конъюнктивных предикатов следует, что при заданных поле зрения и алфавите сигналов для любого локально-конъюнктивного предиката F второго порядка существует такая двумерная грамматика G, что $(F\,(v) = 1) \Leftrightarrow (v \in \mathscr{L}(G))$, причем в этой двумерной грамматике $Tk = \varnothing$ и $k = \varnothing$. Однако справедлива более сильная теорема.

Теорема 7. *Для любого поля зрения Tv, структуры $\mathscr{T}$ поля зрения и алфавита V сигналов и любого предиката $F(Tv)$, локально-конъюнктивного на $\mathscr{T}$, существует двумерная грамматика G, такая, что* $\mathscr{L}\,(G) = \{v : v \in V^T,\ F\,(Tv,\ v) = 1\}$.

Доказательство. Построим грамматику, существование которой требуется доказать. Поле описания Tk в этой грамматике определим как множество клеток, изоморфное множеству $\mathscr{T}$ — структуре предиката $F\,(Tv)$. Для любого $T' \in \mathscr{T}$ через $t\,(T')$ обозначим клетку $t \in Tk$, соответствующую фрагменту T' в принятом изоморфизме.

Структуру $\mathscr{T}_{\text{г}}$ искомой грамматики определим как множество $\{\{t',\ t\,(T')\} : t' \in T',\ T' \in \mathscr{T}\}$. Это значит, что каждый фрагмент поля $Tv \cup Tk$ в структуре грамматики состоит из одной клетки t_1 поля зрения и одной клетки t_2 поля описания, причем клетка t_1 входит в фрагмент поля зрения, которому соответствует клетка t_2.

Предикат $F\,(Tv)$ является локально-конъюнктивным на $\mathscr{T}$, и поэтому существуют предикаты $F\,(T')$, $T' \in \mathscr{T}$, такие, что

$$\forall v\,(F\,(Tv,\ v) = \bigwedge_{T' \in \mathscr{T}} F\,(T',\ v\,(T'))).$$

В качестве алфавита K структурных элементов выберем множество, количество элементов в котором равно

$$\max_{T' \in T} |\{v\,(T') : F\,(T',\ v\,(T')) = 1\}|.$$

Для каждого $T' \in \mathscr{T}$ выберем любую функцию $k\,(T') : \{v\,(T') : F\,(T',\ v\,(T')) = 1\} \to K$, такую, что при $v_1\,(T') \neq v_2\,(T')$ выполняется условие $k\,(T',\ v_1\,(T')) \neq k\,(T',\ v_2\,(T'))$.

Для каждого фрагмента $T' \in \mathscr{T}$, состоящего из клетки $t_1 \in T' \subset Tv$ и клетки $t_2 = t\,(T')$, выберем предикат $F_{\text{г}}\,(T'')$ так, чтобы он принимал значение, равное 1, для тех и только тех пар $v^* = s\,(t_1)$ и $k^* = s\,(t_2)$, для которых существует такой фрагмент $v\,(T')$, что $v\,(T',\ t_1) = v^*$, а $k\,(T',\ v\,(T')) = k^*$.

Выбор предикатов $F_{\text{г}}\,(T'')$, $T'' \in \mathscr{T}_{\text{г}}$ указанным способом приводит к выполнению следующих двух свойств.

Свойство 1. Для любого фрагмента $T' \in \mathscr{T}$ любая функция $s\,(T')$, такая, что $F\,(T',\ s\,(T')) = 1$, имеет единственное расширение $s\,(T' \cup \{t\,(T')\})$, такое, что для любого $T'' = \{t_1,\ t\,(T')\}$, $t_1 \in T'$, будет справедливо равенство $F_{\text{г}}\,(T'',\ s\,(T'')) = 1$. При этом $s\,(t\,(T')) = k\,(T',\ s\,(T'))$.

Для любого $T' \in \mathscr{T}$ через $\mathscr{T}_{\text{г}}\,(T')$ обозначим подмножество фрагментов T'' из $\mathscr{T}_{\text{г}}$, таких что $T'' \subset T' \cup \{t\,(T')\}$.

Свойство 2. Для любого фрагмента $T' \in \mathscr{T}$ и любой функции $s\,(T' \cup \{t\,(T')\})$, такой, что $\bigwedge_{T'' \in \mathscr{T}_{\text{г}}(T')} F_{\text{г}}\,(T'',\ s\,(T'')) = 1$ справедливо равенство $F\,(T',\ s\,(T')) = 1$.

Докажем, что грамматика $G = \langle Tv,\ Tk,\ V,\ K,\ \mathscr{T}_{\text{г}},\ \{F_{\text{г}}\,(T''),\ T'' \in \mathscr{T}_{\text{г}}\}\rangle$, где множества K, Tk, $\mathscr{T}_{\text{г}}$ и функции $F_{\text{г}}\,(T'')$, $T'' \in \mathscr{T}_{\text{г}}$ определены, как указано выше, является искомой.

Докажем вначале, что $\mathscr{L}\,(G) \subset \{v : F\,(Tv,\ v) = 1\}$. Пусть $v\,(Tv) \in \mathscr{L}\,(G)$. Это значит, что существует допустимый вариант $s : T \to S$, такой, что $s\,(Tv) = v\,(Tv)$. Для некоторого фрагмента $T' \in \mathscr{T}$ рассмотрим фрагмент $s\,(T' \cup \{t\,(T')\})$. Так как вариант s допустим, для него, а следовательно, и для фрагмента $s\,(T' \cup \{t\,(T')\})$, справедливо равенство $\bigwedge_{\substack{T'' \in \mathscr{T}_{\text{г}} \\ T'' \subset (T' \cup \{t(T')\})}} F_{\text{г}}\,(T'', s\,(T'')) = 1$. Поэтому из свойства 2 вытекает равенство $F\,(T',\ s\,(T')) = 1$.

Эти рассуждения справедливы для любого $T' \in \mathscr{T}$, и поэтому справедливо равенство $\bigwedge_{T' \in \mathscr{T}} F\,(T',\ s\,(T')) = 1$. Поскольку $s\,(Tv) = v\,(Tv)$, а $T' \subset Tv$, последнее равенство может быть записано в виде $\bigwedge_{T' \in \mathscr{T}} F(T',\ v\,(T')) = 1$, а это значит, что изображение $v\,(Tv)$ допускается заданным локально-конъюнктивным предикатом $F\,(Tv)$.

Докажем теперь, что $\{v : F\,(Tv,\ v) = 1\} \subset \mathscr{L}\,(G)$. Пусть v — функция $Tv \to V$, такая, что $F\,(Tv,\ v) = 1$. Определим расширение этой функции на $Tv \cup Tk$, так, что $s\,(t\,(T')) = k\,(T',\ v\,(T'))$ для любого $t \in Tk$, $t = t\,(T')$. Определенная таким образом функция $s : T \to S$ есть допустимый в G вариант. Действительно, из равенства $F\,(Tv,\ v) = 1$ следует, что для любого $T' \in \mathscr{T}$ выполняется равенство $F\,(T',\ v\,(T')) = 1$. Поэтому в силу свойства 1 для любого $T' \in \mathscr{T}$ и любого $T'' \in \mathscr{T}_{\text{г}}\,(T')$ выполняется равенство $F_{\text{г}}\,(T'',\ s\,(T'')) = 1$. Это значит, что построенный вариант s является допустимым в G, а его сужение $s(Tv) = v$ является изображением, допустимым в G. Теорема доказана.

Доказанную теорему можно пояснить на примере.

Пример. Пусть поле зрения Tv — множество, состоящее из четырех клеток: t_1, t_2, t_3, t_4; структура $\mathcal{T}$ содержит всего два подмножества: $T_1 = \{t_1, t_2, t_3\}$ и $T_2 = \{t_1, t_2, t_4\}$; алфавит сигналов $V = \{0,1\}$. Пусть локальный предикат допустимости $F(T_1)$ принимает значение, равное 1, на двух фрагментах изображения:

$(v_1(t_1) = 1,\ v_1(t_2) = 0,\ v_1(t_3) = 1)$ — первый фрагмент на T_1;

$(v_2(t_1) = 0,\ v_2(t_2) = 1,\ v_2(t_3) = 0)$ — второй фрагмент на T_1.

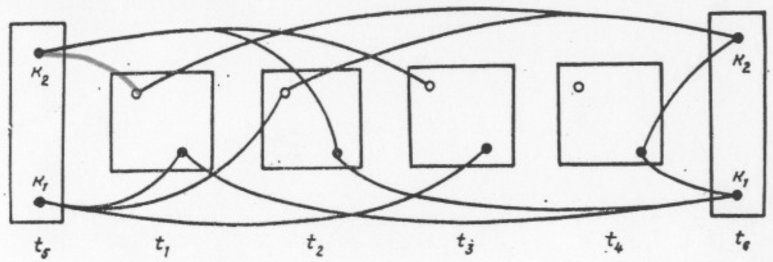


Рис. 1.4. Задание локально-конъюнктивного предиката третьего порядка двумерной грамматикой

Локальный предикат $F(T_2)$ допускает также только два фрагмента:

$(v_3(t_1) = 1,\ v_3(t_2) = 1,\ v_3(t_4) = 1)$ — первый фрагмент на T_2;

$(v_4(t_1) = 0,\ v_4(t_2) = 0,\ v_4(t_4) = 1)$ — второй фрагмент на T_2.

Допустимость изображения v определяем как $F(T_1, v(T_1)) \wedge \wedge F(T_2, v(T_2))$.

Определим двумерную грамматику G, порождающую указанные допустимые изображения и только их. Поскольку структура $\mathcal{T}$ состоит из двух фрагментов, поле описания Tk будет состоять из двух клеток: t_5 и t_6, причем клетка t_5 соответствует фрагменту $\{t_1, t_2, t_3\}$, а клетка t_6 — фрагменту $\{t_1, t_2, t_4\}$. Структура $\mathcal{T}_\text{г}$ грамматики G будет, таким образом, состоять из следующих пар соседних клеток: $\{(t_1, t_5), (t_2, t_5), (t_3, t_5), (t_1, t_6), (t_2, t_6), (t_4, t_6)\}$.

Каждый из локальных предикатов $F(T_1)$ и $F(T_2)$ допускает только два фрагмента, и поэтому структурный алфавит состоит из двух элементов: k_1 и k_2.

Дальнейшее построение грамматики G иллюстрируется рис. 1.4, на котором квадратами представлены клетки t_1, t_2, t_3, t_4 поля зрения и клетки t_5 и t_6 поля описания. Внутри каждой клетки нарисованы два кружочка. Белый кружок в клетке поля зрения соответствует сигналу 0, а черный — 1. Два кружочка в каждой из клеток поля описания соответствуют двум структурным элементам. Для каждой клетки поля зрения, или, что то же, для каждого фрагмента поля зрения, каждый из двух структурных элементов соответствует некоторому допустимому фрагменту изображения. Так, скажем, для клетки t_5 (или, что то же, для фрагмента $\{t_1, t_2, t_3\}$) структурный элемент k_1 соответствует фрагменту $v(t_1) = 1$, $v(t_2) = 0$, $v(t_3) = 1$, а структурный элемент k_2 — фрагменту $v(t_1) = v(t_2) = 1$, $v(t_3) = 0$. Совокупность локальных предикатов $F_\text{г}(T'')$ для грамматики G представлена на рисунке линиями, которые соединяют кружочки, представляющие собой сигналы и структурные элементы. А именно: если какая-то пара, например k_1 в клетке t_6 и 1 в клетке t_1, соединена дужкой, это означает, что $F_\text{г}(\{t_1, t_6\}, (1, k_1)) = 1$.

Таким образом, рисунок представляет двумерную грамматику в следующем смысле. Вариант есть подмножество выбранных кружочков, такое, что в каждой клетке выбран один и только один кружок. Вариант является допустимым тогда и только тогда, когда для любой пары соседних клеток выбранные в этих клетках кружочки соединены линией.

Очевидно, что представленная на рисунке грамматика не содержит ни одного допустимого варианта, а следовательно, ни одно изображение в этой грамматике не является допустимым, что согласуется с введенным локально-конъюнктивным предикатом, который тождественно равен 0 для любого изображения. Пример закончен.

Мы доказали, что двумерные грамматики обладают возможностями по крайней мере не меньшими, чем локально-конъюнктивные предикаты. Это показывает, сколь существенное значение имеет введение структурных элементов как вспомогательных переменных. Выше отмечалось, что локально-конъюнктивные предикаты есть попытка выразить ограничения на совокупность из n признаков, непосредственно описывающих объект, с помощью ограничений на некоторые пары, тройки, четверки и т. д. признаков. Естественно, что с помощью ограничений, накладываемых на пары признаков, можно выразить не все то, что можно выразить с помощью ограничений на тройки, и, вообще, переход от рассмотрения групп из l признаков к группам из $l+1$ признаков есть расширение выразительных возможностей. Как следует из теоремы, такого же расширения возможностей можно добиться введением вспомогательных переменных, а именно структурных элементов, что позволяет оставаться на уровне рассмотрения только пар переменных. Сейчас мы поставим вопрос о том, позволяет ли введение структурных элементов выражать такие ограничения на совокупность из n признаков, которые не сводятся к заданию ограничений на совокупность из l признаков ни при каком $l < n$. Иными словами, мы ставим вопрос о существовании множества изображений, которое порождается некоторой двумерной грамматикой, но не выражается никаким локально-конъюнктивным предикатом, и отвечаем на него утвердительно. Для обоснования такого ответа достаточно рассмотреть любой предикат, не являющийся локально-конъюнктивным, и построить для него двумерную грамматику. Наиболее иллюстративный в этом отношении предикат четности.

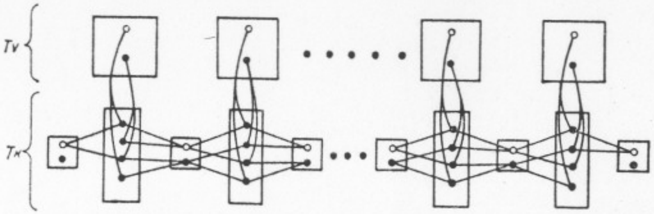


Рис. 1.5. Представление предиката «четность» двумерной грамматикой.

Предикат четности задает множество изображений $v: Tv \to \{0, 1\}$, таких, что количество клеток $t \in Tv$, для которых $v(t) = 1$, является четным. На рис. 1.5 представлена грамматика, порождающая это множество изображений. Нижний ряд квадратов представляет поле зрения; верхний ряд квадратов и прямоугольников — поле описания. Черный кружок в клетке поля зрения соответствует единице, белый — нулю. Черный (или белый) кружок в клетке поля описания обозначенной квадратом, означает, что последовательность сигналов в клетках поля зрения слева от этого квадрата содержит нечетное (соответственно четное) количество единиц. Любая вершина в клетке поля описания, обозначенной прямоугольником, соединена дужкой только с одной вершиной в каждой из соседних клеток. Эти связи выражают правило чередования свойства четность — нечетность при дописывании к части последовательности очередного символа. А именно: если очередной символ равен единице, то свойство четность — нечетность меняется, в противном случае — сохраняется.

1.3. ДВУМЕРНЫЕ ГРАММАТИКИ БОЛЕЕ ЧЕМ ВТОРОГО ПОРЯДКА

Мы видим, что введение поля описания и алфавита структурных элементов существенно увеличивает порождающие способности по сравнению с локально-конъюнктивными предикатами. Сейчас нас интересует вопрос о том, могут ли быть дополнительно увеличены порождающие способности двумерной грамматики, если снять ограничение, заключающееся в том, что структура грамматики должна быть непременно второго порядка, и ответим на него отрицательно. Сформулируем этот результат более определенно.

Двумерной грамматикой, не обязательно второго порядка, называется шестерка $\langle Tv,\ Tk,\ V,\ K,\ \mathcal{T},\ \{F(T'),\ T' \in \mathcal{T}\}\rangle$, где Tv, Tk, V, K — конечные множества, $\mathcal{T}$ — множество подмножеств в $Tv \cup Tk$, а $F(T')$, $T' \in \mathcal{T}$, — функция $(V \cup K)^{T'} \to \{0, 1\}$. Функция $s: (Tv \cup Tk) \to (V \cup K)$ называется допустимым в G вариантом, если для любого $T' \in \mathcal{T}$ справедливо равенство $F(T', s(T')) = 1$. Функция $v: Tv \to V$ называется допустимым в G изображением, если существует вариант s, допустимый в G, такой, что v есть сужение s на Tv. Вопрос заключается в том, существует ли такое множество изображений, которое порождается некоторой двумерной грамматикой и не порождается никакой двумерной грамматикой второго порядка. Ответ на него дает следующая теорема.

Теорема 8. *Для любой грамматики G существует грамматика G^* второго порядка, такая, что $\mathcal{L}(G) = \mathcal{L}(G^*)$.*

Доказательство. Пусть $G = \langle Tv,\ Tk,\ V,\ K,\ \mathcal{T},\ \{F(T'),\ T' \in \mathcal{L}\}\rangle$. Допустимость варианта $(Tv \cup Tk) \to (V \cup K)$ в этой грамматике является по определению локально-конъюнктивным на $\mathcal{T}$ предикатом. По теореме 7 существует двумерная грамматика второго порядка G_0, такая, что $v(G_0) = s(G)$. Эта грамматика имеет вид

$$G_0 = \langle Tv \cup Tk,\ Tk_0,\ V \cup K,\ K_0,\ \mathcal{T}_0,\ \ \{F_0(T'),\ T' \in \mathcal{T}_0\}\rangle.$$

В грамматике G_0 поле описания Tk_0, структурный алфавит K_0, структура $\mathcal{T}_0$ и предикаты $F_0(T')$ должны быть такими, как указано при доказательстве теоремы 7.

Докажем, что искомая грамматика имеет вид

$$G^* = \langle Tv,\ Tk \cup Tk_0,\ V,\ K \cup K_0,\ \mathcal{T}_0,\ \ \{F_0(T'),\ T' \in \mathcal{T}_0\}\rangle.$$

Грамматики G_0 и G^* отличаются лишь разбиением поля на поле зрения и поле описания и разбиением алфавита на сигнальный и структурный алфавиты. Структура $\mathcal{T}_0$ этих грамматик и локальные предикаты допустимости $F_0(T')$, $T' \in \mathcal{T}_0$ у них совпадают, т. е. совпадающим является то, что определяет множество допустимых вариантов. Поэтому $S(G_0) = S(G^*)$.

Докажем вначале, что $\mathcal{L}(G) \subset \mathcal{L}(G^*)$. Пусть изображение $v \in \mathcal{L}(G)$. Значит, существует вариант $s \in S(G)$, такой, что $s(Tv) = v$. По построению грамматики G_0 справедливо включение $s \in \mathcal{L}(G_0)$. Значит существует вариант $s^* \in S(G_0)$, такой, что $s = s^*(Tv \cup Tk)$, а $v = s^*(Tv)$. Поскольку $S(G_0) =$

$= S(G^*)$, то s^* есть допустимый в G^* вариант, сужение которого на Tv есть изображение v. Следовательно, $v \in \mathcal{L}(G^*)$.

Докажем теперь, что $L(G^*) \subset \mathcal{L}(G)$.

Пусть $v^* \in \mathcal{L}(G^*)$. Значит существует вариант $s^* \in S(G^*)$, такой, что $v^* = s^*(Tv)$. Вариант s^* является допустимым и в грамматике G_0, так как $S(G^*) = S(G_0)$. Сужение $s^*(Tv \cup Tk)$ этого варианта на $Tv \cup Tk$ есть изображение в грамматике G_0, а следовательно, оно является допустимым вариантом в G, так как $v(G_0) = S(G)$. Поскольку $s^*(Tv \cup Tk)$ есть допустимый вариант в G, то $s^*(Tv)$ есть допустимое изображение в G. Однако $s^*(Tv) = v^*$. Следовательно, $v^* \in \mathcal{L}(G)$. Теорема доказана.

Справедливость доказанной теоремы позволяет использовать, где это удобно, грамматики более высоких порядков, а при доказательстве отдельных теорем ограничиваться без потери общности рассуждений лишь рассмотрением грамматик второго порядка.

1.4. ПРЕДСТАВЛЕНИЕ АВТОМАТНЫХ И КОНТЕКСТНО-СВОБОДНЫХ ЯЗЫКОВ ДВУМЕРНЫМИ ГРАММАТИКАМИ

В данном параграфе выясним соотношение между двумерными грамматиками и автоматными и контекстно-свободными языками.

Пусть G^* — автоматная или контекстно-свободная грамматика, а $\mathcal{L}(G^*)$ — язык, порождаемый этой грамматикой.

Для заданного целого числа n через $\mathcal{L}(G^*, n)$ обозначим подмножество предложений в $\mathcal{L}(G^*)$, имеющих длину n. Будем считать, что подмножество $\mathcal{L}(G^*, n)$ равно множеству $\mathcal{L}$ изображений, допустимых в грамматике G, если выполняются следующие три условия:

1. Поле зрения Tv грамматики G есть упорядоченное множество, состоящее из n клеток $t_1, t_2, \ldots, t_n$.
2. Для любого предложения $(v_1, v_2, \ldots, v_n) \in \mathcal{L}(G^*, n)$ существует такое допустимое в G изображение $v: Tv \to V$, что $v(t_i) = v_i$ для всех $i = 1, 2, \ldots, n$.
3. Для любого допустимого в G изображения $v: Tv \to V$ справедливо, что $(v(t_1), v(t_2), \ldots, v(t_n)) \in \mathcal{L}(G^*, n)$.

Укажем единый конструктивный прием, с помощью которого для любой автоматной грамматики G_A и любого целого числа n можно построить двумерную грамматику G, такую, что $\mathcal{L}(G) = \mathcal{L}(G_A, n)$. Другой, но тоже единый по всем грамматикам и целым n, прием мы укажем для контекстно-свободных грамматик.

Теорема 9. *Для любой автоматной грамматики G_A и целого числа n существует двумерная грамматика G, такая, что $\mathcal{L}(G_A, n) = \mathcal{L}(G)$.*

Д о к а з а т е л ь с т в о. Автоматная грамматика есть четверка $\langle V, K, k^*, P\rangle$, где V и K — конечные множества, называемые терминальным и нетерминальным алфавитом; $k^* \in K$ — нетерминальный символ, называемый аксиомой; $P = P_3 \cup P_2$, где $P_3 \subset K \times V \times K$, а $P_2 \subset K \times V$. Элемент $p \in P$ называется подстановкой. Считается, что предложение $v_1, v_2, \ldots, v_n$ длины n порождается грамматикой G_A, если существует такая последовательность $k_1, k_2, \ldots, k_n$, что $k_1 = k^*$, $(k_n, v_n) \in P_2$, а для любого $i = 1, 2, \ldots, n-1$ справедливо условие $(k_i, v_i, k_{i+1}) \in P_3$. Из приведенного определения непосредственно следуют указания о построении искомой двумерной грамматики следующим образом.

Алфавит сигналов совпадает с терминальным алфавитом V. Алфавит структурных элементов совпадает с нетерминальным алфавитом K. Поле зрения Tv представляет собой множество из n клеток $t_1, t_2, \ldots, t_n$, поле описания Tk — множество из клеток $\tau_1, \tau_2, \ldots, \tau_n$. Структура $\mathcal{T}$ искомой грамматики есть структура третьего порядка, которая состоит из множества $\{\tau_1\}$, множества $\{\tau_n, t_n\}$ и множеств $\{\tau_i, t_i, \tau_{i+1}\}$, $i = 1, 2, \ldots, n-1$.

Совокупность локальных предикатов допустимости для фрагментов поля $Tv \cup Tk$, входящих в $\mathcal{T}$, должна быть выбрана следующим образом.

Для $T' = \{\tau_1\}$ предикат $F(T')$ должен принимать значение 1 только для $s(\tau_1) = k^*$.

Для $T' = \{\tau_n, t_n\}$ предикат $F(T', s(T'))$ принимает значение 1 только при $s(T') = (s(\tau_n), s(t_n)) \in P_2$.

Для $T' = \{\tau_i, t_i, \tau_{i+1}\}$, $i = 1, 2, \ldots, n$, предикат $F(T', s(T'))$ принимает значение 1 только при $s(T') = (s(\tau_i), s(t_i), s(\tau_{i+1})) \in P_3$. Теорема доказана.

Теорема 10. *Для любой контекстно-свободной грамматики $G_{кс}$ и целого числа n существует двумерная грамматика G, такая, что $\mathcal{L}(G_{кс}, n) = \mathcal{L}(G)$.*

Д о к а з а т е л ь с т в о. Грамматика $G_{кс}$ есть четверка $\langle V, K, k^*, P\rangle$, где V и K — терминальный и нетерминальный алфавиты, $k^* \in K$ — символ, называемый аксиомой. В силу известной теоремы [10], любая КС-грамматика может быть представлена в эквивалентной форме, для которой $P = P_2 \cup P_3$, где $P_3 \subset K \times K \times K$, а $P_2 \subset K \times V$. Через s^* обозначим множество предложений вида $s_1, s_2, \ldots, s_m$, конечной длины, $m \geqslant 0$, где $s_i \in V \cup K$ для любого i. Считается, что предложение $\bar{s}_2 \in s^*$ непосредственно выводится в грамматике $G_{кс}$ из предложения $\bar{s}_1 \in s^*$, $\bar{s}_1 \to \bar{s}_2$, если $\bar{s}_1$

представлено в виде $\bar{s}_1 = As_1B$, где $A \in s^*$ и $B \in s^*$, а $\bar{s}_2$ представимо в одном из следующих двух видов:

$$\bar{s}_2 = A\, s_2 s_3 B, \text{ где } (s_1, s_2, s_3) \in P_3; \tag{1.25}$$

$$\bar{s}_2 = As_2B, \text{ где } (s_1, s_2) \in P_2. \tag{1.26}$$

В случае (1.25) запишем $\bar{s}_1 \xrightarrow{P_3} \bar{s}_2$, а в случае (1.26) — $\bar{s}_1 \xrightarrow{P_2} \bar{s}_2$. Ясно, что в случае (1.25) длина предложения $\bar{s}_2$ ровно на единицу больше длины предложения $\bar{s}_1$, а количество терминальных символов в $\bar{s}_2$ равно количеству терминальных символов в $\bar{s}_1$. В случае (1.26) длина предложения $\bar{s}_1$ равна длине предложения $\bar{s}_2$, а количество терминальных символов в $\bar{s}_2$ ровно на единицу больше количества терминальных символов в $\bar{s}_1$.

Транзитивное замыкание отношения $\bar{s}_1 \xrightarrow{P_2} \bar{s}_2$ обозначим как $\bar{s}_1 \overset{P_2}{\Rightarrow} \bar{s}_2$ транзитивное замыкание $\bar{s}_1 \xrightarrow{P_3} \bar{s}_2$ — как $\bar{s}_1 \overset{P_3}{\Rightarrow} \bar{s}_2$ и, наконец, транзитивное замыкание $\bar{s}_1 \to \bar{s}_2$ — как $\bar{s}_1 \Rightarrow \bar{s}_2$. Отношение $\bar{s}_1 \Rightarrow \bar{s}_2$ будем называть выводимостью в $G_{\text{кс}}$. Множество $\mathscr{L}(G_{\text{кс}}, n)$ есть множество предложений вида $s_1, s_2, \ldots, s_n$, выводимых из предложения, состоящего из единственного символа k^*, и таких, что $s_i \in V$ для любого $i = 1, 2, \ldots, n$.

Для доказательства теоремы докажем вначале вспомогательное утверждение, что для любого предложения $\bar{s} \in \mathscr{L}(G_{\text{кс}}, n)$ существует такое предложение s, для которого справедлива пара отношений $k^* \overset{P_3}{\Rightarrow} s \overset{P_2}{\Rightarrow} \bar{s}$.

Пусть некоторое предложение $\bar{s}$ длины n порождается грамматикой $G_{\text{кс}}$. Это значит, что существует такая последовательность предложений $\bar{s}_2, \bar{s}_3, \ldots, \bar{s}_{2n-1}$, для которой справедлива цепочка отношений $B = k^* \to \bar{s}_2 \to \bar{s}_3 \to \ldots \to \bar{s}_{2n-1} \to \bar{s}$. В силу указанных выше свойств отношений $\xrightarrow{P_3}$ и $\xrightarrow{P_2}$ и на основе того, что предложение $\bar{s}$ длины n содержит n терминальных символов, а предложение $\bar{k}^*$, естественно, не содержит терминальных символов, указанная цепочка отношений содержит ровно $n-1$ отношений $\xrightarrow{P_3}$ и ровно n отношений $\xrightarrow{P_2}$. Рассмотрим в цепочке B тройку предложений $\bar{s}_{i-1}, \bar{s}_i, \bar{s}_{i+1}$, таких, что $\bar{k}_{i-1} \xrightarrow{P_2} \bar{s}_i \xrightarrow{P_3} \bar{s}_{i+1}$. Если такой тройки нет, то это значит, что $\bar{k}^* \overset{P_3}{\Rightarrow} s_n \overset{P_2}{\Rightarrow} \bar{s}$, и вспомогательное утверждение является верным. Если же такая тройка существует, то предложение $\bar{s}_i$ в цепочке B можно заменить на предложение $\bar{s}'_i$ так, что $\bar{s}_{i-1} \xrightarrow{P} \bar{s}'_i \xrightarrow{P_2} \bar{s}_{i+1}$. Эта замена выполняется по следующим правилам. Поскольку $\bar{s}_{i-1} \xrightarrow{P_2} \bar{s}_i$, то $\bar{s}_{i-1}$ представимо в виде Uk_1V, где $U \in S^*$, $V \in S^*$, $k_1 \in K$, а $\bar{s}_i$ — в виде Uv_1V, где $(k, v_1) \in P_2$. Поскольку $\bar{s}_i \xrightarrow{P_3} \bar{s}_{i+1}$, а значит, $Uv_1V \xrightarrow{P_3} \bar{s}_{i+1}$, то имеет место одна из двух ситуаций:

1. $U = Wk_2T, \quad W \in s^*, \quad T \in s^*, \quad k_2 \in K;$

   $\bar{s}_{i+1} = Wk_3k_4Tv_1V, \quad (k_2, k_3, k_4) \in P_3.$
2. $V = Wk_2T, \quad W \in s^*, \quad T \in s^*, \quad k_2 \in K;$

   $\bar{s}_{i+1} = Uv_1Wk_3k_4T, \quad (k_2, k_3, k_4) \in P_3.$

В первой из указанных ситуаций выбираем предложение $\bar{s}'_{i+1} = Wk_3k_4Tk_1V$, для которого справедлива цепочка $Wk_2Tk_1V \xrightarrow{P_3} Wk_3k_4Tk_1V \xrightarrow{P_2} Wk_3k_4Tv_1V$, т. е. $\bar{s}_{i-1} \xrightarrow{P_3} \bar{s}'_i \xrightarrow{P_2} \bar{s}_{i+1}$.

Во второй из указанных ситуаций выбираем предложение $\bar{s}'_i = Uk_1Wk_3k_4T$, для которого справедлива цепочка $Uk_1Wk_2T \xrightarrow{P_3} Uk_1Wk_3k_4T \xrightarrow{P_2} Uv_1Wk_3k_4T$, т. е. справедлива такая цепочка: $\bar{s}_{i-1} \xrightarrow{P_3} \bar{s}'_i \xrightarrow{P_2} \bar{s}_{i+1}$. Произведя подобную замену предложений в цепочке B несколько раз, получим цепочку вида

$$s^* \xrightarrow{P_3} \bar{s}'_2 \xrightarrow{P_3} \bar{s}'_3 \xrightarrow{P_3} \ldots \xrightarrow{P_3} \bar{s}'_n \xrightarrow{P_2} \bar{s}'_{n+1} \xrightarrow{P_2} \bar{s}'_{n+2} \xrightarrow{P_2} \ldots$$

$$\ldots \xrightarrow{P_2} \bar{s}'_{2n-1} \xrightarrow{P_2} \bar{s},$$

что доказывает вспомогательное утверждение.

Идея построения двумерной грамматики заключается в выборе поля зрения Tv в виде упорядоченного множества клеток $t_1, t_2, \ldots, t_n$ и поля описания Tk в виде множества клеток $\{\tau_{ij} : j = 1, 2, \ldots, i;\ i = 1, 2, \ldots, n\}$. Структура и локальные предикаты допустимости в этой грамматике должны быть выбраны таким образом, чтобы любой допустимый вариант s был таким, что $s(\tau_{11}) = k^*$, а для любого $i = 2, 3, \ldots, n$ пара фрагментов $(s(\tau_{i-1,1}), s(\tau_{i-1,2}), \ldots, s(\tau_{i-1,i-1}))$ и $(\underline{s}(\tau_{i1}), \underline{s}(\tau_{i2}), \ldots, \underline{s}(\tau_{ii}))$ представляла бы пару предложений $\bar{s}_{i-1}$ и $\bar{s}_i$, таких, что $\bar{s}_{i-1} \xrightarrow{P_3} \bar{s}_i$, и, наконец, пара фрагментов $(s(\tau_{n1}), s(\tau_{n2}), \ldots, s(\tau_{nn}))$ и $(s(t_1), s(t_2), \ldots, s(t_n))$ представляла бы пару предложений $\bar{s}_n$ и $\bar{s}$, таких, что $\bar{s}_n \overset{P_2}{\Rightarrow} \bar{s}$. Построенная далее двумерная грамматика удовлетворяет этим требованиям.

Поле зрения Tv и поле описания Tk этой грамматики уже определены выше. Алфавит сигналов должен быть равен терминальному алфавиту V. Алфавит $K_{\text{г}}$ структурных элементов должен быть выбран в виде множества $K \times \{\text{Л, З, П}\}$, где K — нетерминальный алфавит, а Л, З, П — начальные буквы слов «левый», «замена», «правый». Это значит, что если символ

k — один из нетерминальных символов, то среди структурных элементов будут символы $k_{Л}$, $k_{З}$ и $k_{П}$, отличающиеся нижним индексом. В дальнейшем будет использовано слово ind для обозначения переменной, принимающей значения Л, З, П. Таким образом, структурный элемент в двумерной грамматике есть пара (k, ind), где k — нетерминальный символ, а $\text{ind} \in \{Л, З, П\}$.

Структура $\mathcal{T}$ двумерной грамматики есть совокупность следующих множеств клеток:

$\{\tau_{11}\}$;

$\{\tau_{ij}, \tau_{ij+1}\}, \quad j = 1, 2, \ldots, i-1; \quad i = 2, 3, \ldots, n;$

$\{\tau_{i1}\}, \quad i = 2, 3, \ldots, n;$

$\{\tau_{ii}\}, \quad i = 2, 3, \ldots, n;$

$\{\tau_{ij}, \tau_{i+1,j}, \tau_{i+1,j+1}\}, \quad j = 1, 2, \ldots, i-1; \quad i = 1, 2, \ldots, n;$

$\{\tau_{nj}, t_j\}, \quad j = 1, 2, \ldots, n.$

Локальные предикаты $F(T')$, $T' \in \mathcal{T}$ должны быть выбраны следующим образом. При $T' = \{\tau_{11}\}$ предикат $F(T')$ должен равняться 1 только для символа $(k^*, З)$. Это означает, что вывод должен начинаться с предложения, состоящего из единственного символа k^*, который должен быть заменен, т. е. к которому должна быть применена подстановка из P_3.

При $T' = \{\tau_{ij}, \tau_{i,j+1}\}$, $j = 1, 2, \ldots, i-1$; $i = 2, 3, \ldots, n$, предикат $F(T')$ должен равняться единице на парах пар вида $((k_1, \text{ind}_1), (k_2, \text{ind}_2))$, таких, что пара $(\text{ind}_1, \text{ind}_2) \in \{(Л, Л), (Л, З), (З, П), (П, П)\}$. Такой выбор предиката $F(T')$ приводит к тому, что в допустимом варианте s для любого $i = 2, 3, \ldots, n$ последовательность $s(\tau_{i1})$, $s(\tau_{i2})$, $s(\tau_{i3})$, …, $s(\tau_{ii})$ содержит не более одного структурного элемента с индексом З (замена). Слева от него все структурные элементы имеют индекс Л (левый), а справа от него — индекс П (правый).

Предикат $F(T')$ для T' вида $\{\tau_{i1}\}$, $i = 2, 3, \ldots, n$, принимает единичное значение для всех структурных элементов вида (k, ind), где $\text{ind} \neq П$, и только для них; $F(T')$ для T' вида $\{\tau_{ii}\}$, $i = 2, 3, \ldots, n$, принимает единичное значение для всех структурных элементов вида (k, ind), где $\text{ind} \neq Л$. Вместе с предыдущим ограничением такое ограничение обеспечивает условие, что для любого $i = 2, 3, \ldots, n$ последовательность $s(\tau_{i1})$, $s(\tau_{i2})$, …, $s(\tau_{ii})$ содержит ровно один структурный элемент с индексом З.

Предикат $F(T')$ для фрагментов вида $\{\tau_{ij}, \tau_{i+1,j}, \tau_{i+1,j+1}\}$, $j = 1, 2, \ldots, i-1$; $j = 2, 3, \ldots, n$ допускает те и только те фрагменты $((k_{ij}, \text{ind}_{ij}), (k_{i+1,j}, \text{ind}_{i+1,j}), (k_{i+1,j+1}, \text{ind}_{i+1,j+1}))$, которые удовлетворяют следующим условиям:

1) если $\text{ind}_{ij} = Л$, то $k_{ij} = k_{i+1,j}$;
2) если $\text{ind}_{ij} = П$, то $k_{ij} = k_{i+1,j+1}$;
3) если $\text{ind}_{ij} = З$, то $(k_{ij}, k_{i+1,j}, k_{i+1,j+1}) \in P_3$.

Вследствие выполнения этих условий два фрагмента допустимого варианта, записанных в клетках $(\tau_{i1}, \tau_{i2}, \ldots, \tau_{ii})$ и $(\tau_{i+1,1}, \tau_{i+1,2}, \tau_{i+1,3}, \ldots, \tau_{i+1,i+1})$, $i = 2, 3, \ldots, n-1$, соответствуют двум предложениям $\bar{s}_i$ и $\bar{s}_{i+1}$, таким, что $\bar{s}_i \xrightarrow{P_3} \bar{s}_{i+1}$.

Предикат $F(T')$ для фрагментов вида $\{\tau_{nj}, t_j\}$, $j = 1, 2, \ldots, n$, должен допускать те и только те пары $(s(\tau_{nj}), s(t_j)) = (k_{nj}, \text{ind}_{nj}, v_j)$, для которых $(k_{nj}, v_j) \in P_2$. Вследствие последнего ограничения два фрагмента допустимого варианта, записанных в клетках $(\tau_{n1}, \tau_{n2}, \ldots, \tau_{nn})$ и $(t_1, t_2, \ldots, t_n)$, соответствуют двум предложениям $\bar{s}_n$ и $\bar{s}$, таким, что $\bar{s}_n \overset{P_2}{\Rightarrow} s$.

Мы построили двумерную грамматику, в которой допустимыми изображениями являются те и только те изображения, которые имеют длину n и являются предложениями в языке, порождаемом заданной контекстно-свободной грамматикой. Теорема доказана.

1.5. УНИВЕРСАЛЬНОСТЬ ДВУМЕРНЫХ ГРАММАТИК

Рассмотренные свойства двумерных грамматик позволяют сделать предположение, что они являются универсальным средством задания множеств изображений на конечном поле зрения Tv и конечном алфавите сигналов V. Это предположение является верным.

Теорема 11. *Для любого конечного поля зрения Tv, конечного алфавита V и множества $\mathcal{V} \subset V^{Tv}$ изображений $Tv \to V$ существует двумерная грамматика G, такая, что $\mathcal{V} = \mathcal{L}(G)$.*

Д о к а з а т е л ь с т в о. Определим поле описания искомой грамматики G как единственную клетку τ, а алфавит структурных элементов K как любое множество, состоящее из $|\mathcal{V}|$ элементов. Определим взаимно-однозначное соответствие между множеством изображений $\mathcal{V}$ и алфавитом K и обозначим через $k(v)$ структурный элемент, соответствующий изображению v, а через $v(k)$ — изображение, соответствующее структурному элементу k. Структуру $\mathcal{T}$ грамматики G определим как множество $\{\{\tau, t\} : t \in Tv\}$. Для каждого $t \in Tv$ предикат $F(T')$, $T' = \{\tau, t\}$ выберем таким образом, чтобы для каждого $k \in K$ он принимал значение 1 для одной и только одной пары $(k, v(k)(t))$. При выбранных таким способом предикатах $F(T')$, $T' \in \mathcal{T}$ для каждого $k \in K$ существует единственный допустимый в G вариант s, такой, что $s(\tau) = k$, причем сужение этого варианта на Tv есть изображение $v(k) \in \mathcal{V}$, которое является допустимым в G изображением. Таким образом, $v(k) \in \mathcal{L}(G)$

для любого $k \in K$. Поскольку множество $\{v(k) : k \in K\}$ по определению есть $\mathcal{V}$, из $\forall k \in K\, (v(k) \in \mathcal{L}(G))$ следует $\mathcal{V} \subset \mathcal{L}(G)$. Так как для любого $k \in K$ существует единственный допустимый вариант s, такой, что $s(\tau) = k$, то $|S(G)| = |K|$. В свою очередь, $|K| = |V|$. Следовательно, $|S(G)| = |\mathcal{V}|$. Очевидно, что $|\mathcal{L}(G)| \leqslant |S(G)|$, а следовательно, $|\mathcal{L}(G)| \leqslant |\mathcal{V}|$. Последнее неравенство совместно с доказанным ранее отношением $\mathcal{L}(G) \supset \mathcal{V}$ дает $\mathcal{L}(G) = \mathcal{V}$. Теорема доказана.

Выше отмечалось, что неуниверсальность какого-либо формального средства для задания множеств изображений мы не считаем существенным его недостатком, равно как и универсальность какого-либо другого средства — достоинством. Доказанная только что универсальность означает лишь тот факт, что для двумерных грамматик невозможно найти контрпример множества изображений. В худшем случае любое множество изображений можно выразить с помощью построенной в доказательстве теоремы грамматики, которая фактически равносильна простому запоминанию всех возможных допустимых изображений. Мы хотели бы доказать, что двумерные грамматики обладают не только принципиальной, но и практически реализуемой способностью выражать содержательно осмысленные зрительные понятия. Анализу таких возможностей двумерных грамматик посвящена следующая глава.

Глава 2

ПРЕДСТАВЛЕНИЕ МНОЖЕСТВ ИЗОБРАЖЕНИЙ СРЕДСТВАМИ ДВУМЕРНЫХ ГРАММАТИК

Для владения каким-либо языком, в том числе и формальным, требуются навыки двух типов. Необходимо знать грамматику этого языка, например синтаксис, или другие внутренние его формальные свойства. Навыки второго типа заключаются в практическом владении языком, т. е. в умении выражать средствами языка те или иные содержательные понятия. Эти два типа навыков владения языком по своему характеру являются существенно разными и означают, что можно владеть отличными приемами разговорного языка и не знать достаточно глубоко его формальных свойств, и наоборот. Такая асимметрия субъекта, владеющего языком, не является исключением, более того, исключением скорее является достаточно хорошее владение и той, и другой стороной языка. И если такая асимметрия является вполне допустимой для субъекта, владеющего языком, то она, конечно же, совершенно не допустима для самого языка, а именно: любое формальное средство должно быть достаточно богатым как в теоремном отношении, так и в содержательном.

Анализ формальных свойств двумерных грамматик, начатый в предыдущей главе, будет продолжен в последующих главах, а данная глава полностью посвящена вопросу о том, является ли аппарат двумерных грамматик в чисто прагматическом отношении удобным средством для выражения содержательно осмысленных зрительных понятий.

При любых сообщениях о возможностях автоматической обработки и распознавания изображений, которые открываются, если соответствующие множества изображений представить двумерной грамматикой (а эти возможности будут рассмотрены в следующих главах), неизменно возникал вопрос, откуда брать, как придумывать саму двумерную грамматику. Этот вопрос аналогичен вопросу о том, как писать программы, правильные не только синтаксически, но и по содержанию, т. е. правильно отображающие замысел программиста. Естественно, что любые рекомендации, данные в ответ на такой вопрос, неизбежно являются небезусловными, субъективными, нечеткими, неполными и т. п. Однозначные рекомендации по поводу конструирования двумерных грамматик возможны лишь при существовании неких формальных средств для выражения замысла конструктора распознающей системы. В этом случае наиболее явным выражением однозначных рекомендаций по построению двумерных грамматик было бы построение транслятора, имеющего на выходе двумерную грамматику, построенную по входному предложению, которое формально выражает замысел конструктора и является заданием на построение грамматики. Такое решение проблемы является лишь кажущимся ответом на вопрос, как строить двумерную грамматику, так как вопрос о практической приемлемости данных двумерных грамматик лишь заменяется вопросом о практической приемлемости других формальных средств.

Таким образом, в данной главе мы ставим вопрос, на который принципиально не может быть дан положительный ответ в рамках чистого математического исследования, а могут быть приведены лишь более или менее убедительные соображения в пользу положительного ответа. С этой целью приведем ряд постепенно усложняющихся примеров содержательно осмысленных множеств зрительных изображений и представим эти множества средствами двумерных грамматик. В каждом из примеров множество изображений первоначально будет определено с помощью как можно более однозначных предложений русского языка. Затем для каждого из рассмотренных множеств будет построена двумерная грамматика, не слишком сложная в том смысле, что формальное ее задание не требует слишком большой памяти. Конечно, рассмотрение даже очень большого количества подобных примеров не гарантирует, что с помощью двумерных грамматик так же экономно может быть представлено любое множество. В действительности же ситуация здесь еще более жесткая. А именно: если существует хотя бы одно-единственное множество, которое может быть удобно выражено средствами двумерных грамматик, то неизбежно существует другое множество, которое не может быть удобно выражено этими же средствами. Такое свойство двумерных грамматик не является их отличительным свойством, а присуще любому формальному средству задания множеств. Непонимание или игнорирование универсальности этого свойства встречается значительно чаще, чем хотелось бы, поэтому в следующем параграфе приводятся его конкретная формулировка и доказательство.

2.1. РАВНОМЕРНО НАИЛУЧШИЙ СПОСОБ ЗАДАНИЯ МНОЖЕСТВ

Пусть Tv (поле зрения) и V (алфавит сигналов) — два конечных множества; V^{Tv} — множество всех возможных изображений, т. е. функций вида $Tv \to V$; $\tilde{V}$ — множество $2^{V^{Tv}}$, т. е. множество всех возможных подмножеств изображений.

И пусть $\{0, 1\}^*$ — множество всех возможных последовательностей конечной длины над алфавитом $\{0, 1\}$. Для каждой последовательности $c \in \{0, 1\}^*$ через $d(c)$ обозначим длину последовательности c.

Формальным средством для задания подмножества изображений будем называть подмножество $C \subset \{0, 1\}^*$ с заданной на этом подмножестве функцией $Q : C \to \tilde{V}$.

Формальное средство для задания подмножества изображений будем называть универсальным, если для любого $\mathcal{V} \in \tilde{V}$ справедливо неравенство $\{c : c \in C,\ Q(c) = \mathcal{V}\} \neq \varnothing$.

Для заданного формального средства и заданного подмножества $\mathcal{V} \in \tilde{V}$ изображений через $D(\mathcal{V})$ обозначим минимальную длину последовательности c из последовательностей, для которых $Q(c) = \mathcal{V}$, т. е.

$$D(\mathcal{V}) = \min_{\{c: c \in C, Q(c) = \mathcal{V}\}} d(c).$$

Определим формальное средство для задания множеств изображений, называемое в дальнейшем стандартным. Пусть $k = |V|^{|Tv|}$ — количество всех возможных изображений. Перенумеруем изображения числами от 1 до k и определим множество $C \subset \{0, 1\}^*$ для стандартного средства как множество последовательностей длины k. Функция Q для стандартного средства должна быть такой, что для любой последовательности $c = \alpha_1, \alpha_2, \ldots, \alpha_k$ изображение с номером i входит в множество $\mathcal{V} = Q(c)$ тогда и только тогда, когда $\alpha_i = 1$.

Стандартное средство является универсальным, причем таким, что $D(\mathcal{V}) = k$ для любого $\mathcal{V} \subset V^{Tv}$. Недостаток его очевиден и заключается в чрезвычайно большом объеме памяти, необходимом для представления любого множества. В определенном смысле слова не существует лучшего универсального средства задания множеств, лучшего, чем стандартный. Более определенную формулировку этого утверждения представляет следующая теорема.

Теорема 1. *Пусть для некоторого универсального средства задания множества изображений существует множество $\mathcal{V}_1 \in \tilde{V}$, такое, что $D(\mathcal{V}_1) < k$. В этом случае существует такое множество $\mathcal{V}_2 \in \tilde{V}$, что $D(\mathcal{V}_2) > k$.*

Д о к а з а т е л ь с т в о. Пусть $C \subset \{0, 1\}^*$ — множество последовательностей, соответствующее средству задания множеств, о котором идет речь в теореме, а Q — соответствующая функция $C \to \tilde{V}$. Так как любое подмножество $\mathcal{V}$ из V может быть выражено некоторой последовательностью из C, то очевидно, что

$$|C| \geqslant |\tilde{V}| = 2^k. \tag{2.1}$$

Допустим, что теорема неверна, т. е. существует такое подмножество $C^* \subset C$, что для любого $\mathcal{V} \in \tilde{V}$ существует такое $c \in C^*$, при котором $Q(c) = \mathcal{V}$, причем $d(c) \leqslant k$, и существует $c_1 \in C^*$, такое, что $Q(c_1) = \mathcal{V}_1$, причем $d(c_1) < k$.

Множество C^* может быть представлено в виде объединения $C_1 \cup C_2 \cup \ldots \cup C_l$, C_i, $i = 1, 2, \ldots, l$,— множество последовательностей, имеющих длину n_i, $n_1 < n_2 < \ldots < n_l \leqslant k$.

Множество C_1 состоит из последовательностей длины n_1, количество которых не превосходит 2^{n_1}. Это значит, что существует такое число k_1, при котором

$$0 \leqslant k_1 < 2^{n_1}, \tag{2.2}$$

$$|C_1| = 2^{n_1} - k_1. \tag{2.3}$$

Множество C_2 состоит из последовательностей длины n_2, начальный участок которых не совпадает ни с одной последовательностью из C_1. Поэтому $|C_2| \leqslant k_1 \cdot 2^{n_2 - n_1}$, что означает существование такого числа k_2, при котором

$$0 \leqslant k_2 < k_2 2^{n_2 - n_1}, \tag{2.4}$$

$$|C_2| = k_1 2^{n_2 - n_1} - k_2. \tag{2.5}$$

И вообще введя обозначения $n_0 = 0$, $k_0 = 1$, можно записать для любого $i = 1, 2, \ldots, l$

$$|C_i| = k_{i-1} \cdot 2^{n_i - n_{i-1}} - k_i, \tag{2.6}$$

$$0 \leqslant k_i < k_{i-1} \cdot 2^{n_i - n_{i-1}}. \tag{2.7}$$

Докажем, что для любого $i = 1, 2, \ldots, l$ справедливо неравенство

$$0 \leqslant k_i < 2^{n_l}. \tag{2.8}$$

Для $i = 1$ неравенство (2.8) справедливо в силу (2.3). Допустим, что (2.8) справедливо для всех i, меньших некоторого i^*, и покажем, что на этом основании оно выполняется и для $i = i^*$.

На основании (2.7) справедливо неравенство

$$k_{i^*} < k_{i^*-1} \cdot 2^{n_{i^*} - n_{i^*-1}}. \tag{2.9}$$

По предположению для k_{i^*-1} выполняется (2.8), т. е.

$$k_{i^*-1} < 2^{n_{i^*-1}}. \tag{2.10}$$

Подставляя правую часть (2.10) в выражение (2.9), получаем $k_{i^*} < 2^{n_{i^*-1}} \cdot 2^{n_{i^*} - n_{i^*-1}} = 2^{n_{i^*}}$, что доказывает (2.8) и для $i = i^*$. Неравенство (2.8), таким образом, справедливо для всех $i = 1, 2, \ldots, l$.

Запишем следующую цепочку равенств и неравенств:

$$|C^*| = \sum_{i=1}^{l} |C_i| = \sum_{i=1}^{l} (k_{i-1} \cdot 2^{n_i - n_{i-1}} - k_i) =$$

$$= 2^{n_1} + \sum_{i=2}^{l} k_{i-1} \cdot (2^{n_i - n_{i-1}} - 1) - k_l < 2^{n_1} +$$

$$+ \sum_{i=2}^{l} (2^{n_i} - 2^{n_{i-1}}) - k_l < 2^{n_1} + \sum_{i=2}^{l} (2^{n_i} - 2^{n_{i-1}}) = 2^{n_l} \leqslant 2^k. \tag{2.11}$$

В этой цепочке первое равенство справедливо, так как множества $C_1, C_2, \ldots, C_l$ попарно не пересекаются. Второе равенство следует из (2.6). Третье равенство получено в результате перегруппировки слагаемых под знаком суммы. Следующие затем два неравенства выполняются в результате (2.8). Оставшиеся два преобразования очевидны.

Следствием цепочки (2.11) является неравенство $|C^*| < 2^k$, противоречащее (2.1). Поэтому предположение о том, что теорема неверна, является ложным. Теорема доказана.

Из доказанной теоремы следует, что для любого универсального формального средства задания множества $\max\limits_{\mathcal{V} \in \tilde{V}} D(\mathcal{V}) \geqslant k$. Поэтому введенное ранее стандартное средство задания множества является равномерно наилучшим, так как для него $\max\limits_{\mathcal{V} \in \tilde{V}} D(\mathcal{V}) = k$.

Таким образом, мы видим, что универсальность не является критерием практической приемлемости того или иного формального средства задания множеств изображений, так как ортодоксальное следование этому критерию приводит к созданию средства, эквивалентного стандартному, с его вполне очевидными недостатками. Практическая приемлемость должна определяться возможностью задания множеств из некоторого формально не заданного класса множеств.

И хотя, как уже сказано ранее, доказательство такой практической приемлемости принципиально не может быть выполнено чистыми математическими методами, тем или иным способом должно быть выполнено семантическое апробирование любого средства, претендующего на формализацию ранее неформализованных понятий. Основным недостатком известных формальных конструкций [134, 136—140], подобных формально-грамматическим, но порождающим двумерные массивы, является именно отсутствие достаточного списка примеров, устанавливающих соответствие между реальными множествами изображений и языками, которые задаются этими формальными средствами. В силу этого упомянутые работы хотя и содержат точные математические результаты, тем не менее не оказывают заметного влияния на практику построения распознающих систем. Данная глава введена в монографию для того, чтобы этого недостатка избежать.

2.2. ИЗОБРАЖЕНИЯ В РАСТРОВОМ ПРЕДСТАВЛЕНИИ

1. По-видимому, простейшим зрительным понятием является кривая линия, разделяющая поле зрения на две части (левую и правую) и идущая снизу вверх. Определим множество изображений, соответствующих этому понятию, с помощью обычных предложений русского языка. Поле зрения Tv (размером $m \times n$, m и n — целые числа) определим как множество пар $\{(i, j) : 1 \leqslant i \leqslant n,\ 1 \leqslant j \leqslant m\}$. Число i обозначает номер строки поля зрения. Самая нижняя строка имеет номер 1. Число j обозначает номер столбца поля зрения. Самый левый столбец имеет номер 1. Алфавит сигналов определим как множество $\{0, 1\}$. Упорядоченное множество клеток $\{(1, j_1), (2, j_2), (3, j_3), \ldots, (n, j_n)\}$ будем считать допустимой траекторией, если $|j_i - j_{i-1}| \leqslant 1$ для всех $i = 2, 3, \ldots, n$. Изображение $v : Tv \to \{0, 1\}$ будем считать допустимым, если существует такая допустимая траектория T, что равенство $v(t) = 1$ выполняется для всех $t \in T$ и только для них. На рис. 2.1, *а*, *б* представлены примеры допустимых изображений, а на рис. 2.1, *в*, *г* — недопустимых.

Определим теперь двумерную грамматику, которая порождает указанное множество изображений. Поле описания Tk в этой грамматике есть множество $\{\tau(t) : t \in Tv\}$, изоморфное полю зрения Tv; через $\tau(t)$ обозначена клетка поля описания, соответствующая клетке $t \in Tv$ в этом изоморфизме.

Структуру $\mathcal{T}$ грамматики представим в виде множества

$$\mathcal{T} = \{\{\tau(t), t\} : t \in Tv\} \cup$$

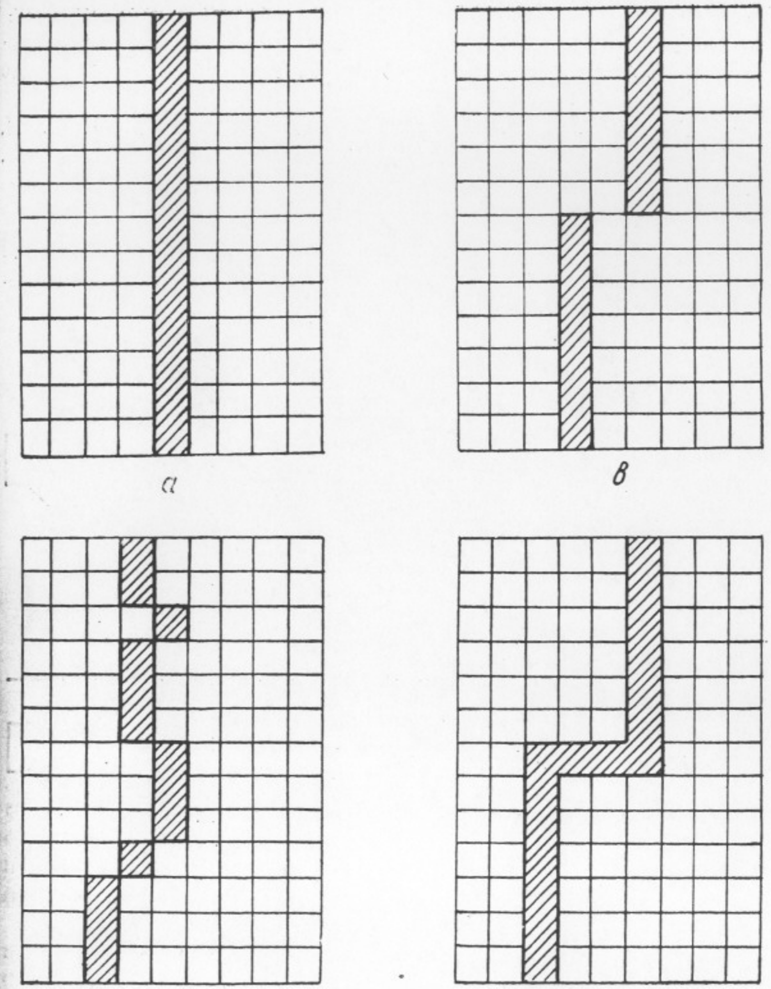


Рис. 2.1. Примеры допустимых и недопустимых кривых линий

$$\cup \{\{\tau(i, j), \tau(i + 1, j)\} : i = 1, 2, \ldots, n - 1;$$
$$j = 1, 2, \ldots, n\} \cup$$
$$\cup \{\{\tau(i, j), \tau(i, j + 1)\} : i = 1, 2, \ldots, n;$$
$$j = 1, 2, \ldots, n - 1\}.$$

Поле зрения Tv, поле описания Tk и структура $\mathcal{T}$ конструируемой грамматики представлены на рис. 2.2, где большими клетками изображены клетки поля описания, маленькими клетками — клетки поля зрения. Структура $\mathcal{T}$ представлена в виде линий, соединяющих клетки t и t', таких, что $\{t, t'\} \in \mathcal{T}$.

Структурный алфавит состоит из трех символов Л, К, П, являющихся начальными буквами в словах «левая», «кривая», «правая».

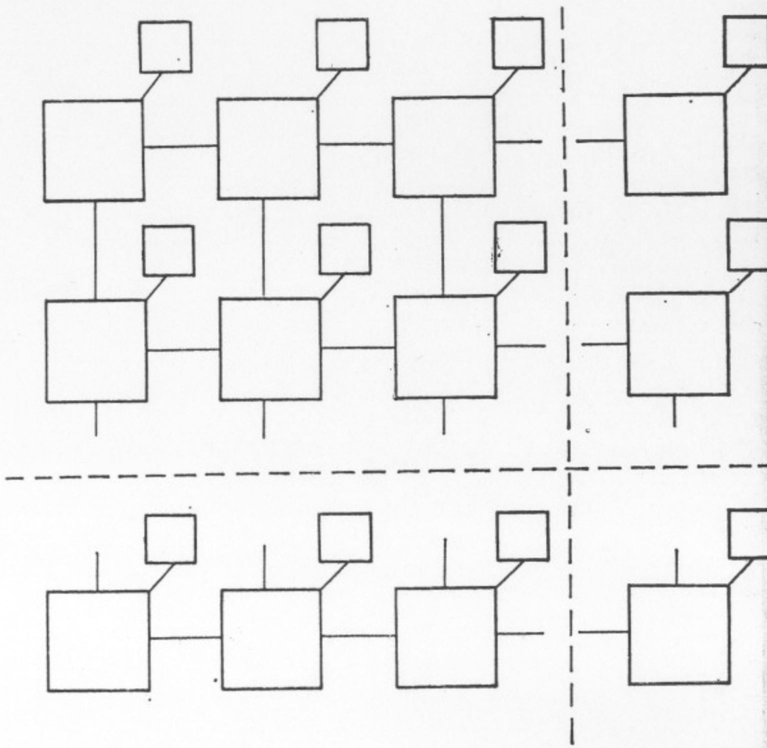


Рис. 2.2. Пример структуры грамматики

Совокупность предикатов $F(T')$, $T' \in \mathcal{T}$ определяем следующим образом.

Для $T' = \{\tau(t), t\}$, $t \in Tv$, предикат $F(T')$ должен равняться единице только для пар (Л, 0), (К, 1), (П, 0). Это означат, что если в клетке t поля зрения записан 0, то в соответстеующей клетке $\tau(t)$ поля описания должен быть записан либо символ Л, либо символ П. Если же в клетке $t \in Tv$ записан символ 1, то в клетке $\tau(t) \in Tk$ должен быть записан символ K.

Для $T' = \{\tau(i, j), \tau(i+1, j)\}$, т. е. для двух клеток поля описания, из которых одна находится над другой, предикат $F(T')$ запрещает только пары символов $\begin{pmatrix} Л \\ П \end{pmatrix}$ и $\begin{pmatrix} П \\ Л \end{pmatrix}$ и допускает в се остальные пары.

Для $T' = \{\tau(i, j), \tau(i, j+1)\}$, т. е. для двух клеток, из которых одна находится справа от другой, предикат должен допускать только четыре пары символов (Л, Л), (Л, К), (К, П), (П, П). Это означает, что в каждой строке поля описания находится не более одного символа К, причем слева от него будут расположены только символы Л, а справа — только

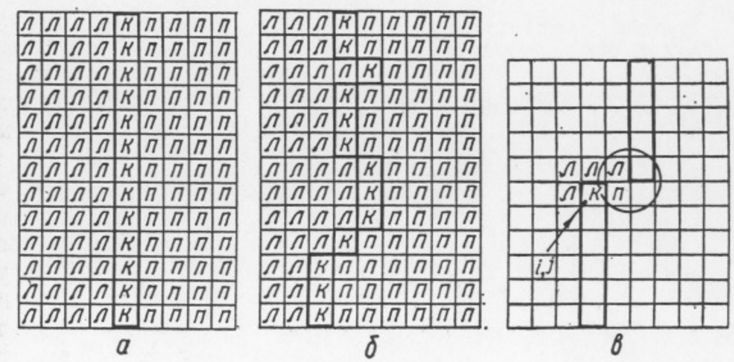


Рис. 2.3. К доказательству допустимости ($а$, $б$) и недопустимости изображения ($в$)

символы П. Изображения на рис. 2.1, $а$ и $б$ допустимы в построенной грамматике, доказательством чему являются допустимые варианты, содержащие эти изображения и представленные на рис. 2.3, $а$ и $б$.

Покажем недопустимость изображения, представленного на рис. 2.3, $в$. Пусть клетка, обозначенная на этом рисунке стрелкой, имеет координаты (i, j). Сигнал в этой клетке равен единице, поэтому если существует допустимый вариант s, то $s(\tau(i, j)) = К$. В силу введенных предикатов $F(T')$ $s(\tau(i, j-1)) = Л$, а $s(\tau(i, j+1)) = П$. В клетке $(i + 1, j)$ $v(i+1, j) = 0$, поэтому для допустимого варианта символ $s(\tau(i+1, j))$ равен либо Л, либо П. Однако, в клетке $\tau(i+1, j)$ возможен лишь символ Л, так как пара $\begin{pmatrix} П \\ Л \end{pmatrix}$ не является допустимой. Подобным образом приходим к выводу, что в клетке $\tau(i+1, j)$ также возможен лишь символ Л, а в клетке $\tau(i+1, j+1)$ — лишь символ Л. Таким образом, мы видим, что если существует допустимый вариант s, то в паре клеток $\begin{pmatrix} \tau(i+1, & j+1) \\ \tau(i, & j+1) \end{pmatrix}$ он должен принимать пару значений $\begin{pmatrix} Л \\ П \end{pmatrix}$, а эта пара недопустима. Следовательно, изображение на рис. 2.1, *в* не является допустимым в построенной грамматике.

Недопустимость изображения на рис. 2.1, *г* доказывается значительно проще. Построенная грамматика легко может быть видоизменена так, что допустимыми изображениями в ней будут не непрерывные кривые, идущие снизу вверх, как в только что рассмотренном случае, а пятна, граница которых есть допустимая траектория, как определено выше. Такое множество изображений рассматривается в работе А. Мартелли [131] как модель понятия «контур изображения». Примеры допустимых изображений такого рода приведены на рис. 2.4, *а*, *б*, а недопустимых — на рис. 2.4, *в*, *г*.

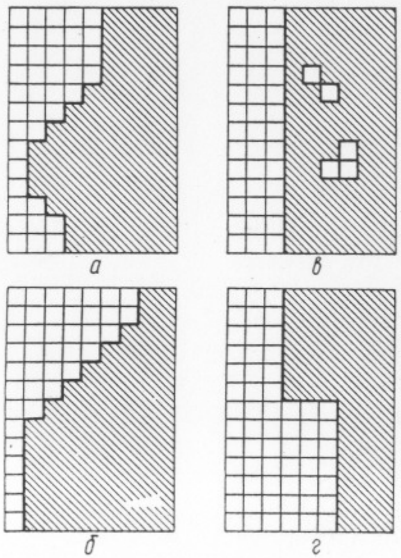


Рис. 2.4. Примеры допустимых (*а*, *б*) и недопустимых (*в*, *г*) изображений контуров

Для задания такого множества изображений достаточно в только что построенной грамматике изменить предикаты $F(T')$ для фрагментов вида $\{\tau\ (t), t\}$, $t \in Tv$. Эти предикаты определяют соответствие между значением сигнала, помещаемого в клетку t поля зрения, и значением структурного элемента в соответствующей клетке $\tau(t)$ поля описания. Если для задания кривой этот предикат должен был объявлять допустимыми только пары (Л, 0), (К, 1), (П, 0), то в данном случае он должен допускать только пары (Л, 0), (К, 1), (П, 1). Далее будут рассмотрены дополнительные усложнения этой грамматики.

2. Рассмотрим грамматику, определяющую понятие «кривая полоса постоянной ширины». Пусть заданы две траектории — $T_Л$ и $T_П$, допустимые в смысле, указанном в предыдущем примере. Одну из этих траекторий будем называть левой, а

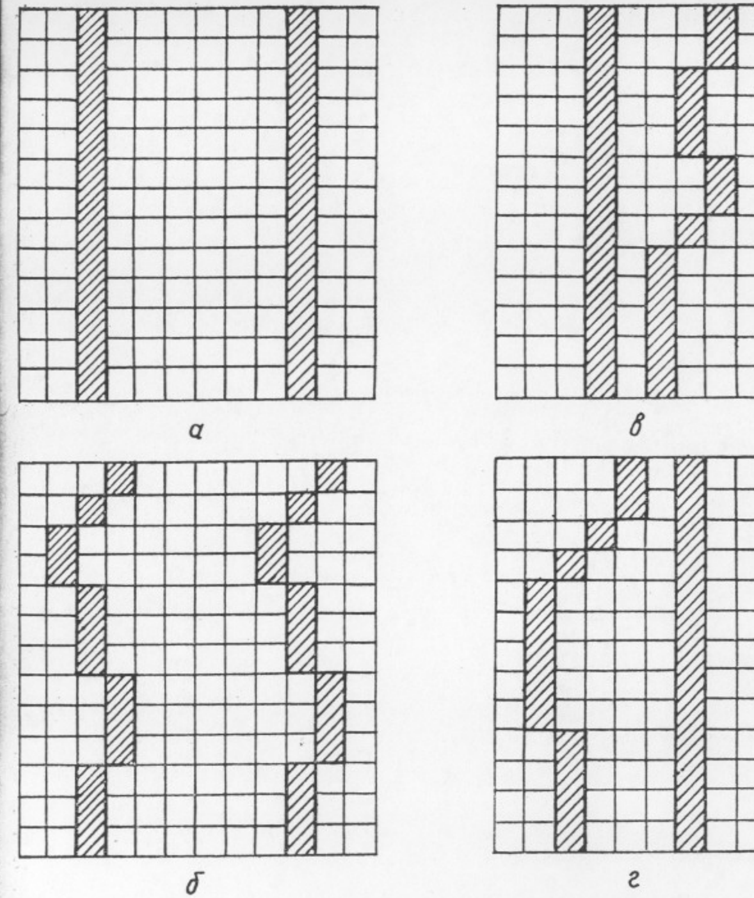


Рис. 2.5. Примеры допустимых (*а*, *б*) изображений кривых полос постоянной ширины и недопустимых (*в*, *г*)

вторую — правой. Это значит, что

$$T_Л = \{(i, j_{Лi}) : | j_{Лi} - j_{Лi-1} | \leqslant 1,\ i = 2, 3, \ldots, n\},$$

$$T_П = \{(i, j_{Пi}) : | j_{Пi} - j_{Пi-1} | \leqslant 1; \quad i = 2, 3, \ldots, n\}.$$

Пару траекторий ($T_Л$, $T_П$) будем считать допустимой, если для любых значений i_1 и i_2 выполняется равенство $(j_{Пi_1} - j_{Лi_1}) = (j_{Пi_2} - j_{Лi_2})$, кроме того, $(j_{П_1} - j_{Л_1}) \geqslant 2$. Изображение v будем считать допустимым, если существует допустимая пара траекторий $T_Л$ и $T_П$, такая, что равенство $v\ (t) = 1$ выполняется для всех $t \in T_Л \cup T_П$ и только для них.

На рис. 2.5, *а*, *б* представлены примеры допустимых изображений, а на рис. 2.5, *в*, *г* — недопустимых. Допустимые множестве реальных вертикальных линий, а не просто о вертикальных линиях, мы имеем в виду, что допустимыми будут не только вертикальные линии в строгом смысле этого слова, но и линии, проведенные человеком, когда он от руки проводит линию, намереваясь начертить вертикальную линию. Практически оправдавшееся определение реальной вертикальной линии заключается в следующем [61]. Реальная вертикальная линия — это упорядоченное множество $\{(i, j_i), i = 1, 2, \ldots, n\}$ клеток поля зрения, такое, что $|j_i - j_{i-1}| \leqslant 1$ для всех i, кроме того, для любых i_1 и i_2, таких, что $i_2 - i_1$ — четное, выполняется неравенство $|j_{i_2} - j_{i_1}|/|i_2 - i_1| \leqslant 2$. На рис. 2.7, *а* приведен пример реальной вертикальной линии, а на рис. 2.7, *б* — пример недопустимой линии.

Л	Л	Л	КЛ КП	П	П	П	П	П	П	П
Л	Л	Л	КЛ КП	П	П	П	П	П	П	П
Л	Л	Л	КЛ	П	П	П	П	П	П	П
Л	Л	Л	Л	КП	П	П	П	П	П	П
Л	Л	Л	Л	КЛ	П	П	П	П	П	П
Л	Л	Л	Л	Л	КП	П	П	П	П	П
Л	Л	Л	Л	Л	КЛ КП	П	П	П	П	П
Л	Л	Л	Л	Л	КЛ КП	П	П	П	П	П
Л	Л	Л	Л	Л	КП	П	П	П	П	П
Л	Л	Л	Л	КЛ	П	П	П	П	П	П
Л	Л	Л	Л	КП	П	П	П	П	П	П
Л	Л	Л	КЛ	П	П	П	П	П	П	П
Л	Л	Л	КЛ КП	П	П	П	П	П	П	П
Л	Л	Л	КЛ	П	П	П	П	П	П	П
Л	Л	Л	Л	КП	П	П	П	П	П	П

Рис. 2.8. Допустимые варианты, содержащие изображение 2.7, *а*.

В грамматике, порождающей указанное множество изображений, поле зрения, поле описания, структура и алфавит сигналов такие же, как и в предыдущих примерах. Структурный алфавит состоит из символов Л (левое), КЛ (клетка линии, выступающая влево), КП (клетка линии, выступающая вправо), П (правое).

Предикат $F(T')$ для T' вида $\{\tau(t), t\}$, $t \in Tv$ допускает пары (Л, 0), (КЛ, 1), (КП, 1), (П, 0). Это означает, что белые клетки изображения бывают двух типов: первому из них соответствует структурный элемент Л, а второму — П. Клетки, принадлежащие линии, т. е. черные клетки, также бывают двух типов: одному из них соответствует символ КЛ, а второму — КП.

Предикат $F(T')$ для T' вида $\{\tau(i, j), \tau(i, j+1)\}$, т. е. пар клеток, соседствующих в горизонтальном направлении, допускает пары символов (Л, Л), (Л, КЛ), (Л, КП), (КЛ, П), (КП, П), (П, П) и означает, что в каждой строке поля описания возможен не более чем один символ из символов КЛ и КП.

Предикат $F(T')$ для T' вида $\left\{\begin{matrix}\tau(i+1, j)\\ \tau(i, j)\end{matrix}\right\}$, т. е. пар клеток, соседствующих по вертикали, допускает следующие пары символов:

$$\begin{pmatrix}\text{Л}\\ \text{Л}\end{pmatrix}, \begin{pmatrix}\text{Л}\\ \text{КЛ}\end{pmatrix}, \begin{pmatrix}\text{КЛ}\\ \text{КЛ}\end{pmatrix}, \begin{pmatrix}\text{КЛ}\\ \text{КП}\end{pmatrix}, \begin{pmatrix}\text{КЛ}\\ \text{Л}\end{pmatrix}, \begin{pmatrix}\text{КП}\\ \text{КП}\end{pmatrix},$$
$$\begin{pmatrix}\text{КП}\\ \text{КЛ}\end{pmatrix}, \begin{pmatrix}\text{КП}\\ \text{П}\end{pmatrix}, \begin{pmatrix}\text{П}\\ \text{КП}\end{pmatrix}, \begin{pmatrix}\text{П}\\ \text{П}\end{pmatrix}.$$

Последние ограничения как раз и обеспечивают приблизительную вертикальность линии на допустимом изображении. Допустимый вариант, соответствующий изображению на рис. 2.7, *а*, представлен на рис. 2.8. Тот факт, что в некоторых клетках помещены несколько структурных элементов, означает, что для данного изображения существует несколько допустимых вариантов.

1	1	1	1	1	1	1
1	1	3	10	10	9	1
1	1	4	2	2	8	1
1	1	4	2	2	8	1
1	1	4	2	2	8	1
1	1	4	2	2	8	1
1	1	4	2	2	8	1
1	1	5	6	6	7	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1

Рис. 2.9. Допустимый вариант в грамматике, порождающей изображения буквы П

4. Рассмотрим построение двумерных грамматик для изображений некоторых букв. Идея построения грамматики изображений буквы П показана на рис. 2.9, где представлен пример допустимого в этой грамматике варианта. Структура грамматики та же, что и в предыдущих примерах, алфавит структурных элементов равен $\{1, 2, \ldots, 10\}$, а алфавит сигналов — $\{\square, \blacksquare\}$. По рисунку видно, какой содержательный смысл имеют выбранные структурные элементы. Например, 3 — это верхний левый угол буквы П, *8* — правая вертикальная линия и т. д. Предикат допустимости для пар клеток $\{\tau(t), t\}$, $t \in Tv$ объявляет допустимыми пары $(1, \square)$, $(2, \square)$, $(6, \square)$, $(3, \blacksquare)$, $(4, \blacksquare)$, $(5, \blacksquare)$, $(7, \blacksquare)$, $(8, \blacksquare)$, $(9, \blacksquare)$, $(10, \blacksquare)$.

Для соседствующих по горизонтали клеток допустимыми являются пары (1, 3), (1, 4), (1, 5), (1, 1), (2, 2), (2, 8), (3, 10), (4, 2), (5, 6), (6, 6), (6, 7), (7, 1), (8, 1), (9, 1), (10, 10), (10, 9), для соседствующих по вертикали — пары $\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}1\\3\end{pmatrix}, \begin{pmatrix}1\\10\end{pmatrix}$,

$$\begin{pmatrix}1\\9\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}, \begin{pmatrix}2\\6\end{pmatrix}, \begin{pmatrix}3\\4\end{pmatrix}, \begin{pmatrix}4\\4\end{pmatrix}, \begin{pmatrix}4\\5\end{pmatrix}, \begin{pmatrix}6\\1\end{pmatrix}, \begin{pmatrix}7\\1\end{pmatrix}, \begin{pmatrix}8\\8\end{pmatrix}, \begin{pmatrix}8\\7\end{pmatrix}, \begin{pmatrix}9\\8\end{pmatrix}, \begin{pmatrix}10\\2\end{pmatrix}.$$

Легко понять, что по примеру этой грамматики нетрудно построить грамматику, порождающую изображения прямоугольников. Для этого необходимо вместо пары (6, □), считающейся допустимой в только что построенной грамматике, ввести допустимую пару (6, ■). Это значит, что структурному элементу 6 должен соответствовать не сигнал □ (белое), а сигнал ■ (черное).

Подобным образом можно построить и грамматику, порождающую стилизованные изображения буквы С.

5. В рассмотренных в предыдущем примере грамматиках допустимыми были не только изображения одной буквы, но и изображения, при которых в поле зрения находилось несколько изображений одной и той же буквы, занимающих различные участки. Пример такого изображения приведен на рис. 2.10. Этого нетрудно избежать, и в данном примере мы укажем именно такие грамматики, при которых в поле зрения возможно одно и только одно изображение. Кроме того, в данном примере иллюстрируется тот факт, что поле описания совсем не обязательно изоморфно полю зрения. И, наконец, будет указан единообразный прием построения грамматик для различных буквенных изображений. Конструируемая ниже грамматика порождает множество изображений, рассмотренных в работе Т. К. Винцюка [25].

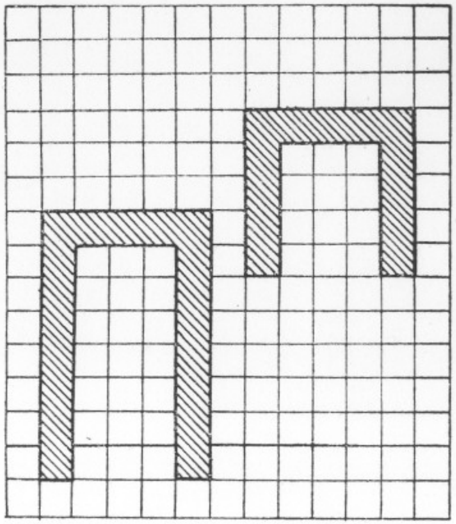


Рис. 2.10. Допустимое изображение в грамматике, построенной в примере 4.

Пусть поле зрения Tv равно множеству $\{(i, j) : 1 \leqslant i \leqslant n; 1 \leqslant j \leqslant n\}$, алфавит сигналов {□, ■}. Поле описания определим как совокупность из $m + n$ клеток, состоящую из двух групп клеток. Клетки первой группы соответствуют строкам изображения, и через $\tau_{стр}(i)$ обозначена клетка, соответствующая i-й строке, $i = 1, 2, \ldots, n$. Клетки второй группы соответствуют столбцам изображения, и через $\tau_{столб}(j)$ обозначена клетка, соответствующая j-му столбцу, $j = 1, 2, \ldots, m$. Поле описания искомой грамматики есть множество $\{\tau_{стр}(i) : 1 \leqslant i \leqslant n\} \cup \{\tau_{столб}(j), 1 \leqslant j \leqslant m\}$. Алфавит структурных элементов определим как множество $\{1, 2, \ldots, 7\}$. Структура $\mathcal{T}$ грамматики есть структура третьего порядка и включает:

1) четыре множества $\{\tau_{стр}(1)\}$, $\{\tau_{стр}(n)\}$, $\{\tau_{столб}(1)\}$, $\{\tau_{ст}(m)\}$, каждое из которых содержит единственную клетку;

2) $n - 1$ множеств $\{\tau_{стр}(i), \tau_{стр}(i + 1)\}$, $i = 1, 2, \ldots, n - 1$, каждое из которых состоит из двух клеток;

3) $m - 1$ множеств $\{\tau_{столб}(j), \tau_{столб}(j + 1)\}$, $j = 1, 2, \ldots, m - 1$, каждое из которых состоит из двух клеток;

4) $m \times n$ множеств $\{(i, j), \tau_{стр}(i), \tau_{столб}(j)\}$, $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m$, каждое из которых состоит из трех клеток.

Локальные предикаты допустимости выбираются таким образом, что:

1) в клетках $\tau_{стр}(1)$ и $\tau_{столб}(1)$ является допустимым лишь символ 1, а в клетках $\tau_{стр}(n)$ и $\tau_{столб}(m)$ — лишь символ 7;

2) в паре клеток второго и третьего типов допускаются лишь пары (1, 1), (1, 2), (2, 3), (3, 3), (3, 4), (4, 5), (5, 5), (5, 6), (6, 7), (7, 7);

3) в тройке клеток вида $\{(i, j), \tau_{стр}(i), \tau_{столб}(j)\}$ допустимые тройки символов вида $(v, k_{стр}, k_{столб})$ определяются так, чтобы для каждой пары $k_{стр}$, $k_{столб}$ существовал единственный символ $v(k_{стр}, k_{столб})$, такой, что тройка $(v(k_{стр}, k_{столб}), k_{стр}, k_{столб})$ является допустимой. Иными словами, множество допустимых троек вида $(v, k_{стр}, k_{столб})$ выбирается так, что оказывается определенной некоторая функция $\varphi : \{1, 2, \ldots, 7\} \times \{1, 2, \ldots, 7\} \to$ {□, ■}.

Грамматики для классов изображений, соответствующих различным буквам, отличаются друг от друга выбором именно функции φ. На рис. 2.11 в виде таблиц 7×7 представлены функции φ, соответствующие различным стилизованным изображениям русских букв.

Процесс порождения допустимого изображения можно представить следующим образом. Вначале в соответствии с ограничениями 1 и 2 строятся две последовательности: $S_{стр} = \{s(\tau_{стр}(i)), i = 1, 2, \ldots, n\}$ и $S_{ст} = \{s(\tau_{столб}(j)), j = 1, 2, \ldots, m\}$, каждая из которых является монотонно неубывающей последовательностью, начинающейся символом 1, а кончающейся числом 7, причем каждый из символов 2, 4 и 6 входит в каждую из последовательностей только один раз. Последовательности $S_{стр}$ и $S_{ст}$ как бы разбивают поле зрения на прямоугольные участки, которые затем в соответствии с функцией φ объявляются либо черными, либо белыми.

6. Рассмотренные в предыдущем примере изображения букв и цифр могут считаться идеальными, так как они составлены из идеальных горизонтальных и вертикальных линий, а не из реальных, которые производит оператор, намеревающийся

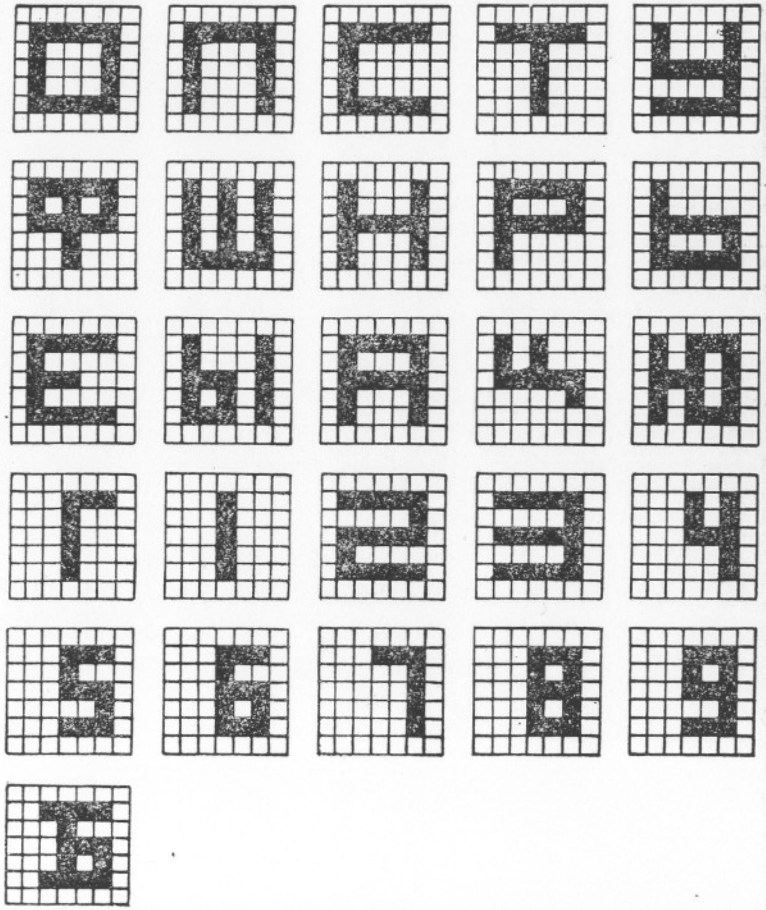


Рис. 2.11. Функции φ для различных алфавитно-цифровых знаков

изобразить букву, составленную из горизонтальных и вертикальных линий. В данном примере будут рассмотрены грамматики, порождающие изображения букв и цифр и составленные из реальных горизонтальных и вертикальных линий. Кроме того, при этом рассмотрении будет введен полезный прием построения грамматик.

Введем в рассмотрение две вспомогательные грамматики — $G^{стр}$ и $G^{столб}$. Пусть для заданных целых чисел m и n множество Tv есть множество пар $\{(i, j) : 1 \leqslant i \leqslant n;\ 1 \leqslant j \leqslant \leqslant m\}$.

Поле зрения $Tv^{стр}$ грамматики $G^{стр}$ определим как множе-

ство, изоморфное множеству Tv, причем клетка, обозначаемая $\tau^{стр}(t) \in Tv^{стр}$, соответствует клетке $t \in Tv$. Поле зрения $Tv^{ст}$ грамматики $G^{столб}$ определим так же, как изоморфное Tv, причем $\tau^{столб}(t) \in Tv^{столб}$ обозначает клетку, соответствующую клетке $t \in Tv$. Таким образом, $Tv^{стр} = \{\tau^{стр}(t) : t \in Tv\}$, $Tv^{столб} = \{\tau^{столб}(t) : t \in Tv\}$.

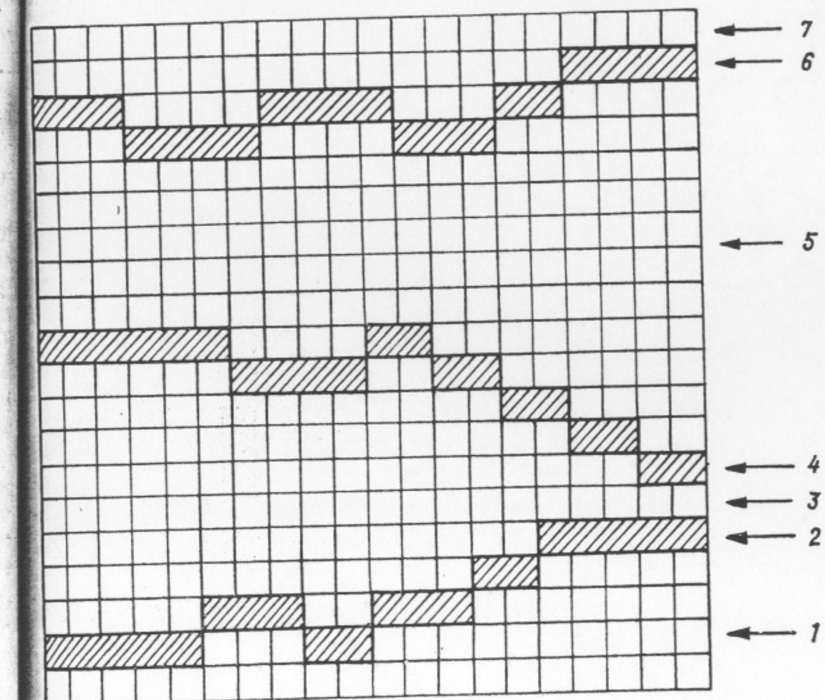


Рис. 2.12. Разбиение поля зрения на части реальными горизонтальными линиями

Обе грамматики имеют алфавит сигналов $\{1, 2, \ldots, 7\}$.

Грамматика $G^{стр}$ сконструирована так, что допустимыми в ней являются изображения, в которых поле зрения разделено на четыре части тремя реальными горизонтальными линиями. Сигналы в клетках, через которые проходят линии, равны 2, 4 и 6, а в остальных клетках — 1, 3, 5, 7, т. е. соответствуют представленным, например, на рис. 2.12. Грамматика $G^{стр}$ строится так, как указано в рассмотренном ранее примере 3.

Грамматика $G^{столб}$ сконструирована так, что допустимыми в ней являются изображения, в которых поле зрения разделено на четыре части тремя реальными вертикальными линиями. Пример изображения, допустимого в грамматике $G^{столб}$, представлен на рис. 2.13.

Пара изображений, допустимых в грамматиках $G^{стр}$ и $G^{столб}$ соответственно, как бы производит сегментацию поля зрения на 49 частей, как показано на рис. 2.14, определяя те

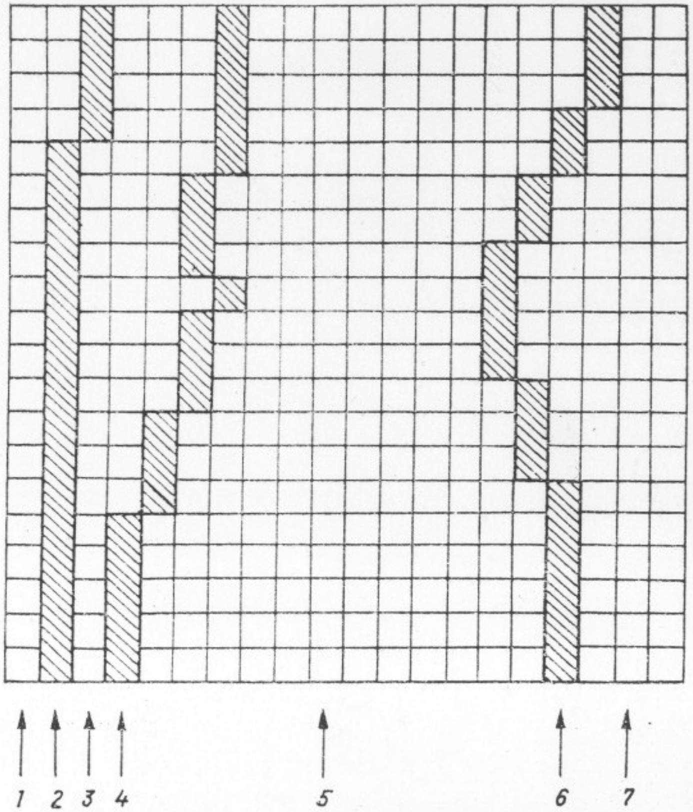


Рис. 2.13. Разбиение поля зрения на части реальными вертикальными линиями

места, где должны будут находиться линии изображения буквы.

Грамматика G, порождающая изображения букв того или иного класса, строится таким образом, что поле зрения в ней есть множество $\{(i, j) : 1 \leqslant i \leqslant n;\ 1 \leqslant j \leqslant m\}$, а поле описания включает в себя множества $Tv^{стр}$ и $Tv^{ст}$, являющиеся полями зрения в грамматиках $G^{стр}$ и $G^{столб}$. Структура грамматики G

включает в себя все тройки вида $\{(i, j), \tau^{стр}(i, j), \tau^{столб}(i, j)\}$, $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m$. Кроме того, поле описания грамматики включает в себя и поля описаний грамматик $G^{стр}$ и $G^{столб}$, равно как структура грамматики G включает в себя и структуры этих грамматик. Предикат допустимости $F(T')$ для троек клеток вида $\{(i, j), \tau^{стр}(i, j), \tau^{столб}(i, j)\}$ должен быть выбран точно так же, как и в рассмотренной в

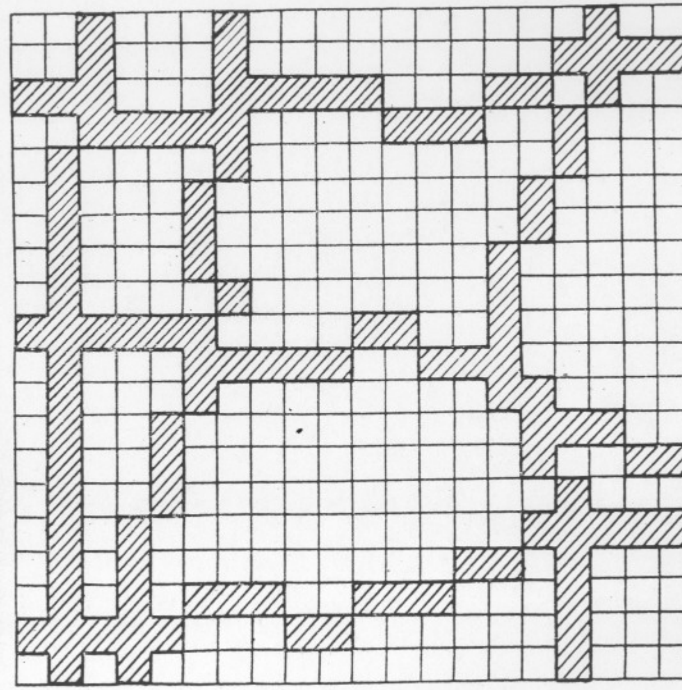


Рис. 2.14. Разбиение поля зрения на части для последующего заполнения их линиями знака

предыдущем примере грамматике, порождающей идеальные изображения.

Изображения, допустимые в построенной указанным способом грамматике, весьма близки к реальным рукописным изображениям. На рис. 2.15 приведен пример допустимого изображения.

7. Мы видим, что конструирование двумерных грамматик, соответствующих тому или иному множеству изображений, является хотя и не сложным, но неформальным процессом, требующим определенной изобретательности и навыков. Формализовать этот процесс полностью невозможно, подобно

тому, как невозможно формализовать процесс написания содержательно правильных программ. В то же время в последних двух рассмотренных примерах достаточно сложных множеств изображений построение грамматики включало в себя использование уже ранее построенных заготовок и выполнялось с помощью определенного повторяющегося приема. Именно

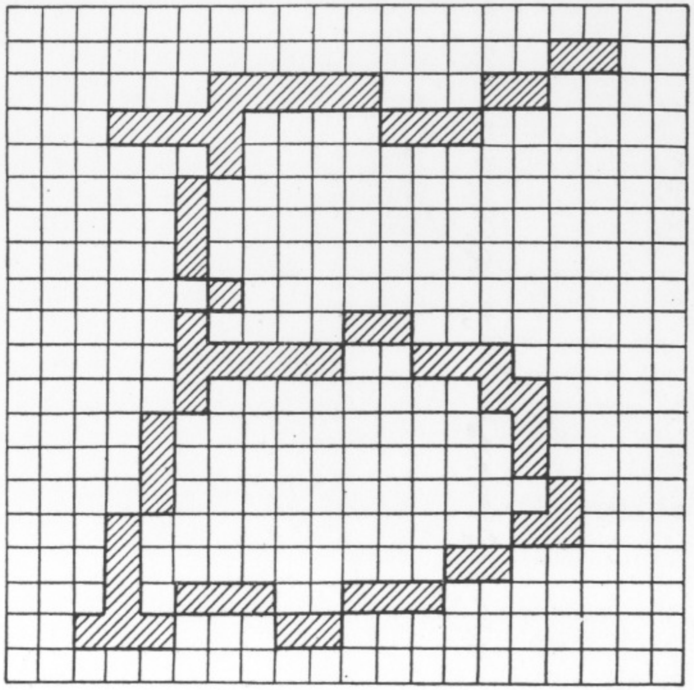


Рис. 2.15. Пример допустимого изображения буквы Б, составленной из реальных горизонтальных и вертикальных линий

этот процесс можно формализовать, т. е. процесс построения грамматики для множества, получаемого в результате известной операции над множествами, для которых двумерная грамматика уже задана. Сформулируем это утверждение более конкретно.

Пусть заданы поле зрения Tv, $n+1$ терминальных алфавитов $V_0, V_1, \ldots, V_n$, не обязательно разных, и $n+1$ изображений $v_i : Tv \to V_i$, $i = 0, 1, 2, \ldots, n$. Для заданного отношения $Z \subset V_0 \times V_1 \times \ldots \times V_n$ будем считать, что изображение v_0 есть Z-компиляция изображений $v_1, v_2, \ldots, v_n$, если для любой клетки $t \in Tv$ справедливо условие $(v_0(t), v_1(t), \ldots, v_n(t)) \in Z$.

Пусть $\mathscr{L}_1, \mathscr{L}_2, \ldots, \mathscr{L}_n$ — множества изображений на поле зрения Tv в алфавитах $V_1, V_2, \ldots, V_n$ соответственно. Множество $\mathscr{L}_0$ изображений $Tv \to V_0$ называется Z-компиляцией множеств $\mathscr{L}_i$, $i = 1, 2, \ldots, n$, если оно содержит все те и только те изображения v_0, для которых существуют такие изображения $v_1 \in \mathscr{L}_1$, $v_2 \in \mathscr{L}_2$, ..., $v_n \in \mathscr{L}_n$, что v_0 есть Z-компиляция изображений $v_1, v_2, \ldots, v_n$.

Примеры Z-компиляций. 1. Множество $\mathscr{L}_0 = \mathscr{L}_1 \cap \mathscr{L}_2$, являющееся пересечением множеств $\mathscr{L}_1$ и $\mathscr{L}_2$, есть Z-компиляция множеств $\mathscr{L}_1$ и $\mathscr{L}_2$ при

$$Z = \{(\sigma_0, \sigma_1, \sigma_2) : \sigma_0 \in V_0,\ \sigma_1 \in V_1,\ \sigma_2 \in V_2,\ \sigma_1 = \sigma_2 = \sigma_3\}.$$

2. Операция типа «изображение v_1 на фоне изображения v_2» может быть представлена как Z-компиляция при

$$Z = \{(\sigma_0, \sigma_1, \sigma_2) : \sigma_0 \in V_0,\ \sigma_1 \in V_1,\ \sigma_2 \in V_2,\ \sigma_0 = \max(\sigma_1, \sigma_2)\}.$$

В этом случае, естественно, предполагается, что алфавиты V_1 и V_2 таковы, что операция max является осмысленной.

3. Интерес представляет такая Z-компиляция множеств $\mathscr{L}_1, \mathscr{L}_2, \ldots, \mathscr{L}_n$, где $\mathscr{L}_n$ — множество изображений в алфавите $\{1, 2, \ldots, n-1\}$, а $Z = \{(\sigma_0, \sigma_1, \sigma_2, \ldots, \sigma_n) : \sigma_0 = \sigma_{\sigma_n}\}$. Использование такой Z-компиляции уместно в случае, когда изображение v_0 составлено из нескольких изображений, каждое из которых занимает свою часть поля зрения. При этом разбиение поля зрения на части, занятые тем или иным изображением, не является произвольным, а определяется изображением v_n (см. рис. 2.16). Частным случаем такой Z-компиляции множеств является объединение $\mathscr{L}_1 \cup \mathscr{L}_2$ множеств. Для этого требуется, чтобы множество $\mathscr{L}_3$ содержало лишь два изображения — v_3' и v_3'', таких, что для любого $t \in Tv$ выполнялись равенства $v_3'(t) = 1$, $v_3''(t) = 2$. Описание примеров закончено.

Пусть заданы n грамматик $G^i = \langle Tv^i, Tk^i, V^i, K^i, \mathscr{T}^i, \{F^i(T'), T' \in \mathscr{T}^i\}\rangle$, $i = 1, 2, \ldots, n$, причем поля зрения Tv^i, $i = 1, 2, \ldots, n$, попарно не пересекаются, равно как и поля описаний Tk^i, $i = 1, 2, \ldots, n$. Кроме того, существует такое поле зрения Tv^0, не пересекающееся ни с одним из Tv^i, $i = 1, 2, \ldots, n$, что существует взаимно-однозначное соответствие между клетками Tv^0 и Tv^i. Клетку поля зрения Tv^i, соответствующую клетке $t \in Tv^0$ поля зрения Tv^0, будем обозначать так: $\tau^i(t) \in Tv^i$. Мы хотим для заданного алфавита V^0 и соотношения $Z \subset V^0 \times V^1 \times \ldots \times V^n$ построить грамматику G^0 такую, что $\mathscr{L}(G^0)$ будет Z-компиляцией множеств $\mathscr{L}(G^1), \mathscr{L}(G^2), \ldots, \mathscr{L}(G^n)$.

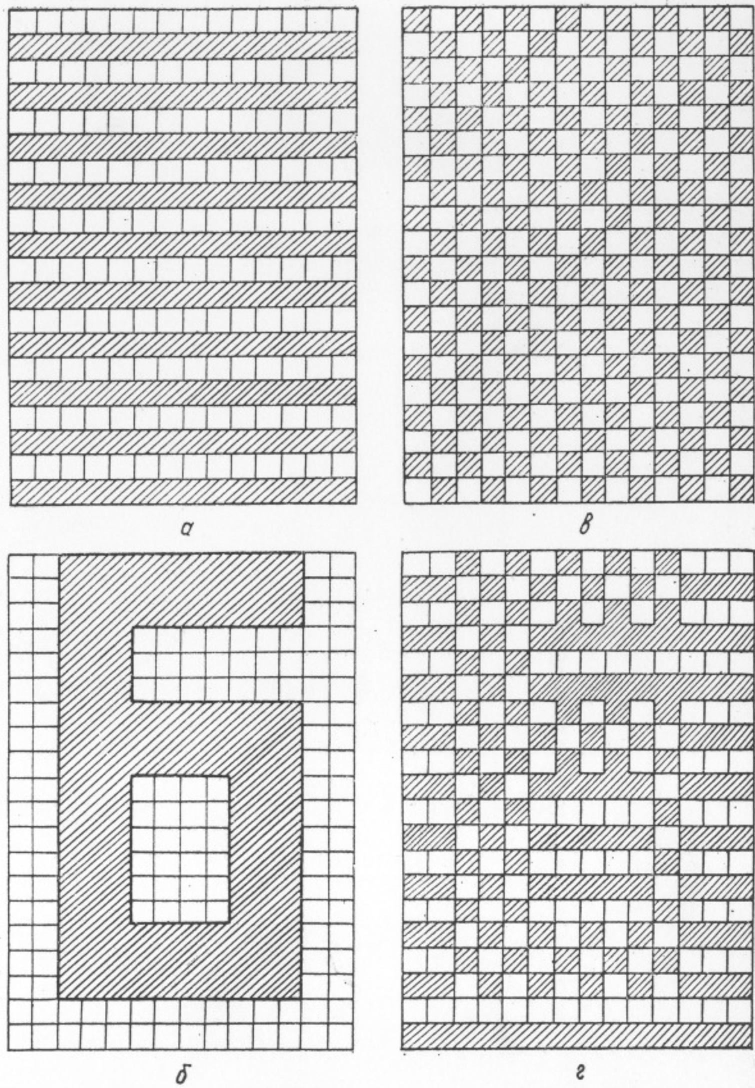


Рис. 2.16. Компиляция изображений *а*, *б*, *в*; *г* — результат компиляции

Поле зрения этой грамматики есть Tv^0, а алфавит сигналов — V^0.

Поле описания этой грамматики есть $\bigcup_{i=1}^{n} (Tv^i \cup Tk^i)$, а алфавит структурных элементов — $\bigcup_{i=1}^{n} (V^i \cup K^i)$.

Структура $\mathcal{T}^0$ грамматики есть $\left(\bigcup_{i=1}^{n} \mathcal{T}^i\right) \cup \mathcal{T}^*$, где $\mathcal{T}^* = \{\{t, \tau^1(t), \tau^2(t), \ldots, \tau^n(t)\} : t \in Tv^0\}$.

Предикаты $F(T')$, $T' \in \mathcal{T}^0$, определяются следующим образом. Для $T' \in \mathcal{T}^i$, $i = 1, 2, \ldots, n$, $F^0(T') = F^i(T')$; для $T' \in \mathcal{T}^*$ $F^0(T')$ должен принимать значение, равное единице, для тех и только тех наборов $(s(t), s(\tau^1(t)), s(\tau^2(t)), \ldots, s(\tau_n(t)))$, которые принадлежат Z.

2.3. ПРЕОБРАЗОВАНИЕ И КОДИРОВАНИЕ ИЗОБРАЖЕНИЙ

Основное назначение двумерных грамматик заключается в задании тех или иных подмножеств изображений. В то же время рассмотрение класса грамматик, в которых допустимыми являются все возможные изображения, также позволяет сформулировать вопросы, представляющие теоретический и практический интерес, и дать на них ответы.

Пусть T и V — два конечных множества, а множество S есть некоторое подмножество в V^T, т. е. подмножество функций вида $T \to V$. Множество S будем называть функциональным относительно некоторого подмножества $T' \subset T$, если для любых $s_1 \in S$, $s_2 \in S$ из неравенства $s_1(T') \neq s_2(T')$ следует $s_1(T \setminus T') \neq s_2(T \setminus T')$.

Двумерная грамматика G с полем зрения Tv, полем описания Tk и алфавитом сигналов V определяет преобразование изображений или является преобразованием изображений, если $\mathcal{L}(G) = V^{Tv}$, т. е. допустимо любое изображение, а множество допустимых вариантов $S(G)$ является функциональным относительно Tk, т. е. любому изображению соответствует единственное описание.

Грамматику G, определяющую преобразование изображений, будем называть к тому же и кодировкой, если множество $S(G)$ является функциональным относительно Tv, т. е. любому описанию соответствует единственное изображение.

Рассмотрим определенный, достаточно распространенный на практике класс преобразований изображений, укажем класс двумерных грамматик, соответствующих этим преобразованиям, и дадим ответы на возникающие при этом вопросы.

Пусть заданы целые положительные числа m, n, I, J, такие, что $I \geqslant m$, $J \geqslant n$. Поле зрения Tv — это множество пар (i, j), $-m + 2 \leqslant i \leqslant I$, $-n + 2 \leqslant j \leqslant J$; V и W — два алфавита, каждый из которых содержит символ $\oplus$, называемый фоном; изображение v с фиксированным краем — это функция $Tv \to V$, такая, что $v(i, j) = \oplus$, если $\min(i, j) \leqslant 0$. Через $Tv'(i, j)$, $1 \leqslant i \leqslant I$, $1 \leqslant j \leqslant J$, обозначен фрагмент поля зрения, содержащий те и только те клетки (i', j'), для которых $i - m + 1 \leqslant i' \leqslant i$, $j - n + 1 \leqslant j' \leqslant j$, а через $v(Tv'(i, j))$ — фрагмент изображения v на фрагменте $Tv'(i, j)$. Через φ обозначена функция от $m \times n$ переменных, которая каждому фрагменту $v(Tv'(i, j))$ ставит в соответствие элемент из алфавита W. При заданной функции φ клетка (i, j) называется особым местом изображения $v : Tv \to V$, если $\varphi(v(Tv'(i, j))) \neq \oplus$. Это особое место имеет тип $K = \varphi(v(Tv'(i, j))) \in W \setminus \{\oplus\}$. Списком особых мест изображения v называется множество троек

$$\{(i, j, k) : 1 \leqslant i \leqslant I, \quad 1 \leqslant j \leqslant J, \quad \varphi(v(Tv'(i, j))) = k, \quad k \neq \oplus\}.$$

Представление изображения списком особых мест — распространенный прием в распознавании изображений, так как для многих классов изображений такое представление является мощным средством сжатия информации. До настоящего времени выбор функции φ основывался полностью на интуиции исследователя.

Для правильного выбора функции φ необходимо уметь отвечать на два вопроса. Во-первых, какой должна быть функция φ, чтобы список особых мест давал однозначное представление об изображении. Иными словами, следует описать множество допустимых функций φ, определяющих обратимое отображение множества изображений с фиксированным краем в множестве списков особых мест. Во-вторых, для данного изображения следует указать такую функцию φ^* из множества допустимых, чтобы список особых мест на этом изображении оказался как можно короче.

Для любой функции φ определим грамматику G_φ, построенную следующим образом. Алфавит сигналов равен V, а алфавит структурных элементов — W. Поле зрения Tv этой грамматики определим как множество

$$\{(i, j) : -m + 2 \leqslant i \leqslant I, \ -n + 2 \leqslant j \leqslant J\}.$$

Поле описания Tk определим как множество, изоморфное множеству $T = \{(i, j) : 1 \leqslant i \leqslant I, 1 \leqslant j \leqslant J\}$. Через $\tau(i, j)$ обозначим клетку поля описания, соответствующую паре чисел $(i, j) \in Tv$. Таким образом, $Tk = \{\tau(t) : t \in T\}$.

Структуру $\mathscr{T}$ грамматики определим как объединение подмножеств $\mathscr{T}_1$ и $\mathscr{T}_2$, первое из которых $\mathscr{T}_1 = \{\{(i, j)\} : \min(i, j) \leqslant 0\}$, а второе — $\mathscr{T}_2 = \{Tv'(i, j) \cup \{\tau(i, j)\} : \min(i, j) > 0\}$.

Предикаты допустимости $F(T')$ для $T' \in \mathscr{T}_1$ выберем таким образом, чтобы в клетках (i, j), таких, что $\min(i, j) \leqslant 0$, был допустимым лишь сигнал $\oplus$. Для $T' \in \mathscr{T}_2$, т. е. для $T' = Tv'(i, j) \cup \{\tau(i, j)\}$, предикат допустимости $F(T')$ должен быть равен единице для тех и только тех фрагментов

$s(Tv'(i, j) \cup \{\tau(i, j)\})$, для которых $s(\tau(i, j)) = \varphi(s(Tv'(i, j)))$.

Очевидно, что введенная таким способом грамматика G_φ при любой функции φ определяет преобразование изображений с фиксированным краем. Первый из поставленных выше вопросов, таким образом, эквивалентен вопросу, какой должна быть функция φ, чтобы грамматика G_φ определяла кодировку. Ответ на этот вопрос дает следующая теорема.

Теорема 2. *Для того чтобы грамматика G_φ определяла кодировку, необходимо и достаточно, чтобы для любых двух фрагментов изображений $v_1(Tv'(i, j))$ и $v_2(Tv'(i, j))$, таких, что $v_1(Tv'(i, j) \setminus \{(i, j)\}) = v_2(Tv'(i, j) \setminus \{(i, j)\})$, а $v_1(i, j) \neq v_2(i, j)$, выполнялось неравенство $\varphi(v_1(Tv'(i, j))) \neq \varphi(v_2(Tv'(i, j)))$.*

Доказательство. 1. Допустим, что функция φ не удовлетворяет условиям, указанным в теореме, и докажем, что в этом случае грамматика G_φ допускает по крайней мере два варианта s_1^* и s_2^*, таких, что $s_1^*(Tk) = s_2^*(Tk)$, а $s_1^*(Tv) \neq s_2^*(Tv)$. Найдем два таких фрагмента $v_1(Tv'(m, n))$ и $v_2(Tv'(m, n))$, что $v_1(m, n) \neq v_2(m, n)$, $v_1(Tv'(m, n) \setminus \{(m, n)\}) = v_2(Tv'(m, n) \setminus \{(m, n)\})$, а

$$\varphi(v_1(Tv'(m, n))) = \varphi(v_2(Tv'(m, n))). \qquad (2.12)$$

Такая пара фрагментов существует в силу сделанного предположения, что функция φ не удовлетворяет требованиям, указанным в теореме. Построим два таких изображения v_1^* и v_2^*, для которых при всех $t \in Tv$, $t \neq (I, J)$ выполняется равенство $v_1^*(t) = v_2^*(t)$, кроме того, $v_1^*(Tv'(I, J)) = v_1(Tv'(m, n))$ и $v_2^*(Tv'(I, J)) = v_2(Tv'(m, n))$. Это значит, что для всех $(i, j) \neq (I, J)$ выполняется равенство $v_1^*(Tv'(i, j)) = v_2^*(Tv'(i, j))$, а следовательно, и равенства

$$\varphi(v_1^*(Tv'(i, j))) = \varphi(v_2^*(Tv'(i, j))), \quad (i, j) \neq (I, J). \quad (2.13)$$

Справедливость равенств (2.12) и (2.13) означает доказательство того, что если функция φ не удовлетворяет условиям, указанным в теореме, то существуют два различных изображения v_1^* и v_2^*, которым соответствуют одинаковые описания.

2. Допустим, что существуют разные изображения v_1^* и v_2^*, которым соответствуют одинаковые описания, и докажем на этом основании, что функция φ не удовлетворяет требованиям, указанным в теореме. Поскольку $v_1^* \neq v_2^*$, множество $D = \{(i, j) : v_1^*(i, j) \neq v_2^*(i, j)\}$ не является пустым. В этом множестве имеется такая клетка (i^*, j^*), что

$$v_1^*(Tv'(i^*, j^*) \setminus \{(i^*, j^*)\}) = v_2^*(Tv'(i^*, j^*) \setminus \{(i^*, j^*)\}). \quad (2.14)$$

В частности, такую клетку можно найти следующим образом. В множестве D выделим подмножество клеток, имеющих наименьший номер i, а в выделенном подмножестве найдем клетку, имеющую наименьший номер j. Найденная таким способом клетка (i^*, j^*) такова, что $i^* > 0$ и $j^* > 0$, поскольку в противном случае выполняется равенство $v_1^*(i, j) = v_2^*(i, j) = \oplus$, которое невозможно для клеток множества D. Кроме того, найденная пара чисел (i^*, j^*) такова, что для всех клеток (i', j), для которых выполняются неравенства $(i', j') = (i^*, j^*)$, $i' \leqslant i^*$, $j' \leqslant j^*$, выполняется равенство $v_1^*(i', j') = v_2^*(i', j')$. Следовательно, $v_1^*(Tv(i^*, j^*) \setminus \{i^*, j^*\}) = v_2^*(Tv(i^*, j^*) \setminus \{i^*, j^*\})$, $v_1^*(i^*, j^*) \neq v_2^*(i^*, j^*)$, однако $\varphi(v_1^*(Tv'(i^*, j^*))) = \varphi(v_2^*(Tv'(i^*, j^*)))$.

Это значит, что функция φ не удовлетворяет условиям теоремы. Теорема доказана.

Доказанная теорема означает, что допустимой является любая функция φ, принимающая различные значения на фрагментах изображений, отличающихся только сигналом в правой верхней клетке фрагмента. Из этого свойства допустимых функций φ почти непосредственно следует ответ на второй из поставленных ранее вопросов, а именно какова оптимальная в определенном смысле функция φ. Ответ на этот вопрос будет дан позже, а здесь укажем еще на одно свойство кодировок, которое также представляет практический интерес. Напомним, что V и W_φ — это алфавиты сигналов и структурных элементов грамматики G_φ, которая однозначно задана функцией φ, как определено выше.

Теорема 3. *Для любой кодировки G_φ существует кодировка $G_{\varphi'}$, такая, что:*

1) количество структурных элементов в алфавите $W_{\varphi'}$ равно количеству сигналов в алфавите V;

2) для любых двух вариантов $s_\varphi \in S(G_\varphi)$ и $s_{\varphi'} \in S(G_{\varphi'})$, таких, что $s_\varphi(Tv) = s_{\varphi'}(Tv)$, выполняется соотношение

$$\{t : t \in Tk,\ s_{\varphi'}(t) = \oplus\} \supset \{t : t \in Tk,\ s_\varphi(t) = \oplus\}.$$

Доказательство. Покажем, каким образом следует выбирать функцию φ', если известна функция φ. Рассмотрим некоторый фрагмент $v^*(Tv'(i, j) \setminus \{(i, j)\})$, обозначаемый далее для краткости через v^*. Через $\bar{v}(v^*)$ обозначим множество расширений фрагмента v^* на множество $Tv'(i, j)$. Очевидно, что $|\bar{v}(v^*)| = |V|$. Выберем алфавит $W_{\varphi'}$ произвольным образом, но так, чтобы $\oplus \in W_{\varphi'}$ и $|W_{\varphi'}| = |V|$. Определим, какие значения должна принимать функция φ' на фрагментах из множества $\bar{v}(v^*)$. Если для некоторого фрагмента $v' \in \bar{v}(v^*)$ справедливо равенство $\varphi(v') = \oplus$ (а это может выполняться не более чем для одного фрагмента из $\bar{v}(v^*)$), то функция φ' на этом фрагменте также должна принимать значение $\oplus$. На всех прочих фрагментах из $\bar{v}(v^*)$ функция φ' может принимать произвольные, но разные значения, не равные $\oplus$. Если же для любого фрагмента $v' = \bar{v}(v^*)$ выполняется неравенство $\varphi(v') \neq \oplus$, то функция φ' на множестве $\bar{v}(v^*)$ может принимать произвольные, но различные для разных фрагментов значения.

Таким же образом определяются значения, которые должна принимать функция φ' на множествах $\bar{v}(v^*)$, соответствующих всем остальным v^*. По способу построения функции φ' видно, что она удовлетворяет требованиям предыдущей теоремы, а значит, грамматика $G_{\varphi'}$ является кодировкой. В то же время функция φ' выбрана таким образом, что если для некоторого фрагмента $v(T'(i, j))$ выполняется равенство $\varphi(v(T'(i, j))) = \oplus$, то для этого же фрагмента функция φ' тоже равна $\oplus$. Поэтому для любого изображения v выполняется соотношение

$$\{(i, j) : \varphi'(v(Tv'(i, j))) = \oplus\} \supset \{(i, j) : \varphi(v(Tv'(i, j))) = \oplus\}.$$

Теорема доказана.

Укажем некоторые следствия из только что доказанной теоремы, которые представляют практический интерес.

Пусть алфавит сигналов V есть $\{1, 2, \ldots, |V|\}$ — множество целых чисел от 1 до $|V|$, которые означают, например, черноту клетки. Распространенным приемом кодирования таких изображений есть составление списка тех координат (i, j), для которых $v(i, j) \neq v(i, j-1)$, с указанием величины $v(i, j) - v(i, j-1)$. Такой прием оправдан, если есть основания предполагать, что в большей части изображения эта разность равна нулю, т. е. что изображение является кусочно-постоянным с небольшим числом кусков. Алфавит W в этом случае состоит из $2|V| - 1$ чисел, что приводит к неудобствам, так как для запоминания сигналов из W и V требуется память различных форматов. Так, если $|V| = 256$, то для хранения информации о черноте требуется один байт, которого уже недостаточно для хранения разности чернот.

В силу доказанной теоремы функция $(v(i, j) - v(i, j-1))$ может быть заменена функцией $d(v(i, j), v(i, j-1))$, которая не хуже, чем разность, но принимает лишь $|V|$ значений. В частности, таковой может быть функция

$$d(v(i, j), v(i, j-1)) = v(i, j) - v(i, j-1),$$
$$\text{если } v(i, j) - v(i, j-1) \geqslant 0,$$
$$d(v(i, j), v(i, j-1)) = |V| + v(i, j) - v(i, j-1),$$
$$\text{если } v(i, j) - v(i, j-1) < 0.$$

Если есть основание считать, что изображение — кусочно-линейное, то обычный прием состоит в вычислении функции $v(i, j) - 2v(i, j-1) + v(i, j-2)$ или функции $(v(i, j) - v(i, j-1)) - (v(i-1, j) - v(i-1, j-1))$ и запоминании координат тех клеток, для которых эти функции принимают ненулевые значения, с указанием этих ненулевых значений. Алфавит W в этом случае уже имеет объем $4 \mid V \mid - 3$. Из доказанной теоремы следует, что любую из двух рассмотренных функций также можно заменить другой, не худшей функцией, но принимающей лишь $\mid V \mid$ значений.

При кодировании графических изображений, т. е. изображений с алфавитом $V = \{0, 1\}$, часто используют представление изображения в виде списка особых мест, содержательный смысл которых выражается словами «уголок», «разветвление» и т. п. При этом молчаливо допускается, что наряду с координатой особого места необходимо запомнить его тип («уголок», «разветвление» и т. п.), а также указать число, которое обозначает ориентацию найденной конфигурации. Однако на основании доказанной теоремы функцию φ, которая для каждого найденного особого места определяет еще и его тип, можно заменить функцией φ', принимающей только два значения, так как $\mid V \mid = 2$. Количество типов особых мест становится равным единице, и запоминать его, следовательно, нет необходимости. Таким образом, для случая графических изображений получаем следующую практическую рекомендацию.

Если существует некоторый способ представления графических изображений с помощью списка, содержащего координаты особых мест на изображении с указанием их характеристик, и это представление допускает однозначное восстановление исходного изображения, что существует другой способ, отличающийся от первого тем, что некоторые особые места не вносятся в список вообще, а для оставшихся запоминаются только координаты без указания каких-либо дополнительных характеристик. При этом новый способ также допускает однозначное восстановление всего исходного изображения в целом, к тому же для любого изображения список особых мест, получаемых по второму способу, оказывается не длиннее, чем список особых мест, получаемых по первому способу. Этот второй способ должен строиться так, как указано в доказательстве теоремы.

В коллективе, где выполнялась данная работа, для кодирования графических изображений длительное время применялась кодировка, основанная на анализе четверок клеток вида (i, j), $(i, j-1)$, $(i-1, j)$, $(i-1, j-1)$. Существует всего 16 фрагментов изображений на такой четверке клеток. Эти фрагменты представлены на рис. 2.17. Исходя из характера решаемых прикладных задач анализа чертежно-графических изображений было принято, что фрагменты, представленные на рисунке под номерами *1—6*, не являются особыми, а все остальные — особые. Такая классификация фрагментов представлялась разумной, потому что фрагменты *1—6* имеются на изображении чертежа в больших количествах, а остальные встречаются редко. Казалось также понятным (и в

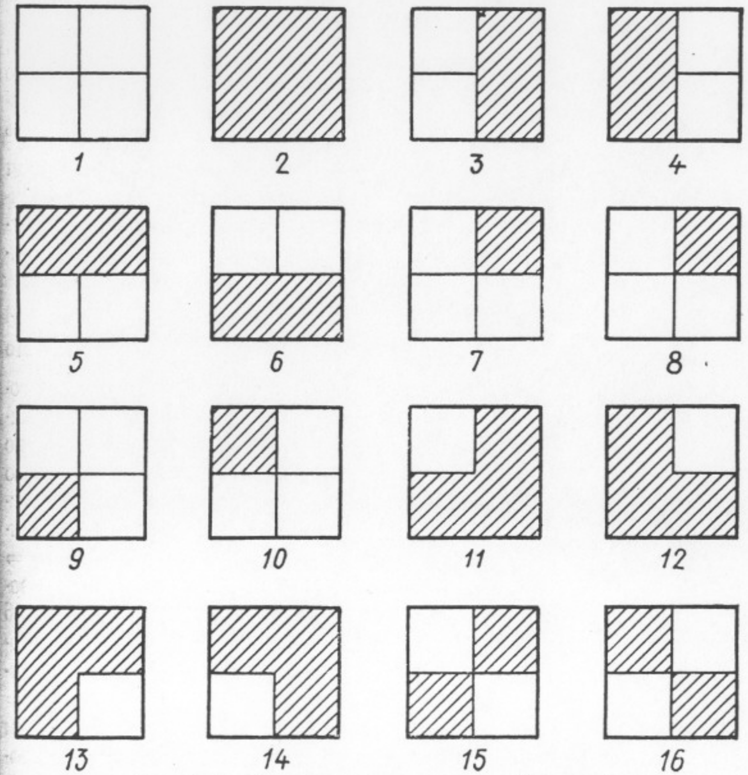


Рис. 2.17. Фрагменты двоичных изображений

действительности верным), что списка одних лишь координат особых мест недостаточно для восстановления изображения. Поэтому наряду с координатой особого места запоминался его тип, принимающий в данном случае одно из десяти возможных значений. При доказательстве теоремы 3 показано, каким образом следует изменить классификацию фрагментов на особые и не особые, чтобы запоминание типа особых фрагментов стало ненужным. Для этого необходимо один и только один из фрагментов 9 и 15 объявить неособым так же, как и один и только один из фрагментов 14 и 16.

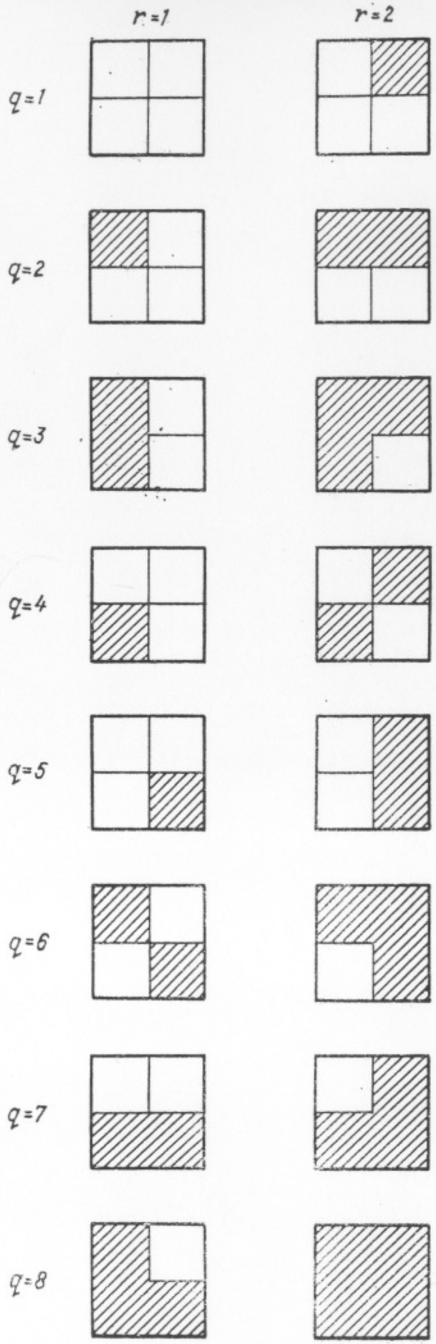


Рис. 2.18. К алгоритму поиска оптимальной кодировки

При заданных алфавитах V и W и числах m и n существует $|V|^{m\cdot n}$ различных фрагментов $v(Tv'(i, j))$ и $|W|^{|V|^{m\cdot n}}$ различных функций φ и, следовательно, столько же грамматик G_φ. Согласно теореме 2 лишь $|W|^{|V|^{m\cdot n-1}}$ из них могут быть кодировками. В силу теоремы 3 можно ограничиться рассмотрением лишь $|V|^{|V|^{m\cdot n-1}}$ кодировок. Для двоичных изображений это число равно $2^{2^{m\cdot n-1}}$, которое даже при очень малых $m = n = 2$ заметно больше, чем количество практически применяемых кодировок. А именно: даже при $m = n = 2$ для двоичных изображений существует 256 кодировок. На этом простейшем примере, именно для $m = n = |V| = 2$, покажем, каким образом для данного изображения v найти ту кодировку, для которой список особых мест оказывается минимальной длины. Результатом такой автоматически настраиваемой кодировки является массив, содержащий список особых мест в изображении, и сопровождающая информация в 1 байт, указывающая, какая из 256 кодировок была применена.

Фрагмент $Tv'(i, j)$ в указанном простейшем случае — это четверка клеток (i, j), $(i-1, j)$, $(i, j-1)$, $(i-1, j-1)$. Множество фрагментов $v(Tv'(i, j))$ представлено на рис. 2.18. Эти фрагменты расположены в восьми строках, каждая из которых содержит два фрагмента, отличающиеся лишь сигналом в клетке (i, j). Фрагмент, расположенный в q-й строке и r-м столбце, обозначен как $p(q, r)$. Построить грамматику G_φ, т. е. построить функцию φ,— значит указать подмножество фрагментов из множества, представленного на рис. 2.18, точнее, указать фрагменты, которые будут считаться особыми фрагментами. Для того чтобы функция G_φ являлась кодировкой, необходимо и достаточно, чтобы в указанное подмножество входил один и только один фрагмент из каждой строки. Задача построения оптимального кодирования заключается в подборе для заданного изображения v такого подмножества фрагментов, чтобы количество особых фрагментов на заданном изображении было минимальным.

Эта задача решается алгоритмом, в котором формируются целые числа $N(1)$, $N(2)$, ..., $N(8)$, принимающие значения от 0 до $[\|Tv\|]_2$, где через $[\|Tv\|]_2$ обозначено минимальное четное число, не меньшее, чем $|Tv|$. Алгоритм работает по шагам, которые пронумерованы парой чисел (i, j), $i = 1, 2, \dots, I$; $j = 1, 2, \dots, J$. Первоначально все числа $N(q)$, $q = 1, 2, \dots, 8$, равны $[\|Tv\|]_2/2$. На шаге с номером (i, j) выполняются следующие действия. Определяется фрагмент $v(Tv'(i, j))$ изображения и находится фрагмент $p(q, r)$ из множества представленных на рис. 2.18, такой, что $p(q, r) = v(Tv'(i, j))$. После этого число $N(q)$ увеличивается на 1, если $N(q) \neq [\|Tv\|]_2$, а $r = 1$, и уменьшается на 1, если $N(q) \neq 0$, а $r = 2$. Во всех прочих случаях число $N(q)$ не изменяется. Все прочие числа $N(q')$, $q' \neq q$, на этом шаге не изменяются. После выполнения всех шагов функция φ определяется следующим образом. Для $q = 1, 2, \dots, 8$, если $N(q) \geqslant [\|Tv\|]_2/2$, функция φ на фрагменте $p(q, 1)$ принимает значение, равное $\oplus$, а на фрагменте $p(q, 2)$ — не равное $\oplus$. Если $N(q) < [\|Tv\|]_2/2$, то функция φ на фрагменте $p(q, 1)$ принимает значение, не равное $\oplus$, а на фрагменте $p(q, 2)$ — равное $\oplus$.

На рис. 2.19 приведены четыре изображения и соответствующие им описания, представляющие кодировку по описанному способу. Клетки (i, j), для которых $W(i, j) = \oplus$, на рисунке обозначены как □, а для которых $W(i, j) \neq \oplus$ — как

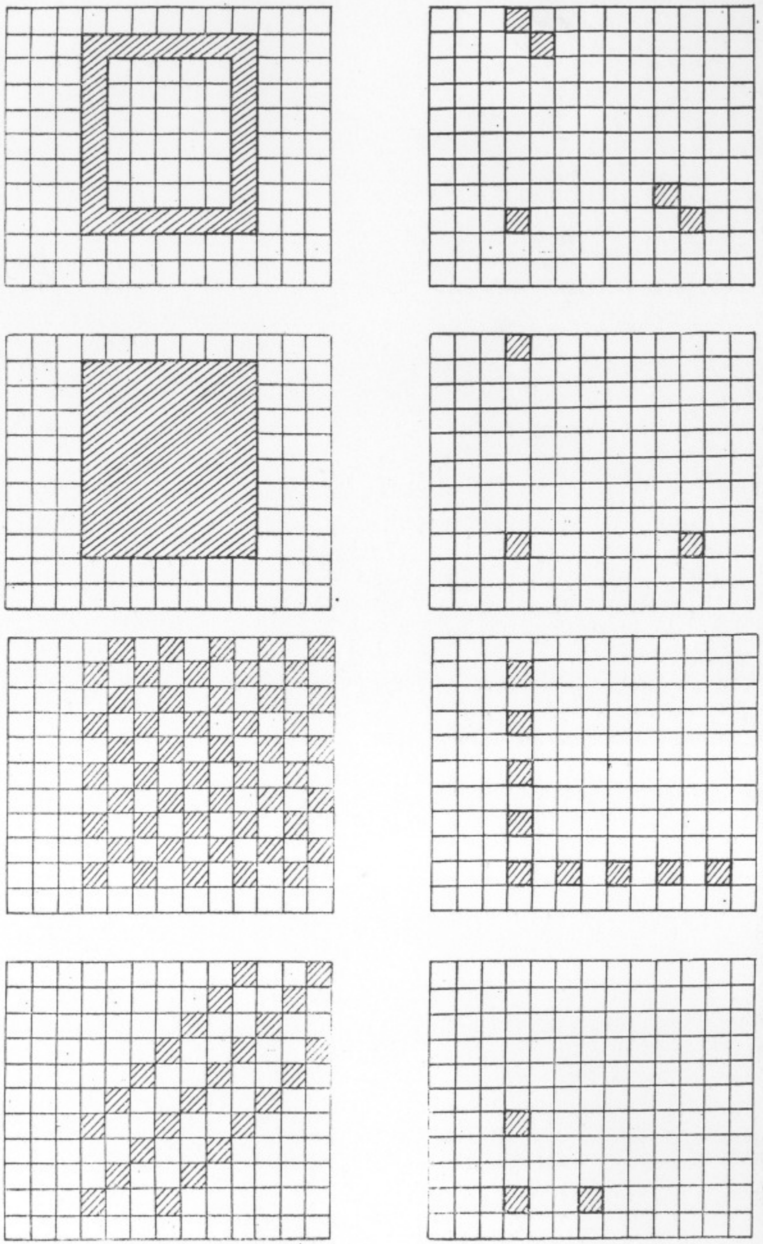


Рис. 2.19. Справа — изображения, слева — соответствующие им описания в оптимальной кодировке

2.4. ИЗОБРАЖЕНИЯ В НЕРАСТРОВОМ ПРЕДСТАВЛЕНИИ

Данная глава была бы неполной без следующего заключительного замечания. Во всех рассмотренных выше примерах областью определения изображения, т. е. полем зрения, служит множество пар $\{(i, j) : 1 \leqslant i \leqslant m;\ 1 \leqslant j \leqslant n\}$. Это множество, т. е. прямоугольный участок двумерной целочисленной решетки, обычно в технике обработки изображений называется дискретным растром, а представление изображения как функции, заданной на растре,— растровым представлением. Мы уделили так много внимания изображениям в растровом представлении, чтобы подчеркнуть, что двумерные грамматики применимы к анализу данных об изображениях, поступающих непосредственно с технического устройства, которое осуществляет сканирование изображения на физическом носителе. Применение двумерных грамматик для обработки изображений не требует специального преобразования изображения к виду, пригодному для их обработки, в отличие от целого ряда известных работ [24, 101], в которых предполагаются методы анализа не собственно изображения, а некоторого предложения, описывающего изображение в виде списка частей изображения и отношений между ними. Получение такого описания изображения не является тривиальной задачей, поэтому применимость этих методов носит условный характер. Применимость двумерных грамматик для задания множеств изображений в их растровом представлении позволяет говорить о безусловной применимости двумерных грамматик, так как возможность растрового представления изображения не вызывает сомнений.

В то же время растровое представление не является единственно возможным, а в ряде случаев — и не самым уместным представлением изображения. Описание изображения в виде контуров, т. е. границ между черными и белыми клетками, в виде списка особых мест и т. п. зачастую не только более экономно, но и более уместно с точки зрения дальнейшей обработки. К таким изображениям также применим аппарат двумерных грамматик, так как в этом аппарате нет каких-либо ограничений на структуру того графа, который служит областью определения изображения. Конечно, применение той или иной структуры должно иметь инструментальную поддержку, т. е. должен существовать аппаратный или программный сервис, обеспечивающий уместное представление изображения. Такие аппаратные и программные средства описаны в приложении.

Глава 3

ДВА АЛГОРИТМА ОБРАБОТКИ И РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ

В предыдущей главе показано, что разнообразные содержательно осмысленные множества изображений могут быть представлены определенным единообразным способом, а именно средствами двумерных грамматик. В данной главе мы покажем, что и задачи распознавания, первоначально представляющиеся различными, допускают единую формулировку и решение с помощью единого алгоритма.

Из множества задач распознавания прежде всего выделим две основные задачи, называемые распознаванием идеальных изображений и распознаванием реальных изображений.

Распознавание идеальных изображений заключается в том, чтобы для заданной двумерной грамматики G и заданного изображения v ответить на вопрос о справедливости отношения $v \in \mathcal{L}(G)$. В параграфе 3.1 формулируются и доказываются условия, достаточные для того, чтобы $v \in \mathcal{L}(G)$, и условия, достаточные для того, чтобы $v \notin \mathcal{L}(G)$. Эти условия легко распознаваемы, т. е. сформулирован алгоритм, проверяющий их безошибочно в том смысле, что исключена ситуация, когда алгоритм принимает решение, при котором $v \in \mathcal{L}(G)$, а в действительности $v \notin \mathcal{L}(G)$, равно как исключается ситуация, когда алгоритм принимает решение, при котором $v \notin \mathcal{L}(G)$, а в действительности $v \in \mathcal{L}(G)$. В то же время в описанном алгоритме допускается такое завершение работы, которое следует интерпретировать, как отказ от распознавания, что, естественно, не препятствует его практическому применению.

Распознавание реальных изображений заключается в том, чтобы при заданной грамматике G, порождающей изображения на поле зрения Tv в алфавите V, заданном изображении $v: Tv \to V$, не обязательно принадлежащем $\mathcal{L}(G)$, и заданной функции $f: V \times V \to R$ найти изображение $v^* \in \mathcal{L}(G)$, максимизирующее функционал $\sum_{t \in Tv} f(v(t),\ v^*(t))$.

В параграфе 3.2 формулируется достаточное условие оптимальности изображения $v^* \in \mathcal{L}(G)$ и описывается алгоритм поиска изображения, для которого выполняются эти достаточные условия оптимальности, если такое изображение вообще существует. Алгоритм работает безошибочно в том смысле, что исключена ситуация, когда алгоритм указывает неоптимальное изображение. Это значит, что исключена ситуация, характерная для алгоритмов оптимизации, отыскивающих разного рода локально-оптимальные решения, т. е. обеспечивающих лишь необходимые условия оптимальности.

В параграфе 3.3 рассматривается конструирование безотказных алгоритмов распознавания идеальных изображений и зависимость сложности этих алгоритмов от структуры грамматики. С этой целью вводится параметр структуры, называемый степенью структуры и упорядочивающий структуры по сложности. Показывается, что степень структуры является распознаваемым параметром, т. е. формулируется алгоритм, определяющий для любой структуры ее степень. Алгоритм не очень сложен, хотя доказательство его правильности не представляется простым.

Алгоритм распознавания реальных изображений, не допускающий ситуации «отказ», оказывается очень близким к алгоритму распознавания идеальных изображений. Формализация этих общих свойств выполнена в параграфе 3.4, где сформулирована задача, названная общей двумерно-грамматической задачей. Она заключается в вычислении определенных характеристик языков, порождаемых двумерной грамматикой. Частными случаями двумерно-грамматической задачи являются как распознавание идеальных и реальных изображений, так и другие практические задачи распознавания.

При описании указанных результатов, к которому мы приступаем, помимо введенного ранее обозначения грамматики $G = \langle Tv, Tk, V, K, \mathcal{T}, \{F(T'),\ T' \in \mathcal{T}\}\rangle$ будем использовать обозначение $G = \langle Tv, Tk, V, K, \mathcal{T}, \{Z(T'),\ T' \in \mathcal{T}\}\rangle$, где $Z(T')$ есть множество тех и только тех фрагментов $s(T')$, для которых $F(T', s(T')) = 1$. Кроме того, в дальнейшем в монографии будем рассматривать лишь двумерные грамматики второго порядка.

3.1. РАСПОЗНАВАНИЕ ИДЕАЛЬНЫХ ИЗОБРАЖЕНИЙ

Алгоритм вычеркивания. Пусть $G = \langle Tv, Tk, V, K, \mathcal{T}, \{Z(T'),\ T' \in \mathcal{T}\}\rangle$ — двумерная грамматика второго порядка. Будем полагать, что $\mathcal{T} \supset \{\{t\} : t \in Tv \cup Tk\}$, $T = Tv \cup Tk$, $S = V \cup K$. Следовательно, $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$, где $\mathcal{T}_1 = \{\{t\} : t \in Tv \cup Tk\}$, $\mathcal{T}_2 \subset (Tv \cup Tk) \times (Tv \cup Tk)$.

Определение 1. Для любого $T' \in \mathcal{T}_2$ и $t \in T'$ проекцией $P(Z(T'), t)$ множества $Z(T')$ на клетку t будем называть множество $\{\beta : \beta \in S,\ \beta = s(T', t),\ s(T') \in Z(T')\}$. Это значит, что символ $\beta \in S$ будем считать входящим в проекцию $P(Z(T'), t)$, если существует фрагмент $s(T'): T' \to S$, входящий в $Z(T')$ и принимающий на клетке t значение β.

Определение 2. Грамматику G будем называть согласованной, если для любой пары T' и t, $T' \in \mathscr{T}_2$, $t \in T'$, справедливо равенство

$$Z(\{t\}) = P(Z(T'), t). \tag{3.1}$$

Определение 3. Грамматику $G_1 = \langle Tv, Tk, V, K, \mathscr{T}, \{Z_1(T'), T' \in \mathscr{T}\}\rangle$ будем называть подграмматикой в $G_2 = \langle Tv, Tk, V, K, \mathscr{T}, \{Z_2(T'), T' \in \mathscr{T}\}\rangle$, $G_1 \subset G_2$, если $Z_1(T') \subset Z_2(T')$ для любого $T' \in \mathscr{T}$. Очевидно, что если $G_1 \subset G_2$, то $L(G_1) \subset L(G_2)$ и $S(G_1) \subset S(G_2)$.

Определение 4. Грамматику $C(G)$ будем называть ядром G, если $C(G)$ — согласованная грамматика, $C(G) \subset G$ и не существует такой согласованной грамматики G', что $G' \neq C(G)$, и в то же время $C(G) \subset G' \subset G$.

Теорема 1. *Любая грамматика $G = \langle Tv, Tk, V, K, \mathscr{T}, \{Z(T'), T' \in \mathscr{T}\}\rangle$ имеет не более одного ядра.*

Доказательство. Для любых двух множеств $Z_1(T)$ и $Z_2(T)$ и клетки $t \in T$ справедливо равенство

$$P(Z_1(T) \cup Z_2(T), t) = P(Z_1(T), t) \cup P(Z_2(T), t). \tag{3.2}$$

Допустим, что существуют грамматики G_1 и G_2, каждая из которых является ядром G, и, кроме того, $G_1 \neq G_2$, т. е.

$$\exists T' \in \mathscr{T} (Z_1(T') \neq Z_2(T')). \tag{3.3}$$

Здесь и далее через $Z_1(T')$ и $Z_2(T')$ обозначены множества Z, относящиеся соответственно к G_1 и G_2. Грамматика G_1 есть ядро G, и поэтому G_1 — согласованная подграмматика в G. На основании определения 2

$$\forall T' \in \mathscr{T} (\forall t \in T' (Z_1(\{t\}) = P(Z_1(T'), t))), \tag{3.4}$$

$$\forall T' \in \mathscr{T} (Z_1(T') \subset Z(T')). \tag{3.5}$$

Поскольку G_2 — также ядро G, для G_2 справедливы аналогичные утверждения:

$$\forall T' \in \mathscr{T} (\forall t \in T' (Z_2(\{t\}) = P(Z_2(T'), t))), \tag{3.6}$$

$$\forall T' \in \mathscr{T} (Z_2(T') \in Z(T')). \tag{3.7}$$

Рассмотрим грамматику $G_3 = \langle Tv, Tk, V, K, \mathscr{T}, \{Z_3(T'), T' \in \mathscr{T}\}\rangle$, где

$$\forall T' \in \mathscr{T} (Z_3(T') = Z_1(T') \cup Z_2(T')). \tag{3.8}$$

Из (3.5), (3.7), (3.8) следует, что $\forall T' \in \mathscr{T} (Z_3(T') \subset Z(T'))$, т. е. $G_3 \subset G$. Из (3.2), (3.4), (3.6), (3.8) следует, что $\forall T' \in \mathscr{T} (\forall t \in T' (Z_3(t) = P(Z_3(T'), t)))$, т. е. грамматика G_3 также является согласованной.

Из (3.3) и (3.8) следует, что либо $\exists T' \in \mathscr{T} (Z_2(T') \neq Z_3(T'))$, либо $\exists T' \in \mathscr{T} (Z_1(T') \neq Z_3(T))$, т. е. либо $G_1 \neq G_3$, либо $G_2 \neq G_3$. Кроме того, из (3.8) следует, что $G_1 \subset G_3$, $G_2 \subset G_3$.

Таким образом, грамматика G_3 есть согласованная подграмматика в G, содержащая как грамматику G_1, так и грамматику G_2, причем по крайней мере одна из них не равна G_3.

Следовательно, по крайней мере одна из грамматик G_1 или G_2 не является ядром G. Теорема доказана.

Пусть $G = \langle Tv, Tk, V, K, \mathscr{T}, \{Z(T'), T' \in \mathscr{T}\}\rangle$ — двумерная грамматика, такая, что $\mathscr{T} = \mathscr{T}_1 \cup \mathscr{T}_2$, $\mathscr{T}_1 = \{\{t\} : t \in Tv \cup Tk\}$, $\mathscr{T}_2 \subset (Tv \cup Tk) \times (Tv \cup Tk)$. Обозначим через $\mathscr{T}_2(t)$ множество $\{T' : T' \in \mathscr{T}_2, t \in T'\}$.

Вычеркиванием вершин в грамматике G назовем получение грамматики $B1(G) = \langle Tv, Tk, V, K, \mathscr{T}, \{Z_1(T'), T' \in \mathscr{T}\}\rangle$, где

$$\left.\begin{array}{l} \forall T' \in \mathscr{T}_2 (Z_1(T') = Z(T')), \\ \forall t \in (Tv \cup Tk) (Z_1(\{t\}) = Z(\{t\}) \cap (\bigcap\limits_{T' \in \mathscr{T}_2(t)} P(Z(T'), t))). \end{array}\right\} \tag{3.9}$$

Вычеркиванием дужек в грамматике G назовем получение грамматики $B2(G) = \langle Tv, Tk, V, K, \mathscr{T}, \{Z_2(T'), T' \in \mathscr{T}\}\rangle$, где

$$\left.\begin{array}{l} \forall t \in (Tv \cup Tk) (Z_2(\{t\}) = Z(\{t\})), \\ \forall \{t_1, t_2\} \in \mathscr{T}_2 (Z_2(\{t_1, t_2\}) = Z(\{t_1, t_2\}) \cap (Z(\{t_1\}) \times Z(\{t_2\}))). \end{array}\right\} \tag{3.10}$$

Определение 5. Операция $B(G) = B1(B2(G))$ называется операцией вычеркивания, а процесс получения последовательности G, $B(G)$, $B(B(G))$, $B(B(B(G)))$, и т. д.— алгоритмом вычеркивания в грамматике G. Алгоритм вычеркивания есть алгоритм получения ядра грамматики, и обоснованием тому служит следующая теорема.

Теорема 2. *Пусть G_i, $i = 1, 2, \ldots,$ — последовательность грамматик, такая, что $G_i = B(G_{i-1})$ для $i = 2, 3, \ldots$ В этом случае для любой грамматики G_1 существует число i^*, такое, что для любого $i > i^*$ выполняется равенство $G_{i+1} = G_i = C(G_1)$.*

Доказательство. По определению вычеркивания для любого $i = 2, 3, \ldots$ справедливо $G_i \subset G_{i-1}$. В то же время очевидно, что если $G_i = G_{i-1}$, то $G_{i+k} = G_i$ для любого k. Очевидно также, что последовательность не равных друг другу грамматик не может быть слишком длинной, так как $\sum\limits_{T' \in \mathscr{T}} |Z(T')|$ есть вполне определенное число, поэтому последовательность грамматик $G_i \subset G_{i-1}$, $G_i \neq G_{i-1}$ не может быть длиннее, чем $\sum_{T' \in \mathcal{T}} |Z(T')| + 1$. (В действительности эта последовательность не может быть длиннее, чем $\sum_{t \in T} |Z(\{t\})| + 1$.)

Прежде всего нам необходимо доказать, что если $G_i = G_{i-1} = G^*$, то грамматика G_i — согласованная. Допустим, что это не так. В таком случае существуют $T' \in \mathcal{T}_2$ и $t \in T'$, такие, что $Z^*(\{t\}) \neq P(Z^*(T'), t)$. Если $Z^*(\{t\}) \supset P(Z^*(T'), t)$, то в результате вычеркивания вершин по (3.9) грамматика изменится, а это противоречит предположению $G_i = G_{i-1}$. Если $Z^*(\{t\}) \subset P(Z^*(T'), t)$, то по правилу вычеркивания дужек (3.10) грамматика изменится. Следовательно, предположение, что $\exists T' \in \mathcal{T} (\exists t \in T' (Z^*(\{t\}) \neq P(Z^*(T), t)))$, не верно, и грамматика G^* является согласованной.

Докажем теперь, что не существует другой такой согласованной грамматики $G' \neq G^*$, что $G^* \subset G' \subset G_1$.

Докажем вначале, что для любой грамматики G и любой согласованной ее подграмматики $\tilde{G}$ справедливо условие $\tilde{G} \subset B1(G)$, где $B1(G)$ — грамматика, получаемая из G в результате вычеркивания вершин (3.9), т. е. мы докажем утверждение

$$(\tilde{G} \subset G) \Rightarrow (\tilde{G} \subset B1(G)), \tag{3.11}$$

если только подграмматика $\tilde{G}$ согласованна.

Напомним справедливость равенства

$$P(Z_1(T') \cup Z_2(T'), t) = P(Z_1(T'), t) \cup P(Z_2(T'), t). \tag{3.12}$$

Введем обозначения:

$$\tilde{G} = \langle Tv, Tk, V, K, \mathcal{T}, \{\tilde{Z}(T'), T' \in \mathcal{T}\}\rangle,$$
$$G = \langle Tv, Tk, V, K, \mathcal{T}, \{Z(T'), T' \in \mathcal{T}\}\rangle,$$
$$G_1 = B1(G) = \langle Tv, Tk, V, K, \mathcal{T}, \{Z_1(T'), T' \in \mathcal{T}\}\rangle.$$

Подграмматика $\tilde{G}$ согласованна, и поэтому равенство

$$\tilde{Z}(\{t\}) = P(\tilde{Z}(T'), t) \tag{3.13}$$

выполняется для всех $T' \in \mathcal{T}_2$ и $t \in T'$.

Поскольку $\tilde{G} \subset G$, равенства

$$\tilde{Z}(T') \subset Z(T'), \quad Z(T') = Z(T') \cup \tilde{Z}(T') \tag{3.14}$$

выполняются для всех $T' \in \mathcal{T}_2$, а равенства

$$\tilde{Z}(\{t\}) \subset Z(\{t\}), \quad Z(\{t\}) = Z(\{t\}) \cup \tilde{Z}(\{t\}) \tag{3.15}$$

выполняются для всех $t \in Tv \cup Tk$.

По определению грамматики $B1(G)$ равенство

$$Z_1(T') = Z'(T) \tag{3.16}$$

справедливо для всех $T' \in \mathcal{T}$, а следовательно, с учетом (3.14) имеем

$$\tilde{Z}(T') \subset Z_1(T'). \tag{3.17}$$

Для любого $t \in Tv \cup Tk$ справедлива следующая цепочка отношений:

$$Z_1(\{t\}) = Z(\{t\}) \cap (\bigcap_{T' \in \mathcal{T}_2(t)} P(Z(T'), t) =$$
$$= Z(\{t\}) \cap (\bigcap_{T' \in \mathcal{T}_2(t)} P(Z(T')) \cup \tilde{Z}(T'), t)) =$$
$$= (Z(\{t\}) \cup \tilde{Z}(\{t\})) \cap (\bigcap_{T' \in \mathcal{T}_2(t)} (P(Z(T'), t) \cup P(\tilde{Z}(T'), t))) =$$
$$= (Z(\{t\}) \cup \tilde{Z}(\{t\})) \cap (\bigcap_{T' \in \mathcal{T}_2(t)} (P(Z(T'), t) \cup \tilde{Z}(\{t\}))) \supset \tilde{Z}(\{t\}).$$

В этой цепочке первое равенство записано по определению грамматики $B1(G)$, второе — в силу (3.14), третье — является следствием (3.15) и (3.12), четвертое — вытекает из (3.13). Последнее соотношение в цепочке очевидно. Из данной цепочки следует справедливость отношения $\tilde{Z}(\{t\}) \subset Z_1(\{t\})$ для всех $t \in Tv \cup Tk$. С учетом (3.17) это означает, что $\tilde{G} \subset B1(G)$.

Докажем теперь, что для любой грамматики G и любой ее согласованной подграмматики $\tilde{G}$ справедливо соотношение $\tilde{G} \subset B2(G)$, где $B2(G)$ — грамматика, получаемая из G в результате вычеркивания дужек (3.10), т. е. мы докажем, что

$$(\tilde{G} \subset G) \Rightarrow (\tilde{G} \subset B2(G)), \tag{3.18}$$

если подграмматика $\tilde{G}$ согласованна.

При введенных ранее обозначениях для G и $\tilde{G}$ введем обозначение еще и для $B2(G)$:

$$G_2 = B2(G) = \langle Tv, Tk, V, K, \mathcal{T}, \{Z_2(T'), T' \in \mathcal{T}\}\rangle.$$

Для любого $t \in Tv \cup Tk$ справедливо включение

$$\tilde{Z}(\{t\}) \subset Z(\{t\}), \tag{3.19}$$

поскольку $\tilde{G} \subset G$.

По определению грамматики $B2(G)$ для любого $t \in Tv \cup Tk$ справедливо равенство $Z_2(\{t\}) = Z(\{t\})$, откуда имеем

$$\tilde{Z}(\{t\}) \subset Z_2(\{t\}). \tag{3.20}$$

Рассмотрим некоторую пару клеток t_1 и t_2, такую, что $\{t_1, t_2\} \in \mathcal{T}_2$. Тогда справедливо соотношение $\tilde{Z}(\{t_1, t_2\}) \subset \subset P(\tilde{Z}(\{t_1, t_2\}), t_1) \times P(\tilde{Z}(\{t_1, t_2\}), t_2$, из которого следует, что

$$\tilde{Z}(\{t_1, t_2\}) \subset \tilde{Z}(\{t_1\}) \times \tilde{Z}(\{t_2\}), \tag{3.21}$$

так как подграмматика $\tilde{G}$ согласованна.

Из справедливости (3.19) вытекает, что

$$\tilde{Z}(\{t_1\}) \times \tilde{Z}(\{t_2\}) \subset Z(\{t_1\}) \times Z(\{t_2\}). \tag{3.22}$$

Следовательно, на основании (3.21) и (3.22) имеем

$$\tilde{Z}(\{t_1, t_2\}) \subset Z(\{t_1\}) \times Z(\{t_2\}). \tag{3.23}$$

Так как $\tilde{G} \subset G$, то справедливо и соотношение

$$\tilde{Z}(\{t_1, t_2\}) \subset Z(\{t_1, t_2\}),$$

из которого с учетом (3.23) вытекает

$$\tilde{Z}(\{t_1, t_2\}) \subset Z(\{t_1, t_2\}) \cup (Z(\{t_1\}) \times Z(\{t_2\})). \tag{3.24}$$

А так как $G_2 = B2(G)$, то справедливо равенство $Z_2(\{t_1, t_2\}) = Z(\{t_1, t_2\}) \cup (Z(\{t_1\}) \times Z(\{t_2\}))$, из которого с учетом (3.24) следует, что $\tilde{Z}(\{t_1, t_2\}) \subset Z_2(\{t_1, t_2\})$, а это значит, что мы доказали (3.18).

Из (3.11) и (3.18) имеем, что для любого i и любой согласованной подграмматики $\tilde{G} \subset G^{i-1}$ справедливо соотношение $\tilde{G} \subset G^i = B(G^{i-1})$. Это значит, что любая согласованная подграмматика $G \subset G^1$ есть подграмматика в грамматике G^*, которая получена алгоритмом вычеркивания из G^1. Следовательно, G^* есть ядро G^1. Теорема доказана.

Теорема 3. *Множество изображений, порождаемых грамматикой G, равно множеству изображений, порождаемых ее ядром $C(G)$.*

Доказательство. Очевидно, что

$$\mathcal{L}(C(G)) \subset \mathcal{L}(G), \tag{3.25}$$

поскольку $C(G) \subset G$.

Докажем, что любое изображение v, допустимое в G, допустимо в $C(G)$.

Тот факт, что $v \in \mathcal{L}(G)$, означает существование варианта $\beta: Tv \cup Tk \to V \cup K$, допустимого в G, такого, что $\beta(Tv) = v$. Построим по этому варианту грамматику $G_\beta = \langle Tv, Tk, V, K, \mathcal{T}, \{\{\beta(T')\}, T' \in \mathcal{T}\}\rangle$. Грамматика G_β есть согласованная подграмматика в G и порождает единственный допустимый вариант β. Как было выяснено в ходе доказательства теоремы 2, $G_\beta \subset C(G)$. Следовательно, $\{\beta\} = S(G_\beta) \subset S(C(G))$, т. е. $\beta \in S(G)$, а $v = \beta(Tv) \in \mathcal{L}(C(G))$. Отсюда $\mathcal{L}(G) \subset \subset \mathcal{L}(C(G))$. Последнее соотношение вместе с (3.25) приводит к равенству $\mathcal{L}(G) = \mathcal{L}(C(G))$. Теорема доказана.

На основании доказанных теорем можно сформулировать довольно очевидную теорему, указывающую на необходимое условие и достаточное условие допустимости изображения в заданной грамматике.

Теорема 4. *Пусть $G = \langle Tv, Tk, V, K, \mathcal{T}, \{Z(T'), T' \in \mathcal{T}\}\rangle$ — двумерная грамматика второго порядка; $v(Tv): Tv \to V$ — изображение; грамматика $G_v = \langle Tv, Tk, V, K, \mathcal{T}, \{Z_v(T'), T' \in \mathcal{T}\}\rangle$, такая, что для всех $t \in Tv$, $Z_v(\{t\}) = Z(\{t\}) \cap \cap \{v(t)\}$, а для всех прочих $T' \in \mathcal{T}$ $Z_v(T') = Z(T')$; $G_v^* = \langle Tv, Tk, V, K, \mathcal{T}, \{Z_v^*(T'), T' \in \mathcal{T}\}\rangle$ — ядро грамматики G_v.*

Тогда, *если существует $t \in Tv \cup Tk$, такое, что $Z_v^*(t) = \varnothing$, то изображение недопустимо.*

Если для всех $t \in Tk$ справедливо $|Z_v^*(\{t\})| = 1$, то изображение допустимо.

Доказательство теоремы не приводится вследствие ее очевидности, а также потому, что далее доказана более сильная теорема.

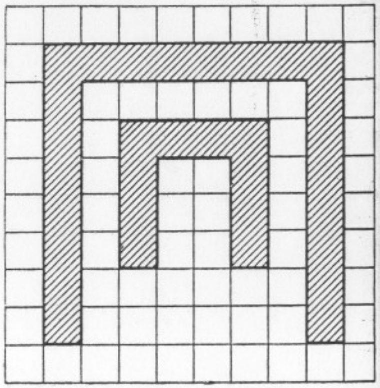
Из теоремы 4 следует исключительно простой алгоритм проверки допустимости, основанный на алгоритме вычеркивания. Своеобразие этого алгоритма заключается в том, что при обработке изображения фактически отсутствует процесс вычисления и построения описания для конкретного изображения, так как это описание уже как бы содержится в памяти алгоритма, но наряду со всеми другими возможными описаниями. Обработка изображения происходит не в виде конструирования нужного описания, а в виде исключения всех ненужных описаний, и это исключение выполняется многократным применением двух простейших правил: (3.9) и (3.10).

6 8-2045

Пример анализа алгоритма вычеркивания. Пусть $Tv = \{(i, j), i = 1, 2, \ldots, n; j = 1, 2, \ldots, m\}$. Для заданных целых чисел i_1, i_2, j_1, j_2 изображением буквы $\Pi(i_1, i_2, j_1, j_2)$ будем называть функцию $\Pi(i_1, i_2, j_1, j_2)\colon Tv \to \{0, 1\}$, такую, что

$$\Pi(i_1, i_2, j_1, j_2)(i, j) = \begin{cases} 1, \text{ если } i = i_2, & j_1 \leqslant j \leqslant j_2; \\ 1, \text{ если } j = j_1, & i_1 \leqslant i \leqslant i_2; \\ 1, \text{ если } j = j_2, & i_1 \leqslant i \leqslant i_2; \\ 0, \text{ во всех прочих случаях.} \end{cases}$$

Допустимым изображением будем считать функцию $v = \sum_{l=1}^{k} v(i_1^l, i_2^l, j_1^l, j_2^l)$, такую, что для любых $l_1, l_2, l_1 \ne l_2$ выполняется по крайней мере одно из следующих четырех неравенств: $i_2^{l_1} < i_1^{l_1} - 1$, $i_2^{l_2} < i_1^{l_1} - 1$, $j_2^{l_1} < j_1^{l_2} - 1$, $j_2^{l_2} < j_1^{l_1} - 1$. На рис. 3.1 показано допустимое изображение, а на рис. 3.2 — недопустимые изображения.



Грамматика, порождающая множество допустимых изображений, имеет вид (рис. 3.3).

поле описания $Tk = \{\tau(t), t \in Tv\}$;

алфавит сигналов $V = \{0, 1\}$;

алфавит структурных элементов $S = \{0, 1, 2, 3, 4, 5, 6\}$;

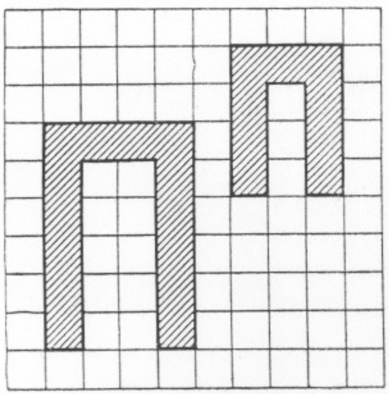


Рис. 3.1. Пример допустимого изображения

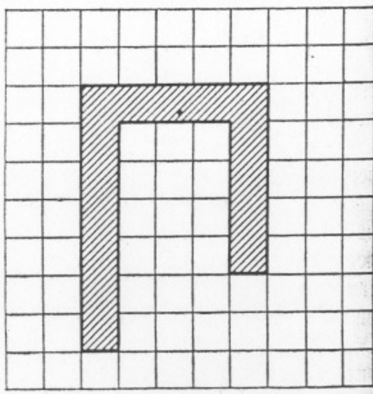


Рис. 3.2. Примеры недопустимых изображений

структура $\mathcal{T} = \{\{\tau(t), t\} : t \in Tv\} \cup \{\{\tau(i, j), \tau(i, j+1)\} : (i, j) \in Tv, j \ne m\} \cup \{\{\tau(i, j), \tau(i+1, j)\} : (i, j) \in Tv, i \ne n\} \cup \{\{\tau(1, j)\} : j = 1, 2, \ldots, m\} \cup \{\{\tau(n, j)\} : j = 1, 2, \ldots, m\} \cup \{\{\tau(i, 1)\} : i = 1, 2, \ldots, n\} \cup \{\{\tau(i, m)\} : i = 1, 2, \ldots, n\}$.

Для пар вида $\{\{\tau(t), t\}, t \in Tv\}$ множество $Z(T')$ равно $(\{0, 1\} \times \{0\}) \cup (\{2, 3, 4, 5, 6\} \times \{1\})$. Это означает, что белым клеткам поля зрения соответствуют структурные элементы 0 и 1, а черным — элементы 2, 3, 4, 5, 6. Множества $Z(T')$ для T' вида $\{\tau(i, j), \tau(i, j+1)\}$ задают ограничения на чередование структурных элементов при переходе от клетки к клетке в горизонтальном направлении слева направо. Как видно из рис. 3.3,

$$Z(T') = (\{0\} \times \{0, 2, 3\}) \cup (\{1\} \times \{1, 6\}) \cup (\{2\} \times \{1\}) \cup (\{3\} \times \{4\}) \cup (\{4\} \times \{4, 5\}) \cup (\{5\} \times \{0\}) \cup (\{6\} \times \{0\}).$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	3	4	4	4	4	5	0	0
0	0	2	1	1	1	1	6	0	0
0	0	2	1	1	1	1	6	0	0
0	0	2	1	1	1	1	6	0	0
0	0	2	1	1	1	1	6	0	0
0	0	2	1	1	1	1	6	0	0
0	0	2	1	1	1	1	6	0	0
0	0	0	0	0	0	0	0	0	0

Рис. 3.3. Пример допустимого варианта

Множество $Z(T')$ для T' вида $\{\tau(i, j), \tau(i+1, j)\}$ имеет вид

$$Z(T') = (\{0\} \times \{0, 1, 2, 6\}) \cup (\{1\} \times \{1, 4\}) \cup (\{2\} \times \{2, 3\}) \cup (\{3, 4, 5\} \times \{0\}) \cup (\{6\} \times \{6, 5\}).$$

Наконец, для всех прочих $T' \in \mathcal{T}$ накладывается ограничение $Z(T') = \{0\}$, для того чтобы сделать недопустимыми изображения, являющиеся частью буквы П.

Рис. 3.4, *а—д* иллюстрируют выполнение процедуры вычеркивания на примере допустимого изображения, а рис. 3.5 — на примере недопустимого изображения. На каждом рисунке представлена совокупность элементов в клетке, невычеркнутых на определенном этапе вычеркивания. Из рис. 3.4, *а—д* видно, что процесс вычеркивания заканчивается в состоянии, когда в каждой клетке оказывается невычеркнутой единственная вершина. Тем самым доказана допустимость изображения. А из рис. 3.5, *а—е* видно, что процесс вычеркивания заканчивается получением клеток (на рис. 3.5, *е* они обозначены наклонными крестами), в которых оказываются вычеркнутыми все вершины, и, следовательно, рассматриваемое изображение недопустимо.

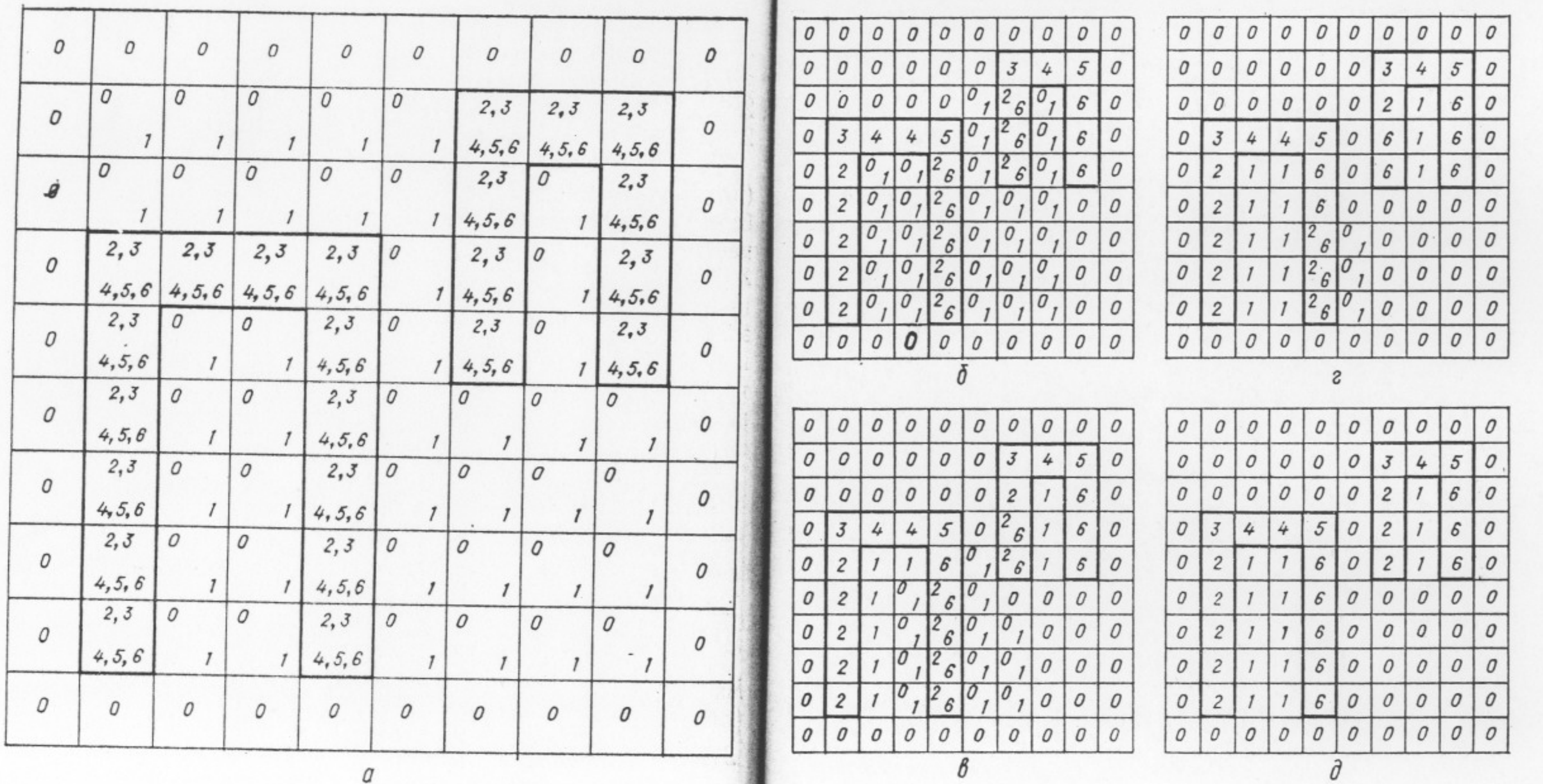


Рис. 3.4. Процесс вычеркивания на допустимом изображении

Таким образом, на двух рассмотренных примерах алгоритм вычеркивания привел к вполне определенному и правильному решению относительно допустимости изображения. Можно доказать, что в данной грамматике алгоритм вычеркивания будет давать такого рода ответы относительно любого изображения, т. е. не будет давать отказ от распознавания. Иными словами, не существует такого изображения, что алгоритм вычеркивания, работая на рассматриваемой грамматике, завершает работу в состоянии, когда в некоторой клетке оказываются невычеркнутыми два и больше символов. Для доказательства этого факта нам достаточно рассмотреть, как будет работать алгоритм вычеркивания, если используется информация лишь об одной строке изображения, скажем, i-й, и доказать, что уже в этом случае применение алгоритма вычеркивания приводит к тому, что в каждой клетке i-й строки поля описания окажется не более одного символа.

Разобьем множество всех возможных строк на два подмножества — подмножество допустимых строк и подмножество недопустимых строк. Допустимая строка имеет вид $q_1, q_2, \ldots, q_k$, где $q_1, q_2, \ldots, q_k$ — допустимые фрагменты, имеющие вид $0^{n_1}10^{n_2}10$ либо $0^{n_3}1^{n_4}0$, $n_1 \geqslant 0$, $n_2 \geqslant 1$, $n_3 \geqslant 0$, $n_4 \geqslant 3$. Все прочие строки считаются недопустимыми.

Пусть i-я строка изображения имеет указанный вид. В таком случае учет лишь ограничений на пары символов в парах клеток вида $\{(i, j), \tau(i, j)\}$ и вида $\{\tau(i, j), \tau(i, j+1)\}$ приводит к тому, что i-я строка поля описания имеет вид $0p_1p_2 \ldots \ldots p_k$, где $p_i = 0^{n_1}21^{n_2}60$, если $q_i = 0^{n_1}10^{n_2}10$, и $p_i = 0^{n_3}34^{n_4-2}50$, если $q_i = 0^{n_3}1^{n_4}0$. Если же i-я строка изображения недопустима, то, как нетрудно убедиться, учет ограничений на те же пары символов приводит к появлению клетки, в которой оказываются вычеркнутыми все символы. Понятно, что учет

0	0	0	0	0	0	0	0	0	0
0	0_1	0_1	0_1	0_1	0_1	0_1	0_1	0_1	0
0	0_1	2,3 4,5,6	2,3 4,5,6	2,3 4,5,6	2,3 4,5,6	2,3 4,5,6	0_1	0_1	0
0	0_1	2,3 4,5,6	0_1	0_1	0_1	2,3 4,5,6	0_1	0_1	0
0	0_1	2,3 4,5,6	0_1	0_1	0_1	2,3 4,5,6	0_1	0_1	0
0	0_1	2,3 4,5,6	0_1	0_1	0_1	2,3 4,5,6	0_1	0_1	0
0	0_1	2,3 4,5,6	0_1	0_1	0_1	2,3 4,5,6	0_1	0_1	0
0	0_1	2,3 4,5,6	0_1	0_1	0_1	0_1	0_1	0_1	0
0	0_1	2,3 4,5,6	0_1	0_1	0_1	0_1	0_1	0_1	0
0	0	0	0	0	0	0	0	0	0

а

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	3	4	4	4	5	0_1	0	0
0	0	2_6	0_1	0_1	0_1	2_6	0_1	0	0
0	0	2_6	0_1	0_1	0_1	2_6	0_1	0	0
0	0	2_6	0_1	0_1	0_1	2_6	0_1	0	0
0	0	2_6	0_1	0_1	0_1	2_6	0_1	0	0
0	0	2_6	0_1	0_1	0_1	0_1	0_1	0	0
0	0	2_6	0_1	0_1	0_1	0_1	0_1	0	0
0	0	0	0	0	0	0	0	0	0

б

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	3	4	4	4	5	0	0	0
0	0	2	1	1	1	6	0	0	0
0	0	2	0_1	0_1	0_1	2_6	0	0	0
0	0	2	0_1	0_1	0_1	2_6	0	0	0
0	0	2	0_1	0_1	0_1	2_6	0	0	0
0	0	2	0_1	0_1	0_1	0_1	0	0	0
0	0	2	0_1	0_1	0_1	0_1	0	0	0
0	0	0	0	0	0	0	0	0	0

в

Рис. 3.5. Процесс вычеркивания на недопустимом изображении

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	3	4	4	4	5	0	0	0
0	0	2	1	1	1	6	0	0	0
0	0	2	1	1	1	6	0	0	0
0	0	2	1	0_1	0_1	6	0	0	0
0	0	2	1	0_1	0_1	6	0	0	0
0	0	2	1	0_1	0_1	0	0	0	0
0	0	2	1	0_1	0_1	0	0	0	0
0	0	0	0	0	0	0	0	0	0

г

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	3	4	4	4	5	0	0	0
0	0	2	1	1	1	6	0	0	0
0	0	2	1	1	1	6	0	0	0
0	0	2	1	1	1	6	0	0	0
0	0	2	1	1	1	6	0	0	0
0	0	2	1	1	0	0	0	0	0
0	0	2	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

д

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	3	4	4	4	5	0	0	0
0	0	2	1	1	1	6	0	0	0
0	0	2	1	1	1	6	0	0	0
0	0	2	1	1	1	6	0	0	0
0	0	2	1	1	1	6	0	0	0
0	0	2	1			0	0	0	0
0	0	2	1			0	0	0	0
0	0	0	0	0	0	0	0	0	0

е

ограничений на пары клеток вида $\{\tau(i, j), \tau(i+1, j)\}$ может лишь уменьшить количество невычеркнутых вершин.

Таким образом, рассматриваемая грамматика, порождающая изображения буквы П, такова, что применение алгоритма вычеркивания приводит к полной определенности относительно допустимости любого изображения и не приводит к отказу от распознавания. Описание примера закончено.

Подобное доказательство безотказности алгоритма вычеркивания должно быть выполнено для каждого конкретного случая его применения. Как правило, если автор грамматики достаточно ясно представляет себе множество допустимых изображений и понимает содержательный смысл того описания, которое он хочет получить для каждого изображения, то вопрос о безотказности грамматики имеет очевидный ответ — положительный или отрицательный. При отсутствии ясного понимания, что требуется от алгоритма описания изображения (а это не такая уж редкая ситуация), следует пользоваться более сложными описанными ниже алгоритмами или, что зачастую может оказаться значительно эффективнее, предусмотреть участие человека в технологической линии обработки изображений, возложив на него задачу снятия неопределенности, если алгоритм вычеркивания закончился отказом.

2. Деревообразные структуры. Пусть T — множество, $\mathcal{T} \subset T \times T$ — структура.

Определение 6. Структуру $(T, \mathcal{T})$ будем называть деревообразной, если $|T| \leqslant 1$, или если для любых двух клеток t_0, t_k, таких, что $t_0 \in T$, $t_k \in T$, $t_0 \neq t_k$, существует не более одной последовательности $t_0, t_1, \ldots, t_k$, такой, что $\{t_i, t_{i+1}\} \in \mathcal{T}$ для всех $i = 0, 1, \ldots, k-1$, и $t_i \neq t_j$ для всех $i \neq j$. Если такая последовательность существует для всех t_0, t_k, $t_0 \neq t_k$, то деревообразную структуру будем называть деревом.

Лемма 1. *Для любой деревообразной структуры $(T, \mathcal{T})$ существует такая нумерация клеток $q(t)$, что для каждой клетки $t \in T$ существует не более одной клетки t', такой, что $(t, t') \in \mathcal{T}$ и $q(t') < q(t)$.*

Доказательство. Укажем алгоритм отыскания требуемой нумерации. Алгоритм работает по шагам и на каждом шаге номер присваивается некоторой очередной клетке. В процессе работы алгоритма некоторые связные компоненты структуры $(T, \mathcal{T})$ могут обладать тем свойством, что некоторым клеткам уже присвоены номера, а некоторым другим — еще нет. Такие связные компоненты назовем неполностью пронумерованными и будем производить нумерацию клеток так, чтобы выполнялось следующее условие:

после каждого шага алгоритма имеется не более одной неполностью пронумерованной связной компоненты и, кроме того, подмножество пронумерованных клеток в этой связной компоненте является связным.

На первом шаге алгоритма выбирается любая клетка и ей присваивается номер 1. Очевидно, что после этого шага выполняется только что указанное условие. Допустим, что уже пронумерованы i вершин, причем их нумерация удовлетворяет требованию, сформулированному в лемме, а подмножество пронумерованных клеток — указанному выше условию. После выполнения i-го шага алгоритма может оказаться одна из двух ситуаций: либо имеется ненумерованная клетка, соседняя с нумерованной, либо такая клетка не существует. Рассмотрим каждую из этих ситуаций.

Пусть t^* — ненумерованная клетка, соседняя с нумерованной клеткой t'. Отметим, что в этом случае не существует другой нумерованной клетки $t'' \neq t'$, соседней с t^*. Действительно, если такая клетка имеется, то клетки t' и t'' принадлежат одной связной компоненте структуры $(T, \mathcal{T})$, причем неполностью пронумерованной компоненте. По указанному выше условию подмножество пронумерованных клеток этой компоненты связно. Это значит, что существует путь из t' в t'', проходящий только по нумерованным клеткам, и имеется другой путь t', t^*, t'', содержащий ненумерованную клетку t^*, а это противоречит условию о деревообразности структуры. Клетка t^*, таким образом, соседствует с единственной нумерованной клеткой t', и поэтому присвоение ей номера $i + 1$ не нарушает требований к нумерации, указанных в лемме.

Если нет ненумерованной клетки, соседней с нумерованной, то любой ненумерованной клетке присваивается номер $i + 1$. Поскольку клетка с номером $i + 1$ не соседствует ни с какой клеткой с меньшим номером, то в этом случае также не нарушается требование к нумерации.

Как в первой, так и во второй ситуации после выполнения $(i + 1)$-го шага не нарушается сформулированное выше условие и имеются предпосылки для выполнения следующего шага. После выполнения $|T|$-го шага получена нумерация, существование которой и требовалось доказать. Лемма доказана.

Сформулируем теперь теорему, указывающую более сильные достаточные условия допустимости изображения, чем указанные в теореме 4.

Теорема 5. *Пусть $G = \langle Tv, Tk, V, K, \mathcal{T}, \{Z(T'), T' \in \mathcal{T}\}\rangle$, $v : Tv \to V$ — изображение, $G(v) = \langle Tv, Tk, V, K, \mathcal{T}, \{Z_v(T'), T' \in \mathcal{T}\}\rangle$, где $Z_v(\{t\}) = Z(\{t\}) \cap \{v(t)\}$ для всех $t \in Tv$, а для всех прочих $T' \in \mathcal{T}$ $Z_v(T') = Z(T')$; $C(G(v)) = \langle Tv, Tk, V, K, \mathcal{T}, \{Z_v^*(T'), T' \in \mathcal{T}\}\rangle$ — ядро грамматики $G(v)$.*

В таком случае если для любой клетки $t \in Tk$ справедливо неравенство $|Z_v^*(\{t\})| \geqslant 1$ и является деревообразным множество клеток, для которых $|Z_v^*(\{t\})| > 1$, то изображение v допустимо.

Доказательство. Пусть $T^* = \{t : t \in Tk, |Z_v^*(t)| > 1\}$. Если $T^* = \varnothing$, то совокупность невычеркнутых элементов образует описание изображения v, которое, таким образом, является допустимым. Если $T^* \neq \varnothing$, то определим для $t \in T^*$ номер $q(t)$, как это требуется в лемме, и будем определять $k : Tk \to K$ для клеток $t \in T^*$ поочередно в соответствии с их номерами.

Вначале определим $k(t)$ для клетки t с номером $q(t) = 1$ так, чтобы $k(t) \in Z_v^*(\{t\})$. Клетка t, возможно, является соседней с клеткой $t' \in Tk \setminus T^*$, однако в t' имеется единственный структурный элемент $k(t')$, причем такой, что $(k(t'), k(t)) \in Z(\{t', t\})$. В противном случае структурный элемент $k(t')$ был бы вычеркнут в клетке t.

Допустим, что уже определены структурные элементы $k(t)$ для всех клеток $t \in T^*$, имеющих номера $q(t) \leqslant q_0$. Клетку с номером $q_0 + 1$ обозначим через t^*, и найдем клетку t', имеющую номер, не больший, чем q_0, и соседнюю с t^*. Если такой клетки нет, то в качестве $k(t^*)$ выбираем любой элемент из $Z_v^*(\{t^*\})$. В противном случае выбираем $k(t^*)$ так, чтобы

$(k(t^*),\ k(t')) \in Z_v^*(\{t^*,\ t'\})$. Такой элемент обязательно найдется, так как в противном случае элемент $k(t^*)$ был бы вычеркнут, т. е. не входил бы в множество $Z_v^*(\{t'\})$. Клетка t, возможно, является соседней с некоторой клеткой $t'' \in T \setminus T^*$, однако в клетке t'' есть лишь единственный невычеркнутый элемент $k(t'')$, причем такой, что $(k(t^*),\ k(t'')) \in Z_v^*(t^*,\ t'')$, потому что в противном случае структурный элемент $k(t^*)$ был бы вычеркнут.

Таким способом мы наращиваем множество клеток, в которых однозначно определен структурный элемент, пока все множество не будет исчерпано. Полученное описание по своему построению будет допустимым описанием изображения v, что доказывает допустимость v. Теорема доказана.

Следствие. В деревообразной грамматике изображение допустимо тогда и только тогда, когда в результате алгоритма вычеркивания по крайней мере в одной клетке окажется невычеркнутым хотя бы один элемент.

Доказательство основано на том очевидном факте, что любое подмножество деревообразного множества также является деревообразным. Следовательно, в результате применения алгоритма вычеркивания мы непременно получим грамматику, удовлетворяющую условиям теоремы 5. Доказательство закончено.

Мы видим, что деревообразные структуры — это, безусловно, хорошие структуры, позволяющие безотказно определять допустимость изображения с помощью простейших средств, а именно с помощью алгоритмов вычеркивания. Тем не менее существует огромный разрыв по сложности между деревообразными структурами и структурами общего вида. В одном из последующих разделов мы этот разрыв сгладим, т. е. произведем более подробную классификацию структур по сложности с разработкой алгоритмов анализа, усложняющихся с ростом сложности структуры постепенно, а не скачком.

3.2. РАСПОЗНАВАНИЕ РЕАЛЬНЫХ ИЗОБРАЖЕНИЙ

В дальнейшем нет необходимости в раздельном рассмотрении поля зрения Tv и поля описания Tk, равно как и в разделении алфавита символов на алфавит сигналов и алфавит структурных элементов. Поэтому обозначим $Tv \cup Tk$ через T, а $V \cup K$ — через S. Кроме того, везде при дальнейшем изложении по-прежнему предполагается, что структура грамматики — это структура второго порядка, включающая в себя множество $\{\{t\} : t \in T\}$.

1. Формулировка задачи. Пусть $G = \langle T,\ S,\ \mathcal{T},\ \{Z(T'),\ T' \in \mathcal{T}\}\rangle$ — грамматика, $s(T) : T \to S$ — вариант, такой, что

$\forall T' \in \mathcal{T}\ (s(T') \in Z(T'))$; $S(G)$ — множество таких вариантов; $f : (S \times T) \to R$ — функция с вещественными значениями. Требуется найти допустимый вариант s^*, такой, что

$$\sum_{t \in T} f(s^*(T),\ t) = \max_{s \in S(G)} \sum_{t \in T} f(s(t),\ t).$$

Существует много содержательно осмысленных примеров, для которых адекватна приведенная формулировка. Укажем некоторые из них.

А. Практические задачи не всегда сводятся к одному только распознаванию допустимости изображения. Для недопустимых изображений, например, недостаточно просто констатировать их недопустимость, а требуется еще и указать степень этой недопустимости, т. е. определить множество клеток поля зрения, изменение сигнала в которых превращает данное недопустимое изображение в некоторое допустимое. При этом, естественно, требуется, чтобы количество этих клеток было как можно меньшим.

Найденную совокупность клеток, определяющих недопустимость изображения, можно использовать по-разному. Можно ее вывести оператору для принятия окончательного решения о целесообразности исправления (редакции) изображения, а можно найденный оптимальный вариант считать уже исправленным и производить дальше его содержательную обработку.

Б. При наблюдении изображения в условиях помех любое наблюдаемое изображение, как правило, недопустимо. В этом случае возникает задача исключения помех, которая обычно ставится следующим образом. Предполагается существование некоторого множества X значений реального сигнала x. Воздействие помех заключается в том, что значение сигнала $v(t)$ в клетке t заменяется сигналом $x(t) \in X$ с вероятностью $p(x/v)$. Исключение помех понимается как отыскание наиболее правдоподобного допустимого изображения, т. е. изображения, в котором вероятность $\prod_{t \in T} p(x(t)/s(t))$ максимальна. Понятно, что эта задача сводится к сформулированной оптимизационной задаче при $f(s(t),\ t) = 0$ для всех $t \in Tk$, $s(t) \in Z(\{t\})$ и $f(v(t),\ t) = \log p(x(t)/v(t))$ для всех $t \in Tv$.

В. В обработке изображений распространена ситуация, когда множество допустимых изображений совпадает с множеством V^{Tv} всех возможных изображений. Однако каждому из изображений соответствует не одно, а много описаний. Среди них имеются более желательные и менее желательные, и задача заключается в отыскании для любого изображения наилучшего его описания. Формально эти требования выражаются в выделении в алфавите S структурных элементов подмножества $S' \subset S$ и нахождении такого описания, чтобы структурные элементы из S' были выбраны как можно в меньшем количестве клеток.

2. Метод кажущихся качеств. Для заданной грамматики $G = \langle T, S, \mathscr{T}, \{Z(T'), T' \in \mathscr{T}\}\rangle$ и функции $f: S \times T \to R$ введем следующие обозначения. По-прежнему будем считать, что $\mathscr{T} = \mathscr{T}_1 \cup \mathscr{T}_2$, где $\mathscr{T}_1 = \{\{t\} : t \in T\}$, а $\mathscr{T}_2 \subset T \times \times T$.

Множество $\bar{\sigma} = (S \times T)$ будем называть множеством вершин. Помимо функции f, заданной на этом множестве, будем считать заданными еще две функции ($t_\sigma : \bar{\sigma} \to T$ и $s_\sigma : \bar{\sigma} \to S$), такие, что для любой вершины $\sigma^* = (s^*, t^*) \in \bar{\sigma}$ справедливо равенство $t_\sigma(\sigma^*) = t^*$, $s_\sigma(\sigma^*) = s^*$. Буква σ в записях t_σ и s_σ — это не индекс, принимающий какое-то множество значений, а часть обозначения (идентификатора). Для заданной вершины σ величина $f(\sigma)$ называется ее качеством, $t_\sigma(\sigma)$ — местом, а $s_\sigma(\sigma)$ — наименованием. Для любой клетки $t \in T$ множество $\{\sigma : \sigma \in \bar{\sigma},\ t_\sigma(\sigma) = t\}$ будет называться множеством вершин, помещенных в клетку t, и обозначаться через $\bar{\sigma}(t)$. Две клетки t_1 и t_2, такие, что $\{t_1, t_2\} \in \mathscr{T}_2$, будут называться соседними. Через $\overline{\sigma\sigma}$ обозначим множество пар (σ_1, σ_2) вершин, расположенных в соседних клетках, т. е. $\{t_\sigma(\sigma_1), t_\sigma(\sigma_2)\} \in \mathscr{T}_2$, и таких, что $(s_\sigma(\sigma_1), s_\sigma(\sigma_2)) \in Z(\{t_\sigma(\sigma_1), t_\sigma(\sigma_2)\})$. Будем считать, что вершины σ_1 и σ_2 соединены дужкой, если $(\sigma_1, \sigma_2) \in \overline{\sigma\sigma}$. Для любой пары соседних клеток t_1 и t_2, $\{t_1, t_2\} \in \mathscr{T}_2$ через $\overline{\sigma\sigma}(t_1, t_2)$ обозначим множество

$$\{(\sigma_1, \sigma_2) : (\sigma_1, \sigma_2) \in \overline{\sigma\sigma}, \quad t_\sigma(\sigma_1) = t_1,\ t_\sigma(\sigma_2) = t_2\}.$$

Двумерная грамматика G, таким образом, является множеством T клеток t и множеством $\bar{\sigma}$ вершин, разделенным на непересекающиеся множества $\bar{\sigma}(t)$ вершин, помещенных в клетке t. Некоторые пары клеток объявлены соседними, а некоторые пары вершин, помещенных в соседние клетки, соединены дужками. Вариант — это подмножество выбранных вершин, такое, что в каждой клетке $t \in T$ выбрана одна и только одна вершина. Допустимым считается такой вариант, что вершины, выбранные в соседних клетках, соединены дужкой. Если задана функция $f : \bar{\sigma} \to R$, то оптимизационная задача состоит в поиске допустимого варианта с максимальной суммой качеств входящих в него вершин. Эту задачу назовем задачей двумерного программирования.

Первоначальные правдоподобные рассуждения по поводу решения задачи двумерного программирования основаны на следующих двух простых идеях. Первая идея заключается в том, что существуют такие функции $f_0 : \bar{\sigma} \to R$, при которых для любого допустимого варианта $s \in S(G)$ суммарное качество $\sum_{t \in T} f_0(s(t), t)$ равно нулю. В этом случае решение любой задачи двумерного программирования с функцией качества f совпадает с решением задачи двумерного программирования с функцией кажущегося качества $f + f_0$. Такие две задачи могут быть названы эквивалентными.

Вторая идея заключается в том, что существуют такие функции $f : \bar{\sigma} \to R$, при которых решение оптимизационной задачи оказывается тривиальным. Допустим, функция f такова, что совокупность вершин $s^*(t) = \arg\max_{s(t) \in S} f(s(t), t)$, $t \in T$, образует допустимый вариант, т. е. выбор в каждой клетке вершины с наилучшим качеством привел к тому, что вершины, выбранные в соседних клетках, оказались соединенными дужками. В данном случае указанный вариант, конечно же, является оптимальным допустимым вариантом. Задача с такой функцией качества вершин может быть названа тривиальной.

Первоначальные правдоподобные соображения по решению задачи двумерного программирования заключаются в нахождении для исходной функции f такой функции f_0, чтобы задача с функцией $f + f_0$ была тривиальной, эквивалентной исходной задаче, которая нетривиальна.

Эти соображения реализуются в следующих построениях. Для каждой клетки $t \in T$ через $M(t)$ обозначим множество $\{t' : t' \in T, \{t, t'\} \in \mathscr{T}_2\}$ клеток, соседних с t.

Для каждой вершины $\sigma \in \bar{\sigma}$ и клетки $\tau \in M(t_\sigma(\sigma))$, соседней с клеткой, в которой σ помещена, введем в рассмотрение число $\varphi(\sigma, \tau)$, такое, что совокупность чисел $\varphi(\sigma, \tau)$, $\sigma \in \bar{\sigma}$, $\tau \in M(t_\sigma(\sigma))$ удовлетворяет следующему условию: для каждой пары вершин $(\sigma_1, \sigma_2) \in \overline{\sigma\sigma}$, соединенной дужкой, должно выполняться неравенство

$$\varphi(\sigma_1, t_\sigma(\sigma_2)) + \varphi(\sigma_2, t_\sigma(\sigma_1)) \leqslant 0. \tag{3.26}$$

Выражение в левой части (3.26) обозначим через $\delta(\sigma_1, \sigma_2)$. Совокупность чисел $\varphi(\sigma, \tau)$, $\sigma \in \bar{\sigma}$, $\tau \in M(t_\sigma(\sigma))$ обозначим буквой φ и назовем набором потенциалов.

Для заданного набора потенциалов φ и функции качества f число $h(\sigma, \varphi) = f(\sigma) - \sum_{\tau \in M(t_\sigma(\sigma))} \varphi(\sigma, \tau)$ будет называться кажущимся качеством вершины σ, а $K(\varphi) = \sum_{t \in T} \max_{\sigma \in \sigma(t)} h(\sigma, \varphi)$ — кажущимся максимумом.

Для любого варианта $s: T \to S$ обозначим его качество $\sum_{t\in T} f(s(t), t)$ через $F(s)$.

Теорема 6. *Пусть φ — любой набор потенциалов, а s^* — любой допустимый вариант. В таком случае $K(\varphi) \geqslant F(s^*)$.*

Доказательство. Будем говорить, что вершина σ входит в вариант s^*, если $s^*(t_\sigma(\sigma)) = s_\sigma(\sigma)$. Множество вершин, входящих в вариант s^*, обозначим через $B_\sigma(s^*)$. И будем говорить, что дужка $(\sigma_1, \sigma_2) \in \overline{\sigma\sigma}(t_\sigma(\sigma_1), t_\sigma(\sigma_2))$ входит в вариант s^*, если $\sigma_1 \in B_\sigma(s^*)$ и $\sigma_2 \in B_\sigma(s^*)$. Множество дужек, входящих в вариант s^*, обозначим через $D_\sigma(s^*)$.

Для любого допустимого варианта s^* справедливо равенство

$$\sum_{\sigma\in B_\sigma(s^*)} \sum_{\tau\in M(t_\sigma(\sigma))} \varphi(\sigma, \tau) = \sum_{(\sigma_1,\sigma_2)\in D_\sigma(s^*)} (\varphi(\sigma_1, t_\sigma(\sigma_2)) + \\ + \varphi(\sigma_2, t_\sigma(\sigma_1))), \tag{3.27}$$

так как любой потенциал $\varphi(\sigma, \tau)$ либо входит по одному разу в каждую из двух сумм в этом выражении (если $\sigma \in B_\sigma(s^*)$), либо не входит ни в одну из сумм (если $\sigma \notin B_\sigma(s^*)$). Сумма в правой части (3.27) неположительна по определению (3.26) набора φ. Следовательно, сумма в левой части также неположительна. Из этого непосредственно следует, что

$$\sum_{\sigma\in B_\sigma(s^*)} \sum_{\tau\in M(t_\sigma(\sigma))} \varphi(\sigma, \tau) \leqslant 0; \tag{3.28}$$

$$\sum_{t\in T} f(s^*(t), t) - \sum_{\sigma\in B_\sigma(s^*)} \sum_{\tau\in M(t_\sigma(\sigma))} \varphi(\sigma, \tau) \geqslant F(s^*). \tag{3.29}$$

Левая часть неравенства (3.29) представляет собой сумму кажущихся качеств вершин σ, входящих в вариант s^*, т. е. равна $\sum_{\sigma\in B_\sigma(s^*)} h(\sigma)$, в результате чего (3.29) может быть переписано в несколько ином виде:

$$\sum_{\sigma\in B_\sigma(s^*)} h(\sigma) \geqslant F(s^*). \tag{3.30}$$

Очевидным является неравенство

$$\sum_{t\in T} \max_{\sigma\in\overline{\sigma}(t)} h(\sigma) \geqslant \sum_{\sigma\in B_\sigma(s^*)} h(\sigma).$$

Левая часть его есть кажущийся максимум на наборе φ, что позволяет записать

$$K(\varphi) \geqslant \sum_{\sigma\in B_\sigma(s^*)} h(\sigma). \tag{3.31}$$

Совместное использование (3.31) и (3.30) приводит к неравенству

$$K(\varphi) \geqslant F(s^*).$$

Теорема доказана.

Введенные нами потенциалы — это средство, которое позволяет производить эквивалентное преобразование задачи двумерного программирования. Действительно, если качеством варианта s^* считать не сумму $\sum_{\sigma\in B_\sigma(s^*)} f(\sigma)$, а сумму $\sum_{\sigma\in B_\sigma(s^*)} h(\sigma) + \sum_{(\sigma_1,\sigma_2)\in D_\sigma(s^*)} \delta(\sigma_1, \sigma_2)$, то с учетом формулы (3.27) и определения кажущегося качества вершины ясно, что эти две суммы равны между собой при любом допустимом наборе потенциалов.

В то же время потенциалы позволяют преобразовать исходную грамматику G, порождающую все возможные допустимые варианты, в другую грамматику G_φ, которая порождает только оптимальные допустимые варианты. Это значит, что с помощью потенциалов происходит как бы отбор из всех вариантов только тех, для которых значение $\sum_{t\in T} f(s(t), t)$ максимально.

Покажем, как это происходит. Для заданной грамматики G и допустимого набора φ потенциалов через G_φ обозначим подграмматику исходной грамматики G, получающуюся из нее вычеркиванием из множества $\overline{\sigma}$ тех вершин, которые не максимальны по значению $h(\sigma)$ в своей клетке, т. е. для которых $h(\sigma) \neq \max_{\sigma'\in\overline{\sigma}(t_\sigma(\sigma))} h(\sigma')$, и вычеркиванием тех дужек (σ_1, σ_2), приращение $\delta(\sigma_1, \sigma_2)$ на которых не равно нулю.

Теорема 7. *Если φ — допустимый набор потенциалов, то любой вариант, допустимый в G_φ, есть оптимальный допустимый вариант в исходной грамматике G.*

Доказательство. Любой вариант, допустимый в G_φ, является допустимым и в G, так как $G_\varphi \subset G$ по определению.

Доказательство оптимальности варианта, допустимого в G_φ, сводится к доказательству, что значение функции $F(s)$ для этого варианта равно значению кажущегося максимума $K(\varphi)$ для набора φ.

Пусть s^* — допустимый в G_φ вариант. Для него справедливо равенство

$$\sum_{\sigma\in B_\sigma(s^*)} \sum_{\tau\in M(t_\sigma(\sigma))} \varphi(\sigma, \tau) = \sum_{(\sigma_1,\sigma_2)\in D_\sigma(s^*)} (\varphi(\sigma_1, t_\sigma(\sigma_2)) + \\ + \varphi(\sigma_2, t_\sigma, (\sigma_1))), \tag{3.32}$$

полученное при доказательстве теоремы 6.

Выражение в правой части (3.32) равно нулю, так как все дужки, для которых $\delta(\sigma_1, \sigma_2) \neq 0$, вычеркнуты по определению G_φ и не входят в грамматику G_φ, а следовательно, не входят в вариант s^*, допустимый в G_φ. Поэтому сумма в левой части (3.32) также равна нулю и

$$\sum_{t\in T} f(s(t), t) - \sum_{\sigma\in B_\sigma(s^*)} \sum_{\tau\in M(t_\sigma(\sigma))} \varphi(\sigma, \tau) = F(s^*). \tag{3.33}$$

Левая часть (3.33) является суммой кажущихся качеств вершин σ, входящих в вариант s^*, и поэтому равенство (3.33) может быть записано в виде

$$\sum_{\sigma\in B_\sigma(s^*)} h(\sigma, \varphi) = F(s^*). \tag{3.34}$$

Справедливо также равенство

$$\sum_{\sigma\in B_\sigma(s^*)} h(\sigma, \varphi) = \sum_{t\in T} \max_{\sigma\in\bar{\sigma}(t)} h(\sigma), \tag{3.35}$$

так как все те вершины σ_1, для которых $h(\sigma_1, \varphi) \neq \neq \max\limits_{\sigma\in\bar{\sigma}(t_\sigma(\sigma_1))} h(\sigma, \varphi)$, вычеркнуты по определению G_φ и, следовательно, не входят в допустимый вариант s^*. В правой части (3.35) записан кажущийся максимум $K(\varphi)$ на наборе φ, поэтому на основании (3.34) и (3.35) имеем $F(s) = K(\varphi)$. Из последнего равенства и на основании теоремы 6 имеем, что $F(s^*) \geqslant F(s)$, где s — любой вариант, допустимый в G. Теорема доказана.

Из теоремы 7 следует, что задача поиска оптимального варианта, допустимого в грамматике G, сводится к поиску набора потенциалов, такого, что грамматика $G(\varphi)$ оказывается непротиворечивой. Из теоремы 6 следует, что если такой набор φ существует, то он обеспечивает минимум потенциального значения $K(\varphi)$. Однако минимум функции $K(\varphi)$ может достигаться на некотором множестве наборов φ, и остается открытым вопрос, непротиворечива ли грамматика $G(\varphi)$, если φ есть набор с минимальным значением кажущегося максимума $K(\varphi)$ и если известно, что существует набор φ^*, такой, что грамматика G_{φ^*} непротиворечива. На этот вопрос положительный ответ дает следующая теорема.

Теорема 8. *Если существует такой набор φ_1 потенциалов, что грамматика G_{φ_1} непротиворечива, то для любого набора φ_2, минимизирующего функцию $K(\varphi)$, грамматика G_{φ_2} также будет непротиворечивой.*

Д о к а з а т е л ь с т в о. Величины h и δ будем писать с индексом 1, если они вычислены по набору φ_1, и с индексом 2, если они вычислены по φ_2.

Докажем, что любой вариант s^*, допустимый в грамматике G_{φ_1}, допустим и в грамматике G_{φ_2}.

Вариант s^* допустим в G_{φ_1}, и поэтому в силу теоремы 7

$$\sum_{t\in T} \max_{\sigma\in\bar{\sigma}(t)} h_1(\sigma) = \sum_{\sigma\in B_\sigma(s^*)} h_1(\sigma) = F(s^*). \tag{3.36}$$

Из допустимости набора φ_2 следует

$$\sum_{t\in T} \max_{\sigma\in\bar{\sigma}(t)} h_2(\sigma) \geqslant \sum_{\sigma\in B_\sigma(s^*)} h_2(\sigma) \geqslant F(s^*). \tag{3.37}$$

По условию теоремы φ_2 минимизирует $K(\varphi)$ так же, как и φ_1. Поэтому

$$\sum_{t\in T} \max_{\sigma\in\bar{\sigma}(t)} h_1(\sigma) = \sum_{t\in T} \max_{\sigma\in\bar{\sigma}(t)} h_2(\sigma). \tag{3.38}$$

При совместном использовании (3.36) — (3.38) получаем

$$\sum_{t\in T} \max_{\sigma\in\bar{\sigma}(t)} h_2(\sigma) = \sum_{\sigma\in B_\sigma(s^*)} h_2(\sigma), \tag{3.39}$$

$$\sum_{\sigma\in B_\sigma(s^*)} h_2(\sigma) = F(s^*). \tag{3.40}$$

Из (3.39) следует, что для любого $t\in T$ справедливо равенство $h_2(\sigma) = \max\limits_{\sigma_1\in\bar{\sigma}(t(\sigma))} h_2(\sigma_1)$, а это значит, что ни одна из вершин, входящих в вариант s^*, не вычеркнута в грамматике G_{φ_2}.

Из (3.40) следует, что

$$\sum_{\sigma\in B_\sigma(s^*)} \sum_{\tau\in M(t_\sigma(\sigma))} \varphi_2(\sigma, \tau) = 0.$$

Из последнего равенства в сочетании с равенством

$$\sum_{\sigma\in B_\sigma(s^*)} \sum_{\tau\in M(t_\sigma(\sigma))} \varphi_2(\sigma, \tau) = \sum_{(\sigma_1,\sigma_2)\in D_\sigma(s^*)} (\varphi_2(\sigma_1, t_\sigma(\sigma_2)) + $$
$$+ \varphi_2(\sigma_2, t_\sigma(\sigma_1))),$$

полученным при доказательстве теоремы 6, следует

$$\sum_{(\sigma_1,\sigma_2)\in D_\sigma(s^*)} \delta_2(\sigma_1, \sigma_2) = 0. \tag{3.41}$$

По условию допустимости φ_2 все слагаемые в (3.41) являются неположительными, и поэтому из равенства (3.41) следует, что все эти слагаемые равны нулю. Следовательно, любая дужка, входящая в вариант, невычеркнута и в грамматике G_{φ_2}. Из этого (вместе с доказанным выше аналогичным утверждением относительно вершин, входящих в вариант s^*) следует, что вариант s^* является допустимым и в грамматике G_{φ_2}. Теорема доказана.

Задача минимизации функции $K(\varphi)$ есть задача линейного программирования и, конечно, может решаться многими известными алгоритмами выпуклой оптимизации. Однако для нашей задачи специального вида возможны более частные, а следовательно, и более эффективные алгоритмы. Возможность построения такого алгоритма следует из сформулированной далее теоремы, причем ясность по этому вопросу вносится не столько самой формулировкой теоремы, сколько ее доказательством.

Теорема 9. *Если для некоторой грамматики G и набора φ грамматика G_φ имеет пустое ядро, то существует такой набор φ', что $K(\varphi') < K(\varphi)$.*

Доказательство. Обозначим грамматику G_φ через G_1 и с помощью алгоритма вычеркивания построим последовательность грамматик $G_1, G_2, G_3, \ldots$, где $G_i = B1(B2(G_{i-1}))$, $B1$ — вычеркивание вершин, $B2$ — вычеркивание дужек. Грамматика $G_\varphi = G_1$ имеет пустое ядро, а это значит, что существует такое целое число n, что в грамматике G_n имеется клетка t^*, в которой все вершины оказываются вычеркнутыми, а в грамматике G_{n-1} еще такой клетки нет. Эту клетку будем называть нуль-клеткой. Каждой вершине σ, имеющейся в грамматике G_φ и отсутствующей в грамматике G_n, т. е. каждой вычеркнутой вершине σ, поставим в соответствие номер $i(\sigma)$, обозначающий, что вершина σ еще имеется в грамматике $G_{i(\sigma)}$ и уже вычеркнута в грамматике $G_{i(\sigma)+1}$. О каждой вычеркнутой вершине σ будем говорить, что она вычеркнута на $i(\sigma)$-м шаге вычеркивания. Вершинам σ, которые имеются в грамматике G, но отсутствуют в грамматике $G_1 = G_\varphi$, присвоим номер $i(\sigma) = 0$. Это те вершины, которые имеют кажущееся сходство $h(\sigma)$, не максимальное в своей клетке. О таких вершинах будем говорить, что они вычеркнуты на нулевом шаге вычеркивания.

Построим граф Г, состоящий из некоторых вычеркнутых вершин, соединенных определенным образом направленными ребрами. Ребра будут называться стрелками. Потребуем, чтобы граф обладал следующими свойствами:

1. Все вершины σ нуль-клетки t^*, имеющие ненулевой номер $i(\sigma)$, входят в граф Г.

2. Если некоторая стрелка выходит из вершины σ_1 и входит в вершину σ_2, то вершины σ_1 и σ_2 находятся в соседних клетках, соединены дужкой в грамматике G, а номер вершины σ_1 больше номера вершины σ_2.

3. Для любой вершины σ в графе Г, из которой не выходят стрелки, выполняется по крайней мере одно из следующих двух условий: а) либо $i(\sigma) = 0$; б) либо существует такая клетка τ, соседняя с клеткой $t_\sigma(\sigma)$, что для любой вершины $\sigma' \in \bar{\sigma}(\tau)$ пара вершин σ и σ' не соединена дужкой в грамматике G_φ.

4. Все вершины, входящие в граф Г, входят в грамматику G и не входят в грамматику G_n, т. е. вычеркнуты.

Построение графа Г выполняется по шагам так, что после любого выполненного шага будет получен граф, удовлетворяющий условиям пунктов 1, 2, 4, и не обязательно удовлетворяющий условию п. 3.

Первый шаг построения графа Г заключается в построении графа, вершинами которого являются вершины нуль-клетки, имеющие ненулевой номер, а множество стрелок пусто. Очевидно, что таким образом построенный граф удовлетворяет условиям пунктов 1, 2 и 4. Допустим, что уже выполнено несколько шагов построения графа, и условия пунктов 1, 2 и 4 по-прежнему выполняются. Если к тому же выполняется и условие п. 3, то построение графа закончено. В противном случае находим вершину σ', которая не удовлетворяет условию п. 3, и выполняем очередной шаг построения графа Г.

Найденная вершина σ' вычеркнута, так как удовлетворяется условие п. 4; номер $i(\sigma')$ не равен нулю, поскольку в противном случае выполнялось бы условие п. 3, а). Следовательно, эта вершина вычеркнута уже в процессе работы алгоритма вычеркивания. Значит, существует клетка τ, соседняя с клеткой $t_\sigma(\sigma')$, такая, что любая вершина $\sigma'' \in \bar{\sigma}(\tau)$ образует с найденной вершиной σ' такую пару, что дужка (σ', σ'') уже отсутствует в грамматике $G_{i(\sigma')}$. Отсюда следует, что в момент вычеркивания вершины σ' любая пара (σ', σ''), $\sigma'' \in \bar{\sigma}(\tau)$, уже была вычеркнута. Некоторые из этих пар имелись в грамматике G_φ, так как в противном случае для вершины σ' выполнялось бы условие п. 3, б). Следовательно, некоторые из вершин в клетке τ, соединенные дужкой с вершиной σ', вычеркнуты, причем вычеркнуты они были еще до вычеркивания вершины σ'. Каждая из этих вершин включается в граф Г и соединяется стрелкой с вершиной σ' так, что стрелка выходит из вершины σ' и входит в только что введенную вершину. Понятно, что такое наращивание графа Г не нарушает условий пунктов 1, 2 и 4, а вершина σ' перестает быть вершиной, из которой не выходят стрелки. Понятно также, что процесс наращивания графа не может происходить бесконечно долго, так как любая вершина или стрелка, введенная в граф, из него не исключается.

Поскольку для построенного графа Г выполняется указанное ранее условие п. 2, этот граф не имеет циклов. Исходя из этого свойства графа укажем, как получить набор φ', такой, что $K(\varphi') < K(\varphi)$. Вычисление набора φ' происходит по шагам, на каждом из которых происходит изменение графа Г

Γ и изменение набора φ. Изменение графа Γ заключается в удалении из него вершины, не имеющей выходящих стрелок, одновременно с удалением всех стрелок, входящих в эту вершину. Ясно, что после каждого удаления, если только оставшаяся часть графа не является пустой, то в ней опять будет содержаться вершина, не имеющая выходящих стрелок, так как направленный граф, не содержащий циклов, при удалении любой вершины остается направленным графом, не содержащим циклов. Одновременно с удалением каждой вершины графа Γ будет изменяться набор φ таким образом, чтобы после каждого шага изменения графа сам граф и очередной набор φ обладали следующим свойством.

Свойство. Пусть σ — вершина графа Γ, не имеющая выходящих стрелок. В таком случае набор φ может быть изменен так, что кажущееся качество $h(\sigma, \varphi)$ уменьшится, а кажущийся максимум $K(\varphi)$ не увеличится.

Первоначально из графа Γ исключаются все вершины, имеющие нулевой номер. Из этих вершин в силу указанного ранее условия п. 2 не выходят стрелки. Удаление из Γ каждой такой вершины σ сопровождается удалением всех входящих в нее стрелок и следующим изменением величин $\varphi(\sigma, \tau)$. Пусть $\sigma_1, \sigma_2, \ldots, \sigma_k$ — вершины, из которых выходят стрелки, входящие в σ, а $\tau_1, \tau_2, \ldots, \tau_m$ — клетки, в которых эти вершины находятся. В таком случае для всех $i = 1, 2, \ldots, k$ справедливо равенство

$$\varphi(\sigma, t_\sigma(\sigma_i)) + \varphi(\sigma_i, t_\sigma(\sigma)) = 0. \tag{3.42}$$

Пусть $h(\sigma, \varphi)$ — кажущееся качество вершины σ. Вершина σ вычеркнута на нулевом шаге вычеркивания, и поэтому

$$\max_{\sigma' \in \bar{\sigma}(t_\sigma(\sigma))} h(\sigma', \varphi) - h(\sigma, \varphi) = \Delta > 0. \tag{3.43}$$

Если вершина σ не находится в нуль-клетке, то величины $\varphi(\sigma, \tau_i)$, $i = 1, 2, \ldots, m$, могут быть заменены такими величинами $\varphi'(\sigma, \tau_i)$, $i = 1, 2, \ldots, m$, что будут выполняться равенства

$$\varphi'(\sigma, t_\sigma(\sigma_i)) + \varphi(\sigma_i, t_\sigma(\sigma)) = -\frac{\Delta}{m} < 0, \tag{3.44}$$

$$\max_{\sigma' \in \bar{\sigma}(t_\sigma(\sigma))} h(\sigma', \varphi') - h(\sigma, \varphi') = 0. \tag{3.45}$$

Для этого необходимо выбрать $\varphi'(\sigma, \tau_i)$ так, чтобы

$$\varphi'(\sigma, \tau_i) = \varphi(\sigma, \tau_i) - \frac{\Delta}{m}, \quad i = 1, 2, \ldots, m.$$

Если вершина σ находится в нуль-клетке, то значения $\varphi(\sigma, \tau_i)$, $i = 1, 2, \ldots, m$, могут быть изменены так, что выполнятся

условия

$$\varphi'(\sigma, t_\sigma(\sigma_i)) + \varphi(\sigma_i, t_\sigma(\sigma)) = -\frac{\Delta}{m+1} < 0, \tag{3.46}$$

$$\max_{\sigma' \in \bar{\sigma}(t_\sigma(\sigma))} h(\sigma', \varphi) - h(\sigma, \varphi') = \frac{\Delta}{m+1} < 0. \tag{3.47}$$

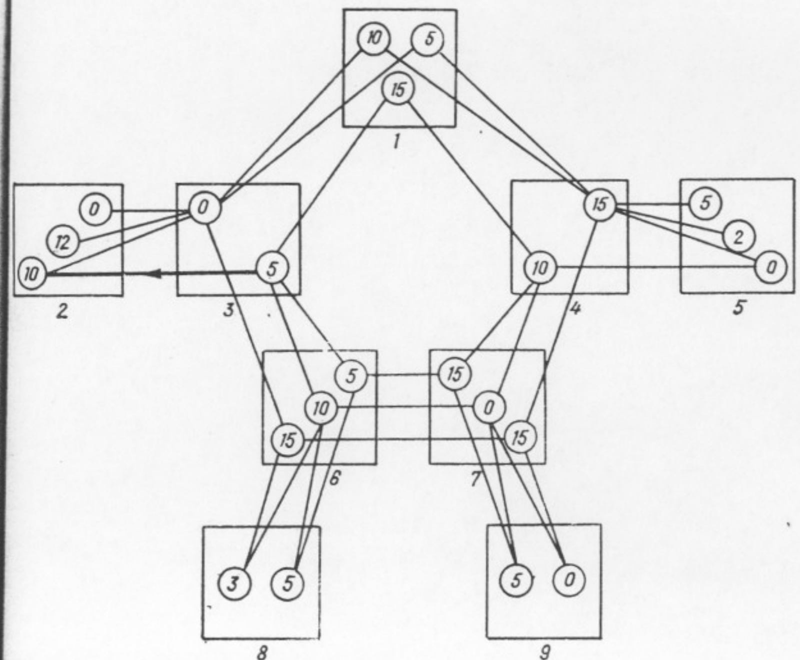


Рис. 3.6. Исходные данные задачи двумерного программирования и вычеркивание на первой итерации

Для этого необходимо выбрать новые значения $\varphi'(\sigma, \tau_i)$ равными

$$\varphi(\sigma, \tau_i) - \frac{\Delta}{m+1}, \quad i = 1, 2, \ldots, m.$$

При указанном изменении $\varphi(\sigma, \tau_i)$, $i = 1, 2, \ldots, m$, новые значения их таковы, что для всех вершин σ', из которых выходит стрелка, входящая в вершину σ, значение $\delta(\sigma, \sigma')$ будет строго отрицательным. Поэтому после удаления из графа Γ всех вершин с нулевым номером вместе с входящими в них стрелками и одновременным изменением $\varphi(\sigma, \tau_i)$ указанное ранее свойство для оставшейся части графа Γ будет выполняться. Дальнейшее изменение φ происходит однотипно.

Пусть σ — вершина графа Γ, из которой не выходят стрелки. Значит, существует такая клетка τ, что для всех вершин $\sigma' \in \bar{\sigma}(\tau)$, соединенных дужкой с σ, значение $\delta(\sigma, \sigma')$ является строго отрицательным. Следовательно, величина $\varphi(\sigma, \tau)$ может быть заменена величиной $\varphi'(\sigma, \tau) = \varphi(\sigma, \tau) + \Delta$, где Δ — положительное число, так что значения $\delta(\sigma, \sigma')$

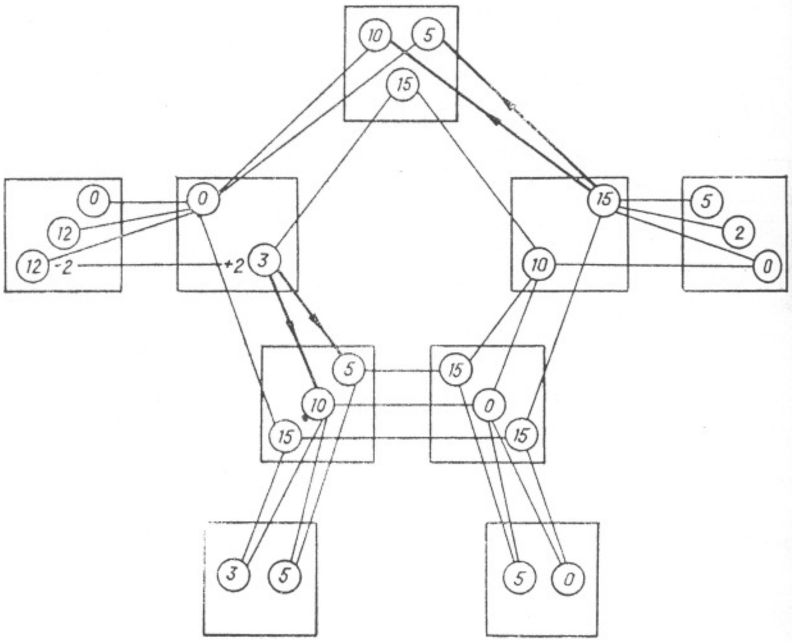


Рис. 3.7. Состояние алгоритма после первой итерации и вычеркивание на второй и третьей итерациях

останутся неположительными. Пусть в вершину σ' входили стрелки, выходящие из вершины $\sigma_1, \sigma_2, \ldots, \sigma_k$, находящихся в клетках $\tau_1, \tau_2, \ldots, \tau_m$. Это значит, что одновременно с изменением $\varphi(\sigma, \tau)$ следует изменить и $\varphi(\sigma, \tau_i)$, $i = 1, 2, \ldots, m$. Причем если вершина σ находится в нуль-клетке, то $\varphi'(\sigma, \tau_i) = \varphi(\sigma, \tau_i) - \frac{\Delta}{m+1}$, $i = 1, 2, \ldots, m$, а в противном случае $\varphi'(\sigma, \tau_i) = \varphi(\sigma, \tau_i) - \frac{\Delta}{m}$. При указанном изменении φ после удаления вершины σ и входящих в нее стрелок для оставшейся части графа Γ будет выполняться указанное выше свойство. Кроме того, если вершина σ находится в нуль-клетке, то ее кажущееся качество уменьшится. Таким образом, в резуль-

тате замены набора φ набором φ' кажущиеся качества некоторых вершин увеличиваются, а именно тех вершин, кажущиеся качества которых были немаксимальны в своей клетке, кажущиеся качества других вершин не изменяются и, наконец, кажущиеся качества всех вершин, имеющих максимальные качества в нуль-клетке, уменьшатся. Теорема доказана.

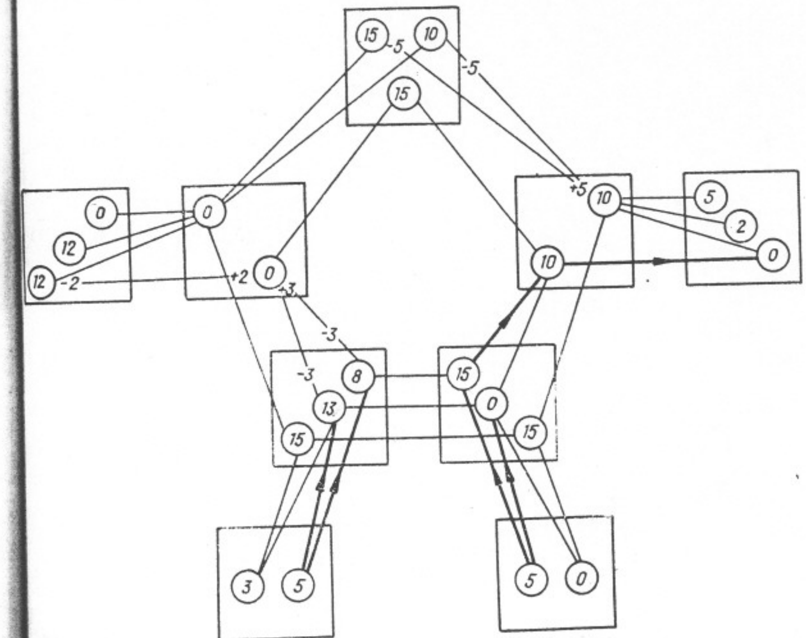


Рис. 3.8. Состояние алгоритма после третьей итерации и вычеркивания на четвертой и пятой итерациях

Пример поиска оптимального варианта. Рис. 3.6—3.10 иллюстрируют работу описанного алгоритма. На рис. 3.6 изображено поле T, состоящее из 9 клеток. В каждой клетке расположены две или три вершины, обозначенные кружками. Из рисунка ясно, какие пары клеток являются соседними, а какие — нет. На этом же рисунке внутри кружочка записано качество вершины, т. е. величина $f(\sigma)$. На последующих рисунках внутри кружочков записаны уже кажущиеся качества, т. е. значение $h(\sigma)$. Ненулевые значения потенциалов $\varphi(\sigma, \tau)$, обусловившие это изменение, записаны вне кружочка, причем со стороны клетки τ. Из рис. 3.6, представляющего первоначальные качества, видно, что грамматика G_φ при $\varphi = 0$ не содержит ядра. Действительно, в клетке *2* оказывается вычеркнутой вершина с качеством 10, поскольку оно не максимально; по этой же причине в клетке *3* вычеркнута вершина с качеством 0; вершина с качеством 5 в этой клетке вычеркнута, поскольку ранее уже оказалась вычеркнутой единственная вершина в клетке *2*, соединенная с ней дужкой. Процесс вычер-

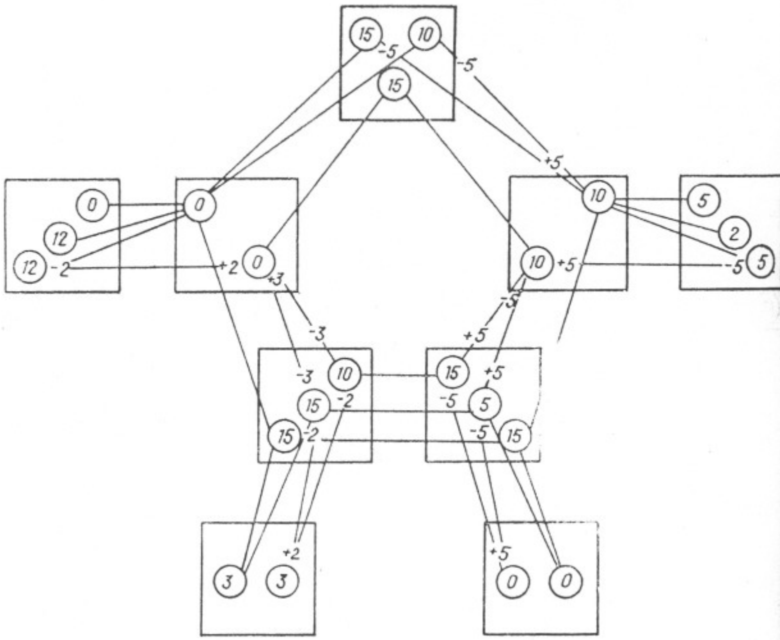


Рис. 3.9. Состояние алгоритма после пятой итерации (получена грамматика с непустым ядром)

кивания, т. е. граф Γ, на каждом рисунке изображен утолщенными линиями, причем если стрелка направлена от вершины σ_1 к вершине σ_2, то это значит, что вначале была вычеркнута вершина σ_2, а затем, как следствие этого,— вершина σ_1. Эти же стрелки иллюстрируют изменение потенциалов. А именно: если стрелка из какой-то вершины σ выходит в сторону τ, то потенциал φ (σ, τ) должен измениться так, чтобы качество вершины σ уменьшилось; если стрелка входит в какую-то вершину σ со стороны клетки τ, то потенциал φ (σ, τ) должен измениться так, чтобы увеличилось качество вершины σ. Поэтому качество тех вершин, из которых стрелки только выходят, уменьшается, качество вершин, имеющих только входящие стрелки, увеличивается, а качество всех прочих вершин остается неизменным. Значения потенциалов после пер-

вого их изменения и новые значения качеств приведены на рис. 3.7. На этом же рисунке отображен процесс вычеркивания в задаче с новыми значениями качеств, причем отображены сразу две итерации этого процесса. Очередные две итерации показаны на рис. 3.8. Потенциалы, полученные после пяти

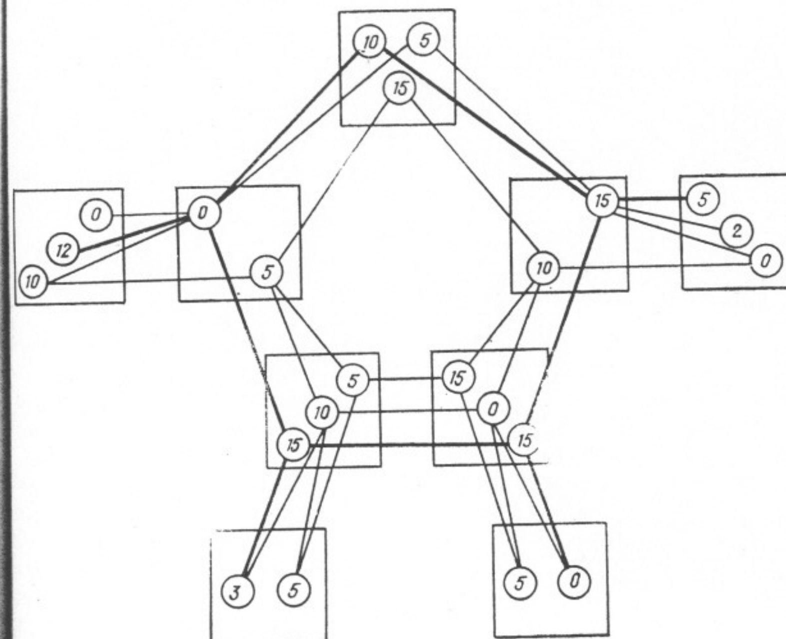


Рис. 3.10. Исходные данные задачи двумерного программирования и найденный оптимальный вариант

итераций изменения набора φ, представлены на рис. 3.9. Грамматика G_φ для этого набора уже имеет непустое ядро, содержащее единственный допустимый вариант, представленный на рис. 3.10. Этот вариант оптимальный. Пример закончен.

Описанный метод решения оптимизационной задачи, как и алгоритм распознавания идеальных изображений, допускает ситуацию отказа, которая возникает, когда для некоторого найденного набора φ ядро грамматики G_φ не является пустым, и не является деревообразным множество клеток, в которых оказалось более одной невычеркнутой клетки. Для грамматик, имеющих деревообразную структуру, естественно, ситуация отказа исключена. В следующем параграфе мы упорядочим структуры грамматик по сложности, а затем выведем безотказные алгоритмы решения рассмотренных ранее и других задач.

3.3. КЛАССИФИКАЦИЯ ДВУМЕРНЫХ ГРАММАТИК ПО СЛОЖНОСТИ

Множество T и структура $\mathcal{T}$ второго порядка определяют симметричный (неориентированный) граф с множеством вершин T и множеством ребер $\mathcal{T}$. Этот граф будем обозначать буквой Γ.

Для целого числа $k \geqslant 0$ определим операцию k-включения вершины t^* в граф $\Gamma = (T, \mathcal{T})$. Пусть $\Gamma^* = (T^*, \mathcal{T}^*)$ — полный подграф графа Γ, такой, что $|T^*| \leqslant k$; k-включение вершины $t^* \notin T$ в граф $(T, \mathcal{T})$ есть получение графа $(T \cup \{t^*\}, \mathcal{T} \cup \{\{t^*, t\} : t \in T^*\})$, т. е. введение в граф вершины t^*, ранее в нем отсутствовавшей, и соединение этой вершины ребрами со всеми вершинами из T^*.

Для базового графа k-й степени дано следующее рекуррентное определение. Для любого целого $k \geqslant 0$ граф является базовым графом k-й степени, если он пуст или если он получен из некоторого базового графа k-й степени в результате k-включения вершины. Последовательность графов $\Gamma_0 = (\varnothing, \varnothing)$, $\Gamma_1, \Gamma_2, \ldots, \Gamma_{n-1}, \Gamma_n$, такая, что каждый граф Γ_i, $i = 2, 3, \ldots, n$, получен из графа Γ_{i-1} с помощью операции k-включения вершины, называется выводом графа Γ_n. Любой подграф базового графа k-й степени называется графом k-й степени. Грамматика $G = \langle T, S, \mathcal{T}, \Omega \rangle$ называется грамматикой k-й степени, если $(T, \mathcal{T})$ есть граф k-й степени.

Примеры. Все деревообразные грамматики — это грамматики первой степени и только они. Цикл — это граф второй степени. Не любой граф второй степени является циклом. Описание примеров закончено.

В дальнейшем мы увидим, что степень грамматики является важнейшей ее характеристикой, определяющей трудоемкость возникающих здесь задач распознавания. Поэтому важно понять, достаточно ли легко распознаваема эта характеристика. Иными словами, нас интересует сложность алгоритма, который для любого графа и целого числа k определяет, является ли данный граф графом k-й степени или нет. Несколько более сложное построение алгоритма, который для каждого графа определяет минимальное целое число k, такое, что предъявленный граф имеет степень k. Для ответов на эти вопросы введем понятие k-исключения вершины, где k — определенное целое число.

Пусть $t \in T$ — одна из вершин графа $(T, \mathcal{T})$; $T^*(t) = \{t' : t' \in T, \{t, t'\} \in \mathcal{T}\}$ — подмножество вершин, соединенных с t ребрами, $\mathcal{T}_1(t) = \{\{t, t'\} : t' \in t^*(t)\}$ — подмножество ребер, выходящих из t, $\mathcal{T}_2(t) = \{\{t_1, t_2\} : t_1 \in T^*(t), t_2 \in T^*(t)\}$. Операция k-исключения вершины $t \in T$ применима к графу $(T, \mathcal{T})$, если $|T^*(t)| \leqslant k$, и заключается в получении графа

$$(T \setminus \{t\}, (\mathcal{T} \setminus \mathcal{T}_1(t)) \cup \mathcal{T}_2(t)).$$

Теорема 10. *Для того чтобы граф $\Gamma_0 = (T_0, \mathcal{T}_0)$ был графом k-й степени, необходимо и достаточно существование последовательности графов $(T_0, \mathcal{T}_0)$, $(T_1, \mathcal{T}_1)$, ..., $(T_{|T_0|}, \mathcal{T}_{|T_0|})$, такой, что $T_{|T_0|} = \varnothing$, $\mathcal{T}_{|T_0|} = \varnothing$, а для любого $i = 1, 2, \ldots, |T_0|$ граф $(T_i, \mathcal{T}_i)$ был получен из $(T_{i-1}, \mathcal{T}_{i-1})$ в результате k-исключения некоторой вершины $t \in T_{i-1}$.*

Доказательство. Вначале докажем, что если существует последовательность графов, указанная в формулировке теоремы, то граф Γ_0 имеет степень k.

Построим граф Γ_0', равный $(T_0 \bigcup_{i=0}^{|T_0|} \mathcal{T}_i)$. Граф Γ_0 — это подграф графа Γ_0'. В свою очередь, Γ_0' — базовый полный граф k-й степени. Последнее верно, так как последовательность графов $\left(\bigcup_{j=|T_0|-i}^{|T_0|} T_j, \bigcup_{j=|T_0|-i}^{|T_0|} \mathcal{T}_j \right)$, $i = 0, 1, \ldots, |T_0|$, образует вывод графа Γ_0' степени k. Следовательно, граф Γ_0 имеет степень k.

Докажем теперь, что для любого графа Γ_0 степени k существует последовательность, указанная в формулировке теоремы. Мы докажем это, если докажем, что в любом графе Γ_0 степени k существует вершина t, k-исключение которой применимо к графу Γ_0, и получаемый в результате этого исключения граф также имеет степень k. Для графа Γ_0 степени k существует базовый надграф k-й степени, который обозначим через Γ_n'. Для графа Γ_n' существует вывод $\Gamma_0', \Gamma_1', \ldots, \Gamma_{n-1}', \Gamma_n'$. Присвоим всем вершинам графа Γ_n' и Γ_0 номера, так, чтобы вершина с номером i имелась в графе Γ_i' и еще отсутствовала в графе Γ_{i-1}'. Найдем в графе Γ_0 вершину с максимальным номером. Без потери общности рассуждений можно считать, что этот максимальный номер равен n. Если же он равен не n, а $m < n$, то в качестве базового надграфа следует принять граф Γ_m', который также имеет степень k. Таким образом, можно считать, что вершина, включенная последней в базовый надграф Γ_n', осталась и в графе Γ_0. Обозначим эту вершину через t^*. Очевидно, что в графе Γ_n' из вершины t^* выходит не более чем k ребер. Следовательно, в графе Γ_0 из этой вершины также выходит не более чем k ребер. Поэтому k-исключение вершины t^* применимо как к графу Γ_n', так и к графу Γ_0. Применение этой операции дает соответственно графы Γ_1 и Γ_{n-1}', причем Γ_1 есть подграф в Γ_{n-1}', который является базовым графом k-й степени. Следовательно, Γ_1 — граф k-й степени. Теорема доказана.

Мы видим, таким образом, что степень графа принципиально может быть распознана, хотя непосредственно из теоремы следует чересчур трудоемкий алгоритм распознавания, требующий перебора $|T_0|!$ вариантов. Для сокращения перебора существенное значение имела бы теорема о том, что k-исключение любой вершины из графа k-й степени дает граф k-й степени. Чтобы доказать эту теорему, потребуется выполнить вспомогательное построение.

Пусть в графе $\Gamma = (T, \mathcal{T})$ имеются две вершины (t_1 и t_2), соединенные ребром. И пусть $T^*(t_2)$ — множество вершин, соединенных ребром с t_2, а $\mathcal{T}^*(t_2)$ — множество ребер, выходящих из t_2. Склеиванием вершин t_1 и t_2 будем называть получение графа $(T \setminus \{t_2\}, (\mathcal{T} \setminus \mathcal{T}^*(t_2)) \cup \{\{t_1, t\} : t \in T^*(t_2)\}$. Иными словами, склеивание вершин t_1 и t_2 — это исключение вершины t_2, всех входящих в нее ребер и подсоединение к t_1 всех вершин, ранее соединенных с t_2.

Теорема 11. *Пусть граф Γ_n имеет степень k, а t' и t'' — две вершины этого графа, соединенные ребром. В таком случае склеивание вершин t' и t'' дает также граф k-й степени.*

Доказательство. Для любого графа H, содержащего вершины t' и t'', соединенные ребром, через $H^{\text{скл}}$ обозначен граф, получаемый из H склеиванием вершин t' и t''.

Справедливо следующее утверждение. Если Γ_2 есть подграф в Γ_1, и оба эти графа содержат вершины t' и t'', соединенные ребром, то $\Gamma_2^{\text{скл}}$ есть подграф в $\Gamma_1^{\text{скл}}$. В силу справедливости этого утверждения при доказательстве теоремы можно ограничиться рассмотрением случая, когда Γ_n — базовый граф k-й степени.

Пусть $\Gamma_0, \Gamma_1, \Gamma_2, \ldots, \Gamma_n$ — вывод базового графа Γ_n; $t_1, t_2, \ldots, t_n$ — последовательность вершин, такая, что граф Γ_i получен в результате k-включения вершины t_i в граф Γ_{i-1}; T_i^*, $i = 1, 2, \ldots, n$,— множество вершин, соединенных с вершиной t_i в графе Γ_i; числа l и m, $l < m$,— номера вершин t' и t'' в последовательности $t_1, t_2, \ldots, t_n$. Все графы Γ_i являются базовыми графами k-й степени, в том числе и граф Γ_{m-1}. Для него справедливо к тому же равенство $\Gamma_{m-1} = \Gamma_m^{\text{скл}}$, потому что множество вершин T_m^*, соединенных в графе Γ_m с вершиной t_m, образует полный граф и включает в себя t_l. Поэтому склеивание вершин t_l и t_m в графе Γ_m означает просто исключение вершины t_m из графа Γ_m вместе со всеми ребрами, выходящими из t_m. Во всех множествах $T_{m+1}^*, T_{m+2}^*, \ldots, T_n^*$, содержащих вершину t_m и не содержащих вершину t_l, заменим t_m на t_l и исключим вершину t_l из тех множеств, которые содержат t_l. Измененные множества обозначим так: $T_{m+1}^{*\text{скл}}, T_{m+2}^{*\text{скл}}, \ldots, T_n^{*\text{скл}}$. Включим клетку t_{m+1} в граф $\Gamma_{m-1} = \Gamma_m^{\text{скл}}$, соединив ее со всеми вершинами из $T_{m+1}^{*\text{скл}}$. Очевидно, что таким образом будет получен граф $\Gamma_{m+1}^{\text{скл}}$. Подобным образом получим графы $\Gamma_{m+2}^{\text{скл}}$ и т. д. до $\Gamma_n^{\text{скл}}$. Количество вершин в каждом из множеств $T_{m+1}^{*\text{скл}}, T_{m+2}^{*\text{скл}}, \ldots, T_n^{*\text{скл}}$ не превышает k, и поэтому $\Gamma_n^{*\text{скл}}$ есть граф n-й степени. Теорема доказана.

Теорема 12. *Пусть Γ — граф k-й степени, а $\dot{t}^*$ — любая его вершина, соединенная ребрами не более чем с k вершинами. В таком случае k-исключение этой вершины также дает граф k-й степени.*

Доказательство основано на справедливости следующего утверждения: k-включение вершин в граф k-й степени, не обязательно базовый, дает граф k-й степени.

Граф Γ имеет степень k, поэтому согласно теореме 10 существует последовательность вершин $t_1, t_2, \ldots, t_n$, такая, что k-исключение вершины t_1 из Γ дает граф Γ_1 k-й степени, а k-исключение вершины t_i из графа Γ_{i-1} — граф Γ_i k-й степени.

Пусть T^* — множество вершин в Γ, соединенных ребрами с вершиной t^*. Множество $T^* \cup \{t^*\}$ представим в виде $\{t_{i_1}, t_{i_2}, \ldots, t_{i_{m-1}}, t_{i_m}\} \ldots$, где $i_1, i_2, \ldots, i_m$ — номера вершин из $T^* \cup \{t^*\}$ в последовательности $t_1, t_2, \ldots, t_m$. Последовательность $t_{i_1}, t_{i_2}, \ldots, t_{i_m}$ упорядочена в том смысле, что $i_1 < i_2 < \ldots < i_m$. Обозначим через i^* номер вершины t^* в последовательности $t_1, t_2, \ldots, t_m$, а через j^* — число, такое, что $i_{j^*} = i^*$. Число m не превосходит $k + 1$, так как вершина t^* соединена ребрами не более чем с k вершинами.

Докажем теорему, расчленив операцию k-исключения вершины t^* на ряд операций, гарантированно не меняющих степень графа.

1. Выполним последовательно k-исключения вершины t_1 из графа Γ, затем t_2 из графа Γ_1 и т. д. вплоть до k-исключения вершины t^* включительно. Полученный граф имеет степень k, поскольку таково свойство специально выбранной последовательности $t_1, t_2, \ldots, t_n$. Однако в этом графе множество вершин $t_{i_{j^*+1}}, t_{i_{j^*+2}}, \ldots, t_{i_m}$, уже образуют полный граф.

2. Некоторому числу j присваиваем значение j^*. Эта операция не меняет графа.

3. Включаем вершину t^* в граф, соединяя ее ребрами с вершинами $t_{i_j}, t_{i_{j+1}}, \ldots, t_{i_m}$, исключая, естественно, из этого списка вершину t^*. Количество вершин в списке не превос-

ходит k, поэтому такое изменение графа является также k-включением, а полученный граф имеет степень k.

4. Производим последовательно k-включение всех вершин последовательности, начиная с $t_{i_{j-1}}$ и кончая $t_{i_{j-1}}$. Из справедливости утверждения, данного в начале доказательства, вытекает, что эта операция также не выводит граф из класса графов степени k.

5. Производим склеивание вершины t^* с вершиной $t_{i_{j-1}}$, присваивая результирующей вершине обозначение $t_{i_{j-1}}$. В силу теоремы 11 полученный граф остается k-й степени, однако теперь в дополнение к вершинам $t_{i_j}, \ldots, t_{i_m}$, образовавшим полный граф, добавляется вершина $t_{i_{j-1}}$, которая соединяется со всеми вершинами $t_{i_j}, \ldots, t_{i_m}$, и подмножество вершин, соединенных с t^* и образующих полный граф, увеличивается на единицу.

6. Число j уменьшается на единицу. Если число j не равно единице, то повторяем процедуру изменения графа, начиная с п. 3.

7. Выполняем последовательно k-включение вершин из последовательности t_1, t_2, ..., t_n, начиная с вершины t_{i_1-1} и до t_1. В результате выполнения указанных действий мы получаем граф, имеющий степень k и содержащий в качестве подграфа результат k-исключения вершины t^*. Теорема доказана.

Ранее уже рассматривались теоремы, в доказательстве которых приводился алгоритм, и в этом смысле доказательство теоремы представляло больший интерес, чем теорема. Приведенный в доказательстве последней теоремы алгоритм, эквивалентный k-исключению вершины, не имеет самостоятельного значения, а служит только для доказательства того важного факта, что k-исключение вершины не меняет степени графа. Наиболее простая реализация k-исключения клетки есть та, которая дана уже в определении.

Мы видим, таким образом, что степень графа может распознаваться не только принципиально, но и с помощью несложного алгоритма. А именно: для того чтобы определить, имеет ли граф степень k, необходимо найти в этом графе вершину, из которой выходит не более чем k ребер, выполнить ее k-исключение, и производить эти действия либо до тех пор, пока не будет получен пустой граф, либо до тех пор, пока после очередного k-исключения в графе не окажется вершин, имеющих не более чем k ребер. При первом исходе граф имеет степень k, а при втором не имеет степени k.

Так же легко может быть найдено минимальное число k,

такое, что заданный граф имеет степень k. Для этого необходимо в качестве очередной исключаемой вершины выбирать вершину с наименьшим количеством ребер.

3.4. ПОЛИНОМИАЛЬНЫЕ ХАРАКТЕРИСТИКИ МНОЖЕСТВ

Пусть T и S — конечные множества, и для любого подмножества $T' \subset T$ определено $S^{T'}$ — множество всех возможных функций вида $T' \to S$, имеющих одну и ту же область определения T'. Множество $S^{T'}$ будем считать определенным и для $T' = \varnothing$ и понимать его как множество, содержащее единственный элемент, обозначаемый знаком $\#$ и называемый пустой функцией. Для любого $T' \subset T$ через $\hat{s}(T')$ будем обозначать множество всех возможных подмножеств функций вида $T' \to S$, т. е. $\hat{s}(T') = 2^{S^{T'}}$. Через 2^T обозначим множество всех подмножеств в T. И, наконец, через $\tilde{s}$ обозначим $\bigcup\limits_{T' \subset T} \hat{s}(T')$.

Пусть задано некоторое множество Q и функции

$$\oplus : Q \times Q \to Q \text{ и } \otimes : Q \times Q \to Q.$$

Функция $D : \tilde{s} \to Q$ будет называться полиномиальной характеристикой множества, если:

1. Для любого $T' \subset T$ и любых $\sigma_1 \in \hat{s}(T')$ и $\sigma_2 \in \hat{s}(T')$, таких, что $\sigma_1 \cap \sigma_2 = \varnothing$, выполняется равенство $D(\sigma_1 \cup \sigma_2) = D(\sigma_1) \oplus D(\sigma_2)$.

2. Для любых $T_1 \subset T$ и $T_2 \subset T$, таких, что $T_1 \cap T_2 = \varnothing$, и любых $\sigma_1 \in \hat{s}(T_1)$ и $\sigma_2 \in \hat{s}(T_2)$ выполняется равенство $D(\sigma_1 \times \sigma_2) = D(\sigma_1) \otimes D(\sigma_2)$. Здесь под $\sigma_1 \times \sigma_2$ понимается множество функций вида $T_1 \cup T_2 \to S$, сужение которых на T_1 принадлежит σ_1, а сужение на T_2 принадлежит σ_2.

Для того чтобы однозначно задать полиномиальную характеристику, необходимо задать множество Q, задать значение функции D на множестве $\bigcup\limits_{t \in T} S^{\{t\}}$, т. е. определить $|S| \times |T|$ элементов из Q, не обязательно разных, иными словами, определить функцию $f : S \times T \to Q$, определить значение функции $D(\varnothing)$ для пустого множества, значение функции $D(\{\#\})$ для множества, содержащего лишь пустую функцию, и $2|Q|^2$ элементов из Q для задания функций $\oplus$ и $\otimes$.

В дальнейшем мы будем использовать обозначения $\overset{0}{\sum}$ и $\overset{0}{\prod}$ в контекстах $\overset{0}{\sum\limits_{i \in I}} q_i$ и $\overset{0}{\prod\limits_{i \in I}} q_i$, где $q_i \in Q$, $i \in I$, I — некоторое множество индексов. Определение этих символов дано следующими четырьмя равенствами:

$$\sum_{i\in\varnothing}^{0} q_i = D(\varnothing);$$

$$\sum_{i\in I\cup\{i^*\}}^{0} q_i = q_{i^*} \oplus \sum_{i\in I}^{0} q_i;$$

$$\prod_{i=\varnothing}^{0} q_i = D(\{\#\});$$

$$\prod_{i\in I\cup\{i^*\}}^{0} q_i = q_{i^*} \otimes \prod_{i\in I}^{0} q_i$$

для любых $i^* \notin I$.

Смысл функции $f: S \times T \to Q$, необходимой для задания полиномиальной характеристики, заключается в том, что для любой функции $v: T' \to S$ справедливо равенство

$$D(\{v\}) = \prod_{t\in T'}^{0} f(v(t), t).$$

Если $\hat{v}$ — некоторое подмножество функций вида $T' \to S$, то

$$D(\hat{v}) = \sum_{v\in\hat{v}}^{0} \prod_{t\in T'}^{0} f(v(t), t).$$

Задача, которую мы будем называть общей двумерно-грамматической задачей, заключается в том, чтобы для заданной двумерной грамматики G и заданной полиномиальной характеристики вычислить значение этой характеристики для множества вариантов, допустимых в этой грамматике.

Примеры полиномиальных характеристик. 1. Пусть на некотором подмножестве $Tv \subset T$ задана функция $v: Tv \to S$, т. е. задано изображение v. Для этого изображения определим характеристику D следующим образом.

Множество Q определим как множество $\{0, 1\}$, содержащее два элемента. В качестве операций $\oplus$ и $\otimes$ выберем операции $\bigvee$ и $\bigwedge$. Функцию $f: S \times T \to \{0, 1\}$ определим как

$$f(s, t) = 1, \text{ для всех } s \in S \text{ и } t \in T \setminus Tv;$$

$$f(s, t) = 1, \text{ для всех } t \in Tv \text{ и } s = v(t);$$

$$f(s, t) = 0, \text{ для всех } t \in Tv \text{ и } s \neq v(t).$$

Кроме того, будем считать, что $D(\varnothing) = 0$, а $D(\{\#\}) = 1$.

Для таким образом определенной полиномиальной характеристики ее значение на $S(G)$ равно истинности отношения $v \in \mathcal{L}(G)$.

2. Если в предыдущем примере считать $Tv = \varnothing$, то $D(S(G))$ означает противоречивость грамматики G.

3. Если Q есть множество чисел, f есть функция $S \times T \to R$, $D(\varnothing) = -\infty$, $D(\{\#\}) = 0$, $\oplus$ есть max, а $\otimes$ есть «+» в обычном смысле этого обозначения, то вычисление $D(S(G))$ есть определение качества оптимального варианта в приведенной ранее формулировке.

4. В отдельных приложениях качество варианта уместно определять не как сумму качеств составляющих его частей, а как качество наихудшей его части. В этом случае операцией $\otimes$ следует считать min, $D(\{\#\}) = \infty$, а все остальное, как в предыдущем примере.

5. Пусть Q есть множество целых чисел, операции $\oplus$ и $\otimes$ являются операциями «+» и «×» в обычном смысле этих обозначений, $D(\varnothing) = 0$, а $D(\{\#\}) = 1$. В таком случае $D(S(G))$ означает количество допустимых вариантов в грамматике G. Нетрудно сообразить, как следует изменить эту полиномиальную характеристику, чтобы $D(S(G))$ означало количество допустимых вариантов, содержащих заданный фрагмент.

6. Зачастую требуется указание качества не только наилучшего варианта, а еще и качества n наилучших вариантов. Это происходит, например, когда помимо оптимальности варианта требуется, чтобы он достаточно отличался по качеству от всех остальных вариантов. В этом случае при заданной функции $\varphi: S \times T \to R$ множество Q есть множество совокупностей из n чисел. Функция $f: S \times T \to Q$ определяется так, что

$$f(s, t) = (\varphi(s, t), \underbrace{-\infty, -\infty, \ldots, -\infty}_{n-1 \text{ раз}}),$$

$$D(\varnothing) = (\underbrace{-\infty, -\infty, \ldots, -\infty}_{n \text{ раз}}),$$

$$D(\{\#\}) = (0, \underbrace{-\infty, -\infty, \ldots, -\infty}_{n-1 \text{ раз}}).$$

Под операцией $(a_1 a_2, \ldots, a_n) \oplus (b_1, b_2, \ldots, b_n)$ должен пониматься выбор n наибольших чисел из $2n$ чисел $(a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n)$, а под операцией $(a_1, a_2, \ldots, a_n) \otimes (b_1, b_2, \ldots, b_n)$ — выбор n наибольших чисел из n^2 чисел $a_i + b_j$, $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant n$. Описание примеров закончено.

Для решения задачи вычисления полиномиальной характеристики языка, порождаемого двумерной грамматикой, выполним дополнительные построения.

Пусть $\Gamma_n = (T_n, \mathcal{T}_n)$ — граф, имеющий n вершин. Пронумеруем все вершины этого графа и получим последовательность $t_1, t_2, \ldots, t_{n-1}, t_n$, где $t_i \in T_n$. Через $\Gamma_i = (T_i, \mathcal{T}_i)$, $i = 0, 1, \ldots, n-1$, обозначим граф, получаемый из графа Γ_{i+1} в результате k-исключения вершины t_{i+1}, а через T_i^* — множество вершин в графе Γ_i, соединенных ребром с вершиной t_i. Везде в данном параграфе нижний индекс при t будет означать именно номер вершины в принятой нумерации. Для различных целых i и j множества T_i^* и T_j^* не обязательно различные. Обозначим через $\mathcal{T}^*$ множество попарно различных множеств из последовательности $T_1^*, T_2^*, \ldots, T_n^*$. Понятно, что $T_n \subset \bigcup\limits_{T^* \in \mathcal{T}^*} T^*$, однако $T_n \neq \bigcup\limits_{T^* \in \mathcal{T}^*} T^*$. Например, вершина t_n в последовательности $t_1, t_2, \ldots, t_n$ не входит ни в какое множество $T^* \in \mathcal{T}^*$. Возможны и другие такие вершины. Рассмотрим множество $\{\{t\} : t \in T_n \setminus \bigcup\limits_{T^* \in \mathcal{T}} T^*\}$ и включим его в множество $\mathcal{T}^*$. Таким образом, $\mathcal{T}^{*\prime} = \mathcal{T}^* \cup \{\{t\} : t \in T_n \setminus \bigcup\limits_{T^* \in \mathcal{T}^*} T^*\}$. Именно это множество в дальнейшем будет обозначено через $\mathcal{T}^*$. Определим функцию $R : T_n \to \mathcal{T}^*$ так, что $R(t_i) = T_i^*$, $i = 1, 2, \ldots, n$, и отношение $\to \subset T_n \times T_n$, такое, что $t' \to t''$, если $t' \in R(t'')$. Транзитивное замыкание отношения $\to$ обозначим через $\Rightarrow$. Расширим отношение $\Rightarrow$ еще на множество $\mathcal{T}^* \times T_n$ и запишем, что $T^* \Rightarrow t$, если существует $t' \in T_n$, такое, что $t' \Rightarrow t$ и $T^* = R(t')$. Аналогично запишем $T^* \to t$, если $R(t) = T^*$. Приведенные определения отношений $\to$ и $\Rightarrow$ вполне однозначны, однако для исключения возможного ложного их понимания отметим, что если $T^* \Rightarrow t$, $T^* \neq \varnothing$, то существует вершина $t' \in T^*$, такая, что $t' \Rightarrow t$. Более того, для любой вершины $t' \in T^*$ справедливо отношение $t' \Rightarrow t$. В то же время если для некоторого t существует вершина t', такая, что $t' \Rightarrow t$ и $t' \in T^*$, то не обязательно $T^* \Rightarrow t$.

Введем четыре обозначения:

$$T_{\text{сл}}(t) = \{t' : t' \in T_n,\ t \Rightarrow t'\} \cup \{t\}, \quad t \in T_n,$$

$$T_{\text{сл}}(T^*) = \{t : t \in T_n,\ T^* \Rightarrow t\}, \quad T^* \in \mathcal{T}^*,$$

$$\hat{\mathcal{T}}(t) = \{T^* : T^* \in \mathcal{T}^*,\ T^* \subset (R(t) \cup \{t\}),\ t \in T^*\}, \quad t \in T_n,$$

$$\hat{T}(T^*) = \{t : t \in T_n,\ T^* \to t\}, \quad T^* \in \mathcal{T}^*.$$

Индекс сл в обозначении $T_{\text{сл}}$ взят как начальные буквы слова «следующий».

Из приведенных определений достаточно ясно, что для любого $T^* \in \mathcal{T}^*$ справедливо равенство

$$T_{\text{сл}}(T^*) = \bigcup_{t \in \hat{T}(T^*)} T_{\text{сл}}(t), \tag{3.48}$$

а для любого $t \in T_n$ равенство

$$T_{\text{сл}}(t) = \Big(\bigcup_{T^* \in \hat{\mathcal{T}}(t)} T_{\text{сл}}(T^*)\Big) \cup \{t\}. \tag{3.49}$$

Менее очевидно, но важно для последующего изложения, что множества-операнды в правых частях (3.48) и (3.49) попарно не пересекаются, что доказывается в следующих двух леммах.

Лемма 2. *Для любых двух вершин t_i и t_j, таких, что $R(t_i) = R(t_j)$, а $t_i \neq t_j$, множества $T_{\text{сл}}(t_i)$ и $T_{\text{сл}}(t_j)$ не пересекаются.*

Доказательство. 1. Докажем, что не существует такого $t^* \subset T_n$, при котором $t^* \Rightarrow t^*$. Допустим, это неверно, т. е. существует последовательность вершин $t^*, t', t^2, \ldots, t^l, t^*$, для которой справедлива цепочка $t^* \to t^1 \to t^2 \to \ldots \to t^l \to t^*$. В данной цепочке номер вершины t^1 строго больше номера вершины t^*, т. е. t^1 равно некоторому t_{i_1} в последовательности $t_1, t_2, \ldots, t_l$, а t^* равно некоторому t_{i^*} в этой последовательности, причем $i_1 > i^*$. Аналогично номер любой вершины t^j больше номера вершины t^{j-1} и, наконец, номер вершины t^* больше номера вершины t^l. Таким образом, номер вершины t^* больше номера вершины t^*, что невозможно, так как любая вершина имеет единственный номер. Поэтому для любого $t^* \in T_n$ невозможно отношение $t^* \Rightarrow t^*$.

2. Докажем, что если для некоторых трех разных вершин t_i, t_j, t_l выполняются условия $t_i \to t_l$ и $t_j \to t_l$, то либо $t_i \to t_j$, либо $t_j \to t_i$.

Запись $t_i \to t_l$ и $t_j \to t_l$ означает, что в момент k-исключения вершины t_l из графа Γ_l эта вершина была соединена ребром как с t_i, так и с t_j. Следовательно, после k-исключения вершины t_l вершины t_i и t_j обязательно оказались соединенными ребром. Значит, если $i > j$, то $t_j \in R(t_i)$ и, следовательно, $t_j \to t_i$. Если же $i < j$, то $t_i \in R(t_j)$ и $t_i \to t_j$.

3. Докажем, что из двух цепочек отношений

$$t^{11} \to t^{12} \to t^{13} \to \cdots \to t^{1,n_1-1} \to t^{1n_1}, \tag{3.50}$$

$$t^{21} \to t^{22} \to t^{23} \to \cdots \to t^{2,n_2-1} \to t^{2n_2}, \tag{3.51}$$

таких, что $t^{11} \neq t^{21}$, а $t^{1n_1} = t^{2n_2}$, следует справедливость либо цепочки

$$t^{11} \to t^{12} \to \cdots \to t^{1,n_1-1} \to t^{2,n_2-1}, \tag{3.52}$$

либо цепочки

$$t^{21} \to t^{22} \to \cdots \to t^{2n_2-1} \to t^{1,n_1-1}. \tag{3.53}$$

Из последних отношений в цепочках (3.50) и (3.51), равенства $t^{1,n_1} = t^{2,n_2}$ и из доказанного в п. 2 утверждения следует, что либо $t^{1,n_1-1} \to t^{2,n_2-1}$, либо $t^{2,n_2-1} \to t^{1,n_1-1}$. В первом случае оказывается справедливым (3.52), а во втором — (3.53).

4. Последовательно применяя доказанное в предыдущем параграфе, получаем, что если $t_i \Rightarrow t_l$ и $t_j \Rightarrow t_l$, то существует такое t_s, $s < l$, при котором либо $t_i \to t_s$ и $t_j \Rightarrow t_s$, либо $t_i \Rightarrow t_s$ и $t_j \to t_s$.

5. Аналогично доказанному в п. 3. можно доказать, что если три разные вершины t_i, t_j, t_l таковы, что $t_i \to t_l$ и $t_j \Rightarrow t_l$, то либо $t_j \Rightarrow t_i$, либо существует такое t_s, $s < l$, при котором $t_i \to t_s$ и $t_j \Rightarrow t_s$.

6. С учетом доказанного в п. 4 и последовательно применяя доказанное в п. 5, получаем, что если для некоторых трех разных клеток t_i, t_j и t_l справедливо $t_i \Rightarrow t_l$ и $t_j \Rightarrow t_l$, то либо $t_i \Rightarrow t_j$, либо $t_j \Rightarrow t_i$.

7. Аналогично доказанному в п. 1, можно доказать, что если справедливо отношение $T^* \Rightarrow t$, то номер вершины t в последовательности $t_1, t_2, \ldots, t_n$ больше номера любой вершины из T^*. Поэтому для любых $T^* \in \mathcal{T}^*$ и $t \in T^*$ невозможно отношение $T^* \Rightarrow t$.

8. Допустим, что лемма неверна, т. е. что существует такое $t_l \in T_n$ при котором $t_i \Rightarrow t_l$ и $t_j \Rightarrow t_l$. В силу доказанного в п. 6 следует, что либо $t_i \Rightarrow t_j$, либо $t_j \Rightarrow t_i$. Рассмотрим лишь одну из этих ситуаций, т. е. когда существуют такие вершины $t^1, t^2, \ldots, t^P$, что справедлива цепочка $t_i \to t^1 \to t^2 \to \ldots \to t^P \to t_j$. Из этой цепочки следует $R(t_i) \Rightarrow t^P$, т. е. в силу доказанного в п. 7 $t^P \notin R(t_i)$. Последнее отношение $t^P \to t_j$ в цепочке означает, что $t^P \in R(t_j)$. Отсюда следует, что $R(t_i) \neq R(t_j)$, а это противоречит условию леммы. Лемма доказана.

Лемма 3. *Пусть t^0 — любая вершина в графе Γ_n, а $T^1 \in \mathcal{T}^*$ и $T^2 \in \mathcal{T}^*$ — два различных множества, таких, что $t^0 \in T^1$, $t^0 \in T^2$, $T^1 \subset R(t^0) \cup \{t^0\}$ и $T^2 \subset R(t^0) \cup \{t^0\}$. В таком случае множества $T_{сл}(T^1)$ и $T_{сл}(T^2)$ не пересекаются.*

Доказательство. Отметим, что множества $\{t : t \in T_n, R(t) = T^1\}$ и $\{t : t \in T_n, R(t) = T^2\}$ не пересекаются. Поэтому если лемма неверна, то существует такие три различные вершины t_i, t_j и t_l, что $T^1 \to t_i \Rightarrow t_l$ и $T^2 \to t_j \Rightarrow t_l$. Из этого утверждения и утверждения, доказанного в п. 6 леммы 2, следует, что либо $T^1 \to t_i \Rightarrow t_j \leftarrow T^2$, либо $T^1 \to t_i \Leftarrow t_j \leftarrow T^2$. Рассмотрим только первый случай, из которого вытекает существование последовательности вершин $t^1, t^2, \ldots, t^P$, для которой справедлива цепочка

$$t_i \to t^1 \to t^2 \to \cdots \to t^P \to t_j. \tag{3.54}$$

Из чего следует, что номер вершины t^P в последовательности $t_1, t_2, \ldots, t_n$ больше номера i вершины t_i. Тогда справедливо равенство $t^0 \in T^1 = R(t_i)$, и поэтому номер вершины t^0 меньше i, кроме того, номер любой вершины из $R(t^0)$ меньше номера вершины t^0. Из этого, в свою очередь, следует, что номер вершины t^P больше номера любой вершины из $R(t^0)$, а следовательно,

$$t^P \notin R(t^0) \cup \{t^0\}. \tag{3.55}$$

Из отношения $t^P \to t_j$ следует, что $t^P \in T^2 = R(t_j)$. Отношение $t^P \in T^2$ совместно с (3.55), немедленно приводит к отношению $T^2 \not\subset R(t^0) \cup \{t^0\}$, что противоречит условию леммы. Лемма доказана.

Пусть задана грамматика $G = \langle T, S, \mathcal{T}, \Omega \rangle$. Будем считать, что клетки поля T пронумерованы так, что они образуют последовательность $t_1, t_2, \ldots, t_n$, которая определяет множества $\mathcal{T}^*$, $T_{сл}(\mathfrak{X})$, $\mathfrak{X} \in T \cup \mathcal{T}^*$, $\hat{\mathcal{T}}(t)$, $t \in T$, $\hat{T}(T^*)$, $T^* \in \mathcal{T}^*$. Существенным является то, что для грамматик k-й степени $|T^*| \leqslant k$ для любого $T^* \in \mathcal{T}^*$. Алгоритм нахождения последовательности $t_1, t_2, \ldots, t_n$ указан в предыдущем параграфе, равно как и алгоритм нахождения последовательности $t_1, t_2, \ldots, t_n$, обеспечивающей минимальное значение k, такое, что $|T^*| \leqslant k$, если степень грамматики неизвестна.

Для любого подмножества $T' \subset T$, не обязательно принадлежащего $\mathcal{T}$ или $\mathcal{T}^*$, фрагмент $s(T') : T' \to S$ будем считать допустимым, если для любого $T'' \in \mathcal{T}$ и $T'' \subset T'$ справедливо равенство $F(T'', s(T')) = 1$.

Определим следующие множества.

Для любого $T^* \in \mathcal{T}^*$ и любого фрагмента $s(T^*) : T^* \to S$ через $S_1(T^*, s(T^*))$ обозначим множество фрагментов $s(T_{сл}(T^*)) : T_{сл}(T^*) \to S$, таких, что $\{s(T^*)\} \times S_1(T^*, s(T^*))$ есть множество допустимых фрагментов на $T^* \cup T_{сл}(T^*)$, сужение которых на T^* есть $s(T^*)$.

Для любого $T^* \in \mathcal{T}^*$ и $t \in T$, такого, что $T^* \to t$, и фрагмента $s(T^*)$ через $S_2(T^*, t, s(T^*))$ обозначим множество фрагментов $s(T_{сл}(t)) : T_{сл}(t) \to S$, таких, что $\{s(T^*)\} \times S_2(T^*, t, s(T^*))\}$ есть множество допустимых на $T^* \cup T_{сл}(t)$ фрагментов, сужение которых на T^* есть $s(T^*)$.

Для любого $T^* \in \mathcal{T}^*$, $t \in T$, такого, что $T^* \to t$, и фрагмента $s(T^* \cup \{t\})$ через $S_3(T^*, t, s(T^* \cup \{t\}))$ обозначим множество фрагментов $s(T_{сл}(t)) : T_{сл}(t) \to S$, таких, что $\{s(T^*)\} \times S_3(T^*, t, s(T^* \cup \{t\}))\}$ есть множество допустимых на $T^* \cup T_{сл}(t)$ фрагментов, сужение которых на $T^* \cup \cup \{t\}$ есть $s(T^* \cup \{t\})$.

Для любых $T^* \in \mathscr{T}^*$, $t \in T$, таких, что $T^* \to t$, и любого фрагмента $s(T^*)$ обозначим через $S_G(T^*, t, s(T^*))$ подмножество значений структурных элементов $s(t)$ в клетке t, таких, что $\{s(T^*)\} \times S_G(T^*, t, s(T^*))$ есть множество допустимых фрагментов на $T^* \cup \{t^*\}$, сужением которых на T^* является $s(T^*)$.

Справедливы следующие три равенства.

Для любого $T^* \in \mathscr{T}^*$ и $s(T^*)$

$$S_1(T^*, s(T^*)) = \underset{t \in \hat{T}(T^*)}{\mathrm{X}} S_2(T^*, t, s(T^*)). \tag{3.56}$$

Для любых $T^* \in \mathscr{T}^*$, $t \in T$, таких, что $T^* \to t$, и любого $s(T^*)$

$$S_2(T^*, t, s(T^*)) = \bigcup_{s(t) \in S_G(T^*, t, s(T^*))} S_3(T^*, t, s(T^* \cup \{t\})). \tag{3.57}$$

Для любых $T^* \in \mathscr{T}$, $t \in T$, таких, что $T^* \to t$, и любого $S(T^* \cup \{t\})$

$$S_3(T^*, t, s(T^* \cup \{t\})) = (\underset{T' \in \hat{\mathscr{T}}(t)}{\mathrm{X}} S_1(T', s(T'))) \times \{s(\{t\})\}. \tag{3.58}$$

Из леммы 2 следует, что множества-сомножители в правой части (3.56) являются множествами вариантов, заданных на непересекающихся подмножествах поля T. Таким же свойством в силу леммы 3 обладают множества-сомножители в правой части (3.58). Множества под знаком объединения в правой части (3.57) попарно не пересекаются. Поэтому для любой полиномиальной характеристики D справедливы равенства

$$D(S_1(T^*, s(T^*)) = \prod_{t \in \hat{T}(T^*)}^{0} D(S_2(T^*, t, s(T^*))), \tag{3.59}$$

$$D(S_2(T^*, t, s(T^*))) = \sum_{s(t) \in S_G(T^*, t, s(T^*))}^{0} D(S_3(T^*, t, s(T^* \cup \{t\}))), \tag{3.60}$$

$$D(S_3(T^*, t, s(T^* \cup \{t\}))) = \left(\prod_{T' \in \hat{\mathscr{T}}(t)}^{0} S_1(T', s(T'))\right) \otimes f(s(t), t). \tag{3.61}$$

Поскольку $T_{сл}(\varnothing)$ есть T, то $D(S_1(\varnothing, \#))$ есть значение этой характеристики для множества вариантов, допустимых в заданной грамматике. Покажем, что формулы (3.59), (3.60), (3.61) могут быть расчетными формулами для вычисления этого значения.

Множество T с заданным на нем отношением $\to$ определяет ориентированный граф, не имеющий циклов. Следовательно, в этом графе и любом его подмножестве имеется по крайней мере одна вершина, не имеющая выходящей стрелки. Вычисления будут выполняться по шагам таким образом, что к началу каждого шага множество T вершин разбито на два подмножества T^+ и T^-. Если какая-то вершина t' принадлежит T^+, то это означает, что уже вычислены все величины $D(S_2(T^*, t, s(T^*)))$ и $D_3(S_3(T^*, t, s(T^* \cup \{t\})))$, в которых $t = t'$. Выполнение очередного шага заключается в том, что выбирается любая вершина $t^* \in T^-$, не имеющая выходящих стрелок, или такая, что все выходящие стрелки входят в вершины из T^+. Затем выполняются такие вычисления, что после них эта вершина уже может быть отнесена к T^+. Покажем, что это возможно.

Для выбранной вершины t^* любое множество $T' \in \hat{\mathscr{T}}(t^*)$ и любая клетка $t' \in \hat{T}(T')$ обладают тем свойством, что $t^* \to \to t'$. Согласно правилу, по которому была выбрана вершина t^*, для любой такой вершины t' уже вычислены значения $D(S_2(R(t'), t, s(R(t'))))$ и $D(S_3(R(t'), t', s(R(t') \cup \{t'\})))$.

Таким образом, уже имеются данные для вычисления $D(S_1(T', s(T')))$ по формуле (3.59) для всех $T' \in \hat{\mathscr{T}}(t^*)$ и всех $s(T')$. После этих вычислений имеются все данные для вычисления $D(S_3(R(t^*), t^*, s(R(t^*) \cup \{t^*\})))$ по формуле (3.61) и для вычисления $D(S_2(R(T^*), t^*, s(R(t^*))))$ по формуле (3.60). Поскольку значения $D(S_2(T^*, t, s(T^*)))$ и $D(S_3(T^*, t, s(T^* \cup \{t\})))$ должны быть вычислены лишь для тех пар T^* и t, для которых $T^* = R(t)$, указанными вычислениями исчерпываются действия на этом шаге, и вершина t^* может быть переведена из множества T^- в множество T^+.

Определим количество операций, которые должны быть выполнены для вычисления $D(S_1(\varnothing, \#))$. Формула (3.60) должна выполняться для всех $t \in T$, $T^* = R(t)$, и любого $s(T^*)$, следовательно, не более чем $|S|^k \cdot |T|$ раз. Вычисление каждой формулы требует не более чем $|S| - 1$ операций $\oplus$. Следовательно, общее количество выполнений операции $\oplus$ не превышает $|S|^{k+1} \cdot |T| - |S|^k \in |T|$.

Каждое из вычисленных значений $D(S_2(T^*, t, s(T^*)))$ используется в формуле (3.59) не более одного раза. Количество этих значений не превышает $|S|^k \cdot |T|$. Следовательно, количество операций $\otimes$, необходимых для всех вычислений по формуле (3.59), не превышает $|S^k| \cdot |T| - N$, где N — количество формул вида (3.59), т. е. количество величин $D(S_1(T^*, s(T^*)))$. Для реализации всех вычислений по формуле (3.61) требуется выполнить ровно N операций $\otimes$, т. е. столько операций $\otimes$, сколько различных величин $D(S_1(T^*, s(T^*)))$, так как каждая из этих величин используется в формуле (3.61) один раз. Таким образом, для всех необходимых вычислений требуется выполнить операцию $\otimes$ не более чем $|S|^k \cdot |T|$ раз. Является, таким образом, справедливой следующая теорема.

Теорема 13. *Для любой двумерной грамматики* $\langle T, S, \mathcal{T}, \Omega\rangle$ *k-й степени вычисление полиномиальной характеристики множества допустимых вариантов требует выполнения не более чем* $|S|^k \cdot |T|$ *операций* $\otimes$ *и не более чем* $|S|^k \times (|S| - 1) \cdot T$ *операций* $\oplus$.

3.5. ЗАКЛЮЧИТЕЛЬНЫЕ ЗАМЕЧАНИЯ

В заключение данной главы укажем, в чем состоит рекомендательная сила полученных результатов, особенно описанных в последних параграфах. Что касается алгоритма вычеркивания и метода кажущихся качеств, то это вполне конкретные алгоритмы, которые могут применяться и практически применяются [53, 56, 64]. Что же касается результатов, приведенных в последних двух параграфах, то оценка автором их полезности имеет несколько иной характер. Важным здесь является то, что описанный общий алгоритм позволяет по-новому оценить те трудности, которые до сих пор считались присущими распознаванию изображений.

Первой бросающейся в глаза трудностью представлялась зашумленность изображения случайными помехами. Преодолению этой трудности посвящено немалое количество работ, основанных на построении и исследовании вероятностных моделей изображения и применения или модификации статистических методов для оценки незашумленного изображения. Из сформулированной общей двумерно-грамматической задачи и описанного алгоритма ее решения хотелось бы сделать вывод о том, что значение помех на изображении до сих пор было сильно преувеличенным. Мы видим из общей формулировки задачи, что, конечно, существует различие между распознаванием идеальных изображений и оптимизационной задачей, к которой сводится распознавание изображений в условиях помех. Однако это различие имеет тот же характер, что и различие логических и арифметических операций на ЭВМ. Как для распознавания идеальных изображений, так и для распознавания зашумленных изображений требуется выполнять одинаковое количество операций, более того, выполнять по одной и той же программе. Только в одном случае эта программа должна работать с булевскими переменными, а в другом — с вещественными. Такое различие, конечно, не столь принципиально, как представлялось ранее, когда предполагалось, что для решения этих двух задач требуются принципиально разные подходы, скажем, структурно-лингвистический — для распознавания идеальных изображений и вероятностный — для распознавания изображений в условиях помех.

По-видимому, значение зашумленности изображения до сих пор было сильно преувеличено. Это, однако, вовсе не означает, что задача распознавания зашумленных изображений оказалась более легкой, чем предполагалось. Просто она оказалась не намного труднее, чем задача распознавания идеальных изображений, в которой обнаружились ранее не замеченные трудности фундаментального характера.

Второй бросающейся в глаза трудностью в распознавании изображений представлялся огромный объем множества T, т. е. большое количество признаков, которое, как правило, составляет около 10^6 и больше. Эта трудность, как следует из сформулированного общего алгоритма, вообще кажущаяся. А именно: мы показали, что при любой степени структуры двумерной грамматики трудоемкость вычисления любой полиномиальной характеристики растет линейно с ростом T, т. е. растет минимально возможным способом. В этом смысле объем T, т. е. количество признаков, вообще не определяет возможность или невозможность практического решения задачи. Решающим фактором здесь является не количество признаков, а структура множества признаков, точнее, ее характеристика, названная степенью. Понимание этого факта позволяет по-новому оценить проблемы минимизации признаков, решению которой или подходам к решению которой посвящено очень большое количество работ. Эта проблема представляет самостоятельный интерес, но ни в коей мере не является средством уменьшения вычислительных затрат при последующем распознавании. Требуется минимизировать не количество признаков, а степень структуры, определяющей зависимость этих признаков. Таким образом, роль минимизации количества признаков как средства уменьшения количества вычислений при последующем распознавании должна оцениваться сейчас значительно более сдержанно, чем до сих пор.

В то же время, несомненно, полезны те эвристические приемы, которые применяются сейчас (хотя и без ясного понимания того, что производится), фактически приводящие к представлению распознаваемого объекта в структуре малой степени. Крайним проявлением стремления уменьшить степень структуры является стремление представить распознаваемый объект совокупностью попарно независимых признаков, т. е. представить его в структуре нулевой степени. Выделение независимых подмножеств признаков, представление изображения контурами или так называемыми скелетными изображениями — все эти до сих пор разрозненные эвристические приемы могут сейчас интерпретироваться как поиск структуры именно с малой степенью, в которой возможно адекватное описание объекта распознавания. Формальное определение степени структуры как важнейшего параметра множества признаков и доказательство распознаваемости последнего позволяют дать единую формализацию разрозненных приемов в виде постановки задачи поиска структуры с минимальной степенью взамен задачи поиска минимального количества признаков.

Естественно, для представления изображения в той или иной структуре необходимо иметь соответствующее аппаратурное или программное обеспечение, каковым является описанный в приложении комплекс.

Глава 4

ВЕРОЯТНОСТНЫЕ ДВУМЕРНЫЕ ГРАММАТИКИ

Проблема распознавания образов настолько многогранна, что ее невозможно охватить единым взглядом. Взгляд на распознавание изображений с точки зрения двумерных грамматик позволяет рассмотреть лишь одну из многочисленных ее граней, и, конечно, очень много других важных свойств изображений остаются скрытыми при таком рассмотрении. Важно по крайней мере то, что в предыдущих трех главах обнаружились новые, ранее неизвестные свойства изображений и множеств изображений, которые не обнаруживались в рамках других известных подходов. Так, мы намеренно игнорировали те свойства изображений, которые исследуются в статистическом подходе к распознаванию, чтобы показать, что задачи распознавания оказываются достаточно нетривиальными даже в том случае, если придерживаться сугубо детерминированного подхода.

Проблема распознавания изображений исследовалась нами в отрыве и от теории обучения распознаванию, в результате чего, хотелось бы надеяться, вскрыты ранее не замеченные пласты проблемы, которые содержат вопросы собственно распознавания, лежащие вне проблемы обучения распознаванию.

В данной главе мы покажем, что такое изолированное рассмотрение не является обязательным, что взгляд на проблему распознавания с точки зрения двумерных грамматик дает не альтернативу, а дополнение к известным подходам. Мы покажем, что задание множеств изображений двумерно-грамматическими средствами никак не препятствует использованию основных рекомендаций статистической теории распознавания и обучения. Реализация этих рекомендаций потребовала, однако, выполнения некоторых дополнительных исследований, выходящих за рамки двумерных грамматик. Так, читатель, возможно, заметит, что взаимосвязь задач обучения I и обучения II, сформулированная в теоремах 3—6, справедлива не только в рамках двумерно-грамматического подхода. Что касается трех теорем о самообучении, то они в самом тексте сформулированы как достаточно общие.

4.1. ФОРМУЛИРОВКА ОСНОВНЫХ ЗАДАЧ

Основные задачи статистической теории распознавания образов основаны на следующих понятиях.

Задано множество S, называемое множеством образов, и функция $P: S \to R$, такая, что $\forall s \in S\ (P(s) \geqslant 0)$ и $\sum_{s \in S} P(s) = 1$. Функция P называется априорным распределением вероятностей образов.

Задано множество X, называемое множеством наблюдений, и функция $PX: S \times X \to R$, такая, что $\forall s \in S\ (\forall x \in X\ (PX(s, x) \geqslant 0))$ и $\forall s \in S \left(\sum_{x \in X} PX(s, x) = 1\right)$.

Функция PX называется условным распределением вероятностей наблюдений x при фиксированном образе s.

Задано множество D, называемое можеством решений, и функция $W: D \times S \to R$, называемая функцией штрафов. Задача заключается в том, чтобы при заданных функциях P, PX, W определить функцию $Q: X \to D$, минимизирующую функционал

$$\sum_{x \in X} \sum_{s \in S} PX(s, x) \cdot P(s) \cdot W(Q(x), s).$$

Общее решение этой задачи, называемой байесовской задачей принятия решений, заключается, очевидно, в том, что для любого $x \in X$

$$Q(x) = \arg\min_{d \in D} \sum_{s \in S} PX(s, x) \cdot P(s) \cdot W(d, s). \tag{4.1}$$

Обычно при исследованиях этой задачи молчаливо допускается, что $|S|$ и $|D|$ малы, а функция PX такова, что вычисление ее значения для любых $x \in X$ и $s \in S$ не составляет особого труда. В этом случае не возникает вопросов о способах задания функций P и W, так как первая из них полностью определяется $|s| - 1$ числами, а вторая — $|D| \cdot |S|$ числами. Не возникает также никаких проблем с реализацией формулы (4.1), так как ее вычисление для заданного значения x требует лишь $|D| \cdot |S|$ несложных вычислений. В силу этого внимание исследователей в большей степени было направлено не на реализацию формулы (4.1), а на исследование круга проблем, известных под названием обучения.

Мы хотим решать байесовскую задачу принятия решений для случая, когда $S = D$, где S — множество вариантов, допустимых в данной двумерной грамматике, т. е. для случая, когда $|S|$ чрезвычайно велико. В этой ситуации рекомендация (4.1) чересчур общая и непосредственно не указывает прагматически хорошего алгоритма для ее реализации. Для того чтобы воспользоваться рекомендацией (4.1), необходимо прежде всего разумно ограничить класс функций P, PX, W так, чтобы задание функций из этих классов могло осуществляться определением небольшого количества чисел, а не $|S|$ чисел, так как $|S|$ чрезвычайно велико. Кроме того, необходимо располагать прагматически хорошим алгоритмом для решения оптимизационной задачи (4.1). Это значит, что мы сталкиваемся с вопросами, которые в общей статистической теории распознавания просто опускались, как тривиальные, и которые действительно являются таковыми, если считать, что $|S|$ есть небольшое число, например 2.

Обозначим через PXP функцию $S \times X \to R$, такую, что $\forall x \in X\ (\forall s \in S\ (PXP(x, s) = PX(s, x) \cdot P(s)))$. Понятно, что для вычислений по формуле (4.1) нет необходимости знать одновременно и функцию P, и функцию PX, а достаточно знать функцию PXP. Задачи обучения возникают, когда эта функция неизвестна, но известно множество $\mathcal{P}$, содержащее эту функцию. Известна также некоторая последовательность $\hat{X}$ наблюдений $x_1, x_2, \ldots, x_m$ и последовательность $\hat{S}$ вариантов $s_1, s_2, \ldots, s_m$. Эта пара последовательностей называется обучающей выборкой. Обучение заключается в том, чтобы по обучающей последовательности разумно выбрать функцию $PXP \in \mathcal{P}$. Наиболее известными являются следующие три постановки задачи обучения.

Задача обучения I. Если предположить, что в обучающей последовательности каждая из пар (x_i, s_i), $i = 1, 2, \ldots, m$, имеет вероятность $PXP(x, s)$, а пара x_i, s_i не зависит от пары (x_j, s_j) при $i \neq j$, то вероятность обучающей последовательности определяется величиной $\prod_{i=1}^{m} PXP(x_i, s_i)$. Задача обучения I заключается в выборе такой функции $P\widehat{X}P \in \mathcal{P}$, которая максимизирует эту вероятность, а именно:

$$P\widehat{X}P = \arg\max_{PXP \in \mathcal{P}} \prod_{i=1}^{m} PXP(x_i, s_i). \tag{4.2}$$

Задача обучения II. Зачастую на практике (особенно при конструировании читающих автоматов) настройку распознающей системы производят не по случайной, а по тщательно подобранной выборке изображений, достаточно хороших с точки зрения конструктора распознающей системы и отобранных конструктором в качестве тех изображений, которые уж наверняка должны признаваться распознающей системой как весьма вероятные. Задача обучения в этом случае заключается в выборе такой функции $P\widehat{X}P \in \mathcal{P}$, при которой вероятность каждой пары (x_i, s_i) из обучающей последовательности оказывается достаточно большой, т. е.

$$P\widehat{X}P = \arg\max_{PXP \in \mathcal{P}} \min_i PXP(x_i, s_i). \tag{4.3}$$

Задача обучения III. Пусть $D = S$, а функция W выбрана таким образом, что принимает только значения 0 и 1, причем $W(s_1, s_2) = 0$, если $s_1 = s_2$, и $W(s_1, s_2) = 1$, если $s_1 \neq s_2$. Тогда формула (4.1) приобретает вид $Q(x) = \arg\max_{s \in S} PXP(x, s)$, а функция PXP может рассматриваться уже не как распределение вероятностей, а как параметр решающего правила $Q(x)$. Задача обучения III заключается в выборе такой функции PXP из $\mathcal{P}$ чтобы выполнялась система неравенств

$$P\widehat{X}P(x_i, s_i) > P\widehat{X}P(x_i, s), \quad s \neq s_i, \quad i = 1, 2, \ldots, m,$$

и уместнее было бы ее называть задачей синтеза решающего правила, чем задачей обучения.

В данной главе мы определим и исследуем класс $\mathcal{P}$ распределений вероятностей на множестве вариантов, допустимых в заданной двумерной грамматике. Класс $\mathcal{P}$ будет определен так, что распределения из этого класса уместно будет назвать марковскими. Для класса марковских распределений мы укажем алгоритмы решения некоторых, а именно двух, байесовских задач. Затем покажем, что задачи обучения в известных, а именно в только что приведенных, трех формулировках могут ставиться и в рамках двумерных грамматик и решаться с помощью методов, которые несущественно отличаются от известных.

В случае когда обучающая последовательность состоит из одних только наблюдений $x_1, x_2, \ldots, x_m$ без указания для каждого x_i образа s_i, которому данное наблюдение соответствует, и требуется принять разумное решение о функции $PXP \in \mathcal{P}$, то речь идет о задачах самообучения распознаванию. В одном из последующих параграфов будет дана формулировка этой задачи и ее решение в общем виде. Как и для задач обучения, будет показано, что задача самообучения в предложенной постановке и ее общее решение могут быть реализованы в рамках двумерных грамматик.

4.2. ВЕРОЯТНОСТНАЯ МОДЕЛЬ МНОЖЕСТВА ДОПУСТИМЫХ ВАРИАНТОВ

Пусть $G = \langle T, S, \mathcal{T}, \Omega \rangle$ — двумерная грамматика. Для любого $T' \in T$, не обязательно из $\mathcal{T}$, через $\hat{s}(T')$ будем обозначать $S^{T'}$ — множество всех возможных функций вида $T' \to S$, имеющих область определения T', не обязательно допустимых в G.

Пусть R — множество вещественных чисел, $P : \hat{s}(T) \to R$ — распределение вероятностей вариантов, т. е. функция, такая, что $\forall s \in S^T \; (P(s) \geqslant 0)$ и $\sum_{s \in S^T} P(s) = 1$. Для любого распределения вероятностей P и любого $T' \subset T$ через $P(T')$ будем обозначать функцию $\hat{s}(T') \to R$, такую, что для всех $s(T') \in \hat{s}(T')$ выполняется равенство $P(T', s(T')) = \sum_{s(T \setminus T') \in \hat{s}(T \setminus T')} P(s)$. Очевидно, что $P(T')$ также есть распределение вероятностей на $\hat{s}(T')$, т. е. $\forall s(T') \in \hat{s}(T') \; (P(T', s(T')) \geqslant 0)$ и $\sum_{s(T') \in \hat{s}(T')} P(T')) = 1$.

Множество $\hat{s}(T')$ и функцию $P(T')$ будем считать определенной и для $T' = \varnothing$, так, что $\hat{s}(T')$ содержит единственный элемент $\#$, называемый пустым фрагментом, а вероятность $P(\varnothing, \#)$ равна единице.

Для любого распределения вероятностей P и для любых $T' \subset T$ и $T'' \subset T$, таких, что $T' \cap T'' = \varnothing$, через $P(T'/T'')$ будем обозначать функцию от $|T'| + |T''|$ переменных, имеющую областью определения множество тех фрагментов $s(T' \cup T'') : T' \cup T'' \to S$, для которых $P(T'', s(T'')) \neq \neq 0$, и такую, что для всех таких фрагментов выполняется равенство

$$P(T'/T'', s(T' \cup T'')) = \frac{P(T' \cup T'', s(T' \cup T''))}{P(T'', s(T''))}.$$

Очевидно, что $P(T'/T'')$ есть также распределение вероятностей на $\hat{s}(T')$, т. е.

$$\forall s(T'') \in \hat{s}(T'') ((P(T'', s(T'')) > 0) \Rightarrow$$

$$\Rightarrow \forall s(T') \in \hat{s}(T') (P(T'/T'', s(T' \cup T'')) \geqslant 0)),$$

$$\forall s(T'') \in \hat{s}(T'') \left[((P(T'', s(T'')) > 0) \Rightarrow \right.$$

$$\left. \Rightarrow \left(\sum_{s(T') \in \hat{s}(T')} P(T'/T'', s(T' \cup T'')) = 1 \right) \right].$$

В грамматике $G = \langle T, S, \mathcal{T}, \Omega\rangle$ для любого $T' \subset T$ будем считать определенным граничное подмножество $T'_{\text{гр}} = \{t : t \in T', \ \exists t' \notin T' (\{t, t'\} \in \mathcal{T})\}$. Индекс «гр» в обозначении $T'_{\text{гр}}$ следует понимать как оператор, переводящий множество T' в множество граничных его клеток. Таким образом, в дальнейшем будем считать осмысленными записи вида $(T' \cup T'')_{\text{гр}}$, $(T \setminus T')_{\text{гр}}$ и т. п.

Распределение вероятностей $P : \hat{s}(T) \to R$ будет называться марковским на грамматике $\langle T, S, \mathcal{T}, \Omega\rangle$, если для любых $T' \subset T$, $T'' \subset T$, $T' \cup T'' = T$, $T' \cap T'' = \varnothing$ справедливо равенство $P(T'/T'') = P(T'/T''_{\text{гр}})$ и, кроме того, $\sum_{s \in S(G)} P(s) = 1$. Случайный вариант, имеющий распределение вероятностей, марковское на двумерной грамматике, будем называть марковским вариантом. Определение марковского варианта есть очевидное расширение понятия 1-марковского поля, исследованного Р. Л. Добрушиным [34, 35]. Расширение понятия d-марковского поля для любого целого d на случай двумерных грамматик не составляет труда, но в контексте данной работы в этом нет необходимости.

Нас интересует вопрос о задании марковского распределения, несмотря на то что исчерпывающий ответ на этот вопрос дает известная теорема Аверинцева [1, 2]. Мы возвращаемся к этому вопросу, так как хотим найти набор параметров, который не только однозначно определяет марковское распределение, а позволяет к тому же выполнять различные операции с этим распределением. Например, если заданы значения параметров, однозначно определяющие некоторое конкретное распределение вероятностей $P : \hat{s}(T) \to R$, то нам хотелось бы, чтобы существовал простой алгоритм вычисления значений этих параметров для функций $P(T')$, $P(T'/T'')$ и т. п. Поэтому тот набор параметров, который мы далее предложим для задания марковских распределений, окажется менее экономным, чем набор потенциалов, достаточность которых при определенных условиях регулярности была доказана Аверинцевым.

По аналогии с k-исключением вершины из графа, введенным в предыдущей главе, введем понятие k-исключения клетки t из грамматики.

Считается, что грамматика $G_1 = \langle T_1, S_1, \mathcal{T}_1, \Omega_1\rangle$ есть результат k-исключения вершины $t \in T$ из грамматики $G = \langle T, S, \mathcal{T}, \Omega\rangle$, если $\langle T_1, \mathcal{T}_1\rangle$ — граф, получаемый в результате k-исключения вершины из графа $\langle T, \mathcal{T}\rangle$, $\Omega = \{F(T'), T' \in \mathcal{T}\}$, $\Omega_1 = \{F_1(T'), T' \in \mathcal{T}_1\}$, причем для всех $T' \in \mathcal{T}_1 \cap \mathcal{T}$

выполняется равенство $F_1(T') = F(T')$, а для всех $T' \in \mathcal{T}_1$, $T' \notin \mathcal{T}$ и $s(T') \in \hat{s}(T')$ выполняется условие $F_1(T', s(T')) = 1$.

Отметим, что $S(G_1)$ не есть множество сужений на $T \setminus \{t\}$ вариантов из $S(G)$. Это значит, что в грамматике G_1 возможен допустимый вариант $(T \setminus \{t\}) \to S$, для которого не существует расширения на T, допустимого в G. И тем не менее справедлива следующая теорема.

Теорема 1. *Пусть $G = \langle T, S, \mathcal{T}, \Omega\rangle$; Q — двумерная грамматика, получаемая в результате k-исключения клетки t из G; P — распределение вероятностей, марковское на G. В таком случае $P(T \setminus \{t\})$ есть распределение вероятностей, марковское на Q.*

Доказательство. Рассмотрим два любых множества T' и T'', таких, что $T' \cup T'' = T \setminus \{t\}$, а $T' \cap T'' = \varnothing$, и проанализируем условное распределение вероятностей $P(T'/T'')$. Этот анализ зависит от конфигурации границы множества T'', которая, в свою очередь, зависит от того, в какой грамматике, G или Q, эта граница определяется. Границу множества T'' в грамматике G обозначим через $(T'')^G_{\text{гр}}$, а в грамматике Q — через $(T'')^Q_{\text{гр}}$. Докажем равенство $P(T'/(T')^Q_{\text{гр}}) = P(T'/T'')$ отдельно для каждого из следующих двух случаев:

$$1. \ (T'')^G_{\text{гр}} = (T'')^Q_{\text{гр}}. \tag{4.4}$$

$$2. \ (T'')^G_{\text{гр}} \neq (T'')^Q_{\text{гр}}. \tag{4.5}$$

Случай 1. Для любого $s(T' \cup T'') \in \hat{s}(T' \cup T'')$ справедливо равенство

$$P(T'/T'', s(T' \cup T'')) = \sum_{s(t) \in S} P(T' \cup \{t\}/T'', s(T)). \tag{4.6}$$

Вследствие того что P есть марковское на G распределение, для любого $s(T)$ справедливо равенство

$$P(T' \cup \{t\}/T'', s(t)) = P(T' \cup \{t\}/(T'')^G_{\text{гр}}, s(T' \cup \{t\} \cup (T'')^G_{\text{гр}})). \tag{4.7}$$

В свою очередь по правилу вычисления вероятности совместного события равенство

$$P(T' \cup \{t\}/(T'')^G_{\text{гр}}, s(T' \cup \{t\} \cup (T'')^G_{\text{гр}})) = $$
$$= P(T'/(T'')^G_{\text{гр}}, s(T' \cup (T'')^G_{\text{гр}})) \times$$
$$\times P(\{t\}/(T' \cup (T'')^G_{\text{гр}}, s(T' \cup \{t\} \cup (T'')^G_{\text{гр}})) \tag{4.8}$$

справедливо для любого $s(T' \cup \{t\} \cup (T'')^G_{\text{гр}})$.

С учетом (4.7) и (4.8) имеем

$$P(T' \cup \{t\}/T'', s(T)) = P(T'/(T'')^G_{\text{гр}}, s(T' \cup (T'')^G_{\text{гр}})) \times$$
$$\times (d\{t\}/(T' \cup (T'')^G_{\text{гр}}), s(T' \cup \{t\} \cup (T'')^G_{\text{гр}})). \qquad (4.9)$$

Первый сомножитель в правой части (4.9) не зависит от $s(t)$, а второй таков, что для любого $s(T' \cup (T'')^G_{\text{гр}})$ справедливо равенство $\sum_{s(t) \in S} P(\{t\}/(T' \cup (T'')^G_{\text{гр}}), s(T' \cup \{t\} \cup (T'')^G_{\text{гр}})) = 1$. Подставляя с учетом сказанного правую часть (4.9) в выражение (4.6), для любого $s(T' \cup T'')$ получаем

$$P(T'/T'', s(T' \cup T'')) = P(T'/(T'')^G_{\text{гр}}, s(T' \cup (T'')^G_{\text{гр}})). \qquad (4.10)$$

Учитывая предположение (4.4), для любого $s(T' \cup T'')$ имеем

$$P(T'/T'', s(T' \cup T'')) = P(T'/(T'')^Q_{\text{гр}}, s(T' \cup (T'')^Q_{\text{гр}})),$$

или, в более сокращенной записи,

$$P(T'/T'') = P(T'/(T'')^Q_{\text{гр}}), \qquad (4.11)$$

что требовалось доказать для случая (4.4).

Случай 2. Докажем вначале, что из неравенства (4.5) следует равенство

$$(T'' \cup \{t\})^G_{\text{гр}} = (T'')^Q_{\text{гр}}. \qquad (4.12)$$

Неравенство (4.5) в более развернутой записи означает

$$\exists t^* \in T'' ((\exists t' \in T' \cup \{t\} (\{t^*, t'\} \in \mathcal{T})) \wedge (\forall t'' \in T' (\{t^*, t''\} \notin \mathcal{T}_1))). \qquad (4.13)$$

Клеткой t', существование которой утверждает (4.13), может быть лишь клетка, не входящая ни в T'', ни в T', т. е. клетка t.
Действительно, клетка t' не входит в T'', поскольку $t' \in T' \cup \{t\}$. В то же время клетка t' не входит и в T', поскольку в этом случае из $\{t^*, t'\} \in \mathcal{T}$ следует $\{t^*, t'\} \in \mathcal{T}_1$, что противоречит утверждению $\forall t'' \in T' (\{t^*, t''\} \notin \mathcal{T}_1)$, входящему в (4.13). Вследствие того, что $t' = t$, высказывание (4.13) приобретает вид $\exists t^* \in T'' ((\{t^*, t\} \in \mathcal{T}) \wedge (\forall t'' \in T' (\{t^*, t''\} \notin \mathcal{T}_1)))$. Отсюда следует, что клетка t не имеет соседей в T', так как в противном случае в соответствии с правилами k-исключения клетки t клетка t^* также стала бы соседней в структуре $\mathcal{T}_1$ с некоторой клеткой из T', (что противоречит утверждению $\forall t'' \in T' (\{t^*, t''\} \notin \mathcal{T}_1)$. Таким образом, клетка t в грамматике G не входит в границу множества $T'' \cup \{t\}$, т. е. это значит, что

$$t \notin (T'' \cup \{t\})^G_{\text{гр}}, \qquad (4.14)$$

или, что то же,

$$\forall t' \in T' (\{t, t'\} \notin \mathcal{T}). \qquad (4.15)$$

Множество $(T'' \cup \{t\})^G_{\text{гр}}$ по определению есть множество клеток из $T'' \cup \{t\}$, имеющих соседей в T'. Из (4.14) следует, что это множество совпадает с множеством M_1 клеток из T'', имеющих соседей в T' в структуре $\mathcal{T}$.

Множество $(T'')^Q_{\text{гр}}$ по определению включает в себя множество M_1 и еще некоторые клетки $t^{**} \in T''$, такие, что $\exists t^* \in T' ((\{t^*, t\} \in \mathcal{T}) \wedge (\{t^{**}, t\} \in \mathcal{T}))$. Однако в силу (4.15) таких клеток не существует, следовательно, $(T'')^Q_{\text{гр}} = M_1$, что доказывает равенство (4.12).

По правилу вычисления вероятности совместного события для любого $s(T' \cup T'')$ справедливо равенство

$$P(T'/T'', s(T' \cup T'')) =$$
$$= \sum_{s(t) \in S} P(T'/(T'' \cup \{t\}), s(T)) \cdot P(\{t\}/T' \cup T'', s(T)). \qquad (4.16)$$

Поскольку P есть марковское распределение вероятностей, для первого сомножителя под знаком суммы справедливо равенство

$$P(T'/(T'' \cup \{t\}), s(T)) = P(T'/(T'' \cup \{t\})^G_{\text{гр}}, s(T' \cup (T'' \cup \{t\})^G_{\text{гр}})). \qquad (4.17)$$

В силу (4.14) это выражение не зависит от $s(t)$. Второй сомножитель под знаком суммы в (4.16) таков, что для любого $s(T' \cup T'')$ можно записать

$$\sum_{s(t) \in S} P(\{t\}/T' \cup T'', s(T)) = 1. \qquad (4.18)$$

Поэтому равенство (4.16) приобретает вид

$$P(T'/T'', s(T' \cup T'')) = P(T'/(T'' \cup \{t\})^G_{\text{гр}}, s(T' \cup (T'' \cup \{t\})^G_{\text{гр}})),$$

а с учетом доказанного равенства (4.12) — вид

$$P(T'/T'', s(T' \cup T'')) = P(T'/(T'')^Q_{\text{гр}}, s(T' \cup (T'')^Q_{\text{гр}})).$$

Последнее равенство в более короткой записи примет вид

$$P(T'/T'') = P(T'/(T'')^Q_{\text{гр}}). \qquad (4.19)$$

Таким образом, равенство (4.19) справедливо как в первом (см. (4.11)), так и во втором рассматриваемых нами случаях. Теорема доказана.

Рассмотрим грамматику $G = \langle T, S, \mathcal{T}, \Omega \rangle$, последовательность клеток $t_1, t_2, \ldots, t_{|T|}$ и грамматик $G_0, G_1, \ldots, G_{|T|}$, таких, что $G_{|T|} = G$, а для $i = 1, 2, \ldots, |T|$ грамматика G_{i-1} есть результат k-исключения клетки t из грамматики G_i. Пусть $\mathcal{T}_i$ — структура грамматики G_i.

Введем обозначения: $T_{i-1} = T_i \setminus \{t\}$; $T_i^* = \{t \in T_i : \{t, t_i\} \in \mathcal{T}_i$. Для любого распределения вероятностей P справедливо равенство.

$$P(T,\ s(T)) = \prod_{i=1}^{|T|} P(\{t_i\}/T_{i-1},\ s(T_i)).$$

Для марковского распределения вероятностей из только что доказанной теоремы следует, что

$$P(T,\ s(T)) = \prod_{i=1}^{|T|} P(\{t_i\}/T_i^*,\ s(\{t\} \cup T_i^*)). \tag{4.20}$$

Как было показано в предыдущей главе, для любой грамматики k-й степени возможен такой выбор последовательности клеток $t_1,\ t_2,\ \ldots,\ t_{|T|}$, т. е. такая нумерация клеток, что для всех i выполняется неравенство $|T_i^*| \leqslant k$. Следовательно, для этого случая задание всех функций $P(\{t_i\}/T_i^*)$, являющихся сомножителями в (4.20), выполняется заданием не более чем $|S|^k \cdot ((|S|-1)(|T|-k)+1)-1$ чисел, а (4.20) есть рабочая формула для вычисления вероятности каждого варианта.

4.3. ВЕРОЯТНОСТНАЯ МОДЕЛЬ НАБЛЮДЕНИЙ И ЗАДАЧИ РАСПОЗНАВАНИЯ

Рассмотрим две вероятностные модели наблюдений.

Модель 1. Пусть X — конечное множество, элементы которого называются результатами наблюдений. Наблюдением называется функция $x : T \to X$, т. е. совокупность результатов $x(t)$, $t \in T$, где $x(t)$ — результат наблюдения варианта s в клетке t. Считается, что $x(t)$ есть случайная величина, имеющая распределение вероятностей $U(t, s(t)) : X \to R$, которое зависит от номера t наблюдаемой клетки и структурного элемента $s(t)$, записанного в этой клетке. Считается также, что результат наблюдения $x(t)$ в клетке t при фиксированном структурном элементе $s(t)$ не зависит от структурных элементов в других клетках и не зависит от результатов наблюдений в других клетках. Таким образом, при фиксированном варианте $s : T \to S$ вероятность наблюдения $x : T \to X$ равна $\prod_{t \in T} U(t, s(t), x(t))$.

Байесовская задача распознавания варианта в этом случае имеет следующую формулировку. Дано: $G = \langle T, S, \mathcal{T}, \Omega \rangle$ — двумерная грамматика; $P : S(G) \to R$ — априорное распределение вероятностей вариантов $T \to S$, являющееся марковским; X — конечное множество результатов наблюдений; $U(t, s) : X \to R$ — условные распределения вероятностей, определенные для каждого $t \in T$ и $s \in S$; $W : S^T \times S^T \to R$ — функция потерь; $x : T \to X$ — наблюдение.

Требуется определить вариант s^*, минимизирующий апостериорное математическое ожидание потери, т. е.

$$s^* = \arg\min_{s \in S^T} \sum_{s' \in S^T} P(s') \left(\prod_{t \in T} U(t,\ s'(t),\ x(t))\right) W(s,\ s').$$

Модель II. В модели II не вводится дополнительное множество X результатов наблюдений, а предполагается, что разумное решение о варианте $s : T \to S$ должно быть принято, если известно сужение $s(T')$ этого варианта на некоторое подмножество T'. Это значит, что распознаванием варианта считается восстановление варианта по известному его фрагменту.

Пусть $P : S^T \to R$ — распределение вероятностей вариантов $T \to S$. Любую пару $(T^H,\ s^H(T^H))$, такую, что $P(T^H,\ s^H(T^H)) \neq 0$, будем считать наблюдением. Апостериорное распределение вероятностей вариантов по наблюдению $(T^H,\ s^H(T'))$ обозначим как $P_{T^H, s^H(T^H)}$. Апостериорная вероятность варианта s определяется по формуле

$$\begin{aligned} &P_{T^H, s^H(T^H)}(s) = 0, \text{ если } s(T^H) \neq s^H(T^H); \\ &P_{T^H, s^H(T^H)}(s) = P(T \setminus T^H/T^H,\ s(T)), \text{ если } s(T^H) = s^H(T^H). \end{aligned} \tag{4.21}$$

Байесовская задача распознавания варианта в этом случае заключается в следующем. Дано: $G = \langle T, V, S, \Omega \rangle$ — двумерная грамматика; $P : S(G) \to R$ — априорное распределение вероятностей вариантов $T \to S$, являющееся марковским; $W : S^T \times S^T \to R$ — функция потерь; $(T^H \subset T,\ s^H(T^H))$ — наблюдение.

Требуется определить вариант s^*, минимизирующий апостериорное математическое ожидание потери, т. е.

$$s^* = \arg\min_{s \in S^T} \sum_{s' \in S^T} P_{T^H, s^H(T^H)}(s') \cdot W(s,\ s').$$

Любая задача распознавания вариантов, сформулированная в терминах первой модели, может быть представлена в форме только что приведенной задачи. Поэтому в дальнейшем будет рассмотрена только вторая модель, в которой распознавание понимается как восстановление варианта по известному его фрагменту. Для решения этой задачи необходимо исследовать свойства апостериорного распределения $P_{T^H, s^H(T^H)}$, что сделано в двух следующих леммах и теореме.

Индекс T^H, $s^H(T^H)$ будем понимать как оператор, превращающий распределение вероятностей P в распределение $P_{T^H, s^H(T^H)}$ в соответствии с формулой (4.21).

Лемма 1. *Для любого распределения вероятностей $P : S^T \to R$, любых подмножеств $T' \subset T$ и $T'' \subset T$, таких, что $T' \cap T'' = \varnothing$, и любого фрагмента $s^H(T' \cup T'') \in \hat{s}(T' \cup T'')$, такого, что $P(T' \cup T'', s^H(T' \cup T'')) \neq 0$, справедливо равенство*

$$P_{T' \cup T'', s^H(T' \cup T'')} = (P_{T', s^H(T')})_{T'', s^H(T'')}.$$

Доказательство. Нам необходимо доказать, что равенство

$$P_{T' \cup T'', s^H(T' \cup T'')}(s) = (P_{T', s^H(T')})_{T'', s^H(T'')}(s) \tag{4.22}$$

выполняется для всех $s \in S^T$. В случае когда $s(T' \cup T'') \neq s^H(T' \cup T'')$, это равенство достаточно очевидно следует из (4.21). Докажем (4.22), когда $s(T' \cup T'') = s^H(T' \cup T'')$.

В соответствии с определением (4.21) для любого $T^* \subset T$, $s^*(T^*)$ и s, такого, что $s(T^*) = s^*(T^*)$, выполняется равенство

$$P_{T^*, s^*(T^*)}(s) = P_{T^*, s^*(T^*)}(T \setminus T^*, s(T \setminus T^*)). \tag{4.23}$$

По определению (4.21) справедливы еще следующих три равенства:

$$P_{T' \cup T'', s^H(T' \cup T'')}(s) = \frac{P(s)}{P(T' \cup T'', s^H(T' \cup T''))}, \tag{4.24}$$

$$P_{T', s^H(T')}(s) = \frac{P(s)}{P(T', s^H(T'))}, \tag{4.25}$$

$$(P_{T', s^H(T')})_{T'', s^H(T'')}(s) = \frac{P_{T', s^H(T')}(s)}{P_{T', s^H(T')}(T'', s^H(T''))}. \tag{4.26}$$

Из (4.25) следует

$$P(s) = P(T', s^H(T')) \cdot P_{T', s^H(T')}(s). \tag{4.27}$$

Справедлива цепочка равенств

$$P_{T' \cup T'', s(T' \cup T'')}(s) = \frac{P(T', s^H(T'))\, P_{T', s^H(T')}(s)}{P(T' \cup T'', s^H(T' \cup T''))} =$$

$$= \frac{P_{T', s^H(T')}(s)}{\left(\dfrac{P(T' \cup T'', s^H(T' \cup T''))}{P(T', s^H(T'))}\right)} = \frac{P_{T', s^H(T')}(s)}{P_{T', s^H(T')}(T' \cup T'', s^H(T' \cup T''))} =$$

$$= \frac{P_{T', s^H(T')}(s)}{P_{T', s^H(T')}(T'', s^H(T''))} = (P_{T', s^H(T')})_{T'', s^H(T'')}(s).$$

Первое равенство в этой цепочке получено подстановкой (4.27) в (4.24); второе равенство очевидно; третье получено в соответствии с (4.25); четвертое — в силу (4.23); наконец; пятое — в соответствии с (4.26). Следствием рассмотренной цепочки является равенство (4.22), которое и требовалось доказать. Доказательство закончено.

Лемма 2. *Пусть $G = \langle T, S, \mathcal{T}, \Omega \rangle$ — грамматика; P — марковское на этой грамматике распределение вероятностей; $t \in T$, а $s^H(\{t\}) \in S$ такой структурный элемент, что $P(\{t\}, s^H(\{t\})) \neq 0$. В таком случае апостериорное распределение вероятностей $P_{\{t\}, s^H(\{t\})}$ также является марковским на G.*

Доказательство. Нам необходимо доказать, что для любых $T' \subset T$ и $T'' \subset T$, $T' \cup T'' = T$, $T' \cap T'' = \varnothing$, и для любых s, таких, что $P_{\{t\}, s^H(\{t\})}(T'', s(T'')) \neq 0$, выполняется равенство

$$P_{\{t\}, s^H(\{t\})}(T'/T'', s(T)) = P_{\{t\}, s^H(\{t\})}(T'/T''_{\text{гр}}, s(T' \cup T''_{\text{гр}})), \tag{4.28}$$

или, что то же, доказать независимость $P_{\{t\}, s^H(\{t\})}(T'/T'', s(T))$ от $s(T'' \setminus T''_{\text{гр}})$. Доказательство этого факта выполним для следующих трех случаев:

1) $t \in T'$, $s(t) \neq s^H(t)$;

2) $t \in T'$, $s(t) = s^H(t)$;

3) $t \in T''$, $s(t) = s^H(t)$.

Четвертый случай, когда $t \in T''$, а $s(t) \neq s^H(t)$, рассматривать не следует, так как при этом $P_{\{t\}, s^H(\{t\})}(T'', s(T'')) = 0$.

Случай 1. Значение $P_{\{t\}, s^H(\{t\})}(T'/T'', s)$ равно нулю при всех значениях s, поэтому равенство (4.28) выполняется.

Случай 2. Справедливо равенство

$$P_{\{t\}, s^H(\{t\})}(T'/T'', s) = P(T' \setminus \{t\}/T'' \cup \{t\}, s). \tag{4.29}$$

Поскольку P есть марковское распределение, правая часть в этом равенстве равна $P(T' \setminus \{t\}/(T'' \cup \{t\})_{\text{гр}}, s)$. Это значит, что левая часть в (4.29) не зависит от $s((T'' \cup \{t\}) \setminus (T'' \cup \{t\})_{\text{гр}})$. Поскольку $T'' \subset (T'' \cup \{t\})$, имеем $T'' \setminus T''_{\text{гр}} \subset (T'' \cup$

$\cup \{t\}) \setminus (T'' \cup \{t\})_{гр}$ и, следовательно, значение $P_{\{t\},s^H(\{t\})}$ $(T'/T'', s)$ не зависит от $s\,(T'' \setminus T''_{гр})$.

Случай 3. Справедливо равенство

$$P_{\{t\},s^H(\{t\})}(T'/T'', s) = P(T'/T'', s). \tag{4.30}$$

Поскольку выражение в правой части (4.30) не зависит от $s\,(T'' \setminus T''_{гр})$, выражение в левой части также не зависит от $s\,(T'' \setminus T''_{гр})$. Лемма доказана.

Теорема 2. *Пусть P — распределение вероятностей, марковское на грамматике G; $T^H \subset T$, а $s^H\,(T^H)$ — фрагмент, такой, что $P\,(T^H,\ s^H\,(T^H)) \neq 0$. В этом случае апостериорное распределение $P_{T^H, s^H(T^H)}$ также является марковским на G.*

Доказательство. Пусть $T^H = \{t^1,\ t^2,\ \ldots,\ t^m\}$. Тогда в силу леммы 1 справедливо равенство

$$P_{T^H, s^H(T')} = (\ldots ((P_{\{t^1\},s^H(t^1)})_{\{t^2\},s^H(t^2)})\ \ldots)_{\{t^m\},s^H(\{t^m\})}.$$

В силу леммы 2 распределение вероятностей $P_{\{t^1\},s^H(\{t^1\})}$ — марковское. В таком случае в силу этой же леммы $(P_{\{t^1\},s^H(\{t^1\})})_{\{t^2\},s^H(\{t^2\})}$ — также марковское и т. д. Теорема доказана.

Отметим, что доказательство аналогичной теоремы в терминах первой модели было бы более громоздким, чем доказательство двух предшествующих лемм.

В дальнейшем до конца главы будем предполагать, что если задана двумерная грамматика $G = \langle T,\ S,\ \mathcal{T},\ \Omega\rangle$, то одновременно с ней известно следующее.

1. Последовательность $t_1,\ t_2,\ \ldots,\ t_n,\ n = |\,T\,|$, клеток поля T, т. е. определенная фиксированная нумерация клеток. Везде в дальнейшем нижний индекс при идентификаторе клетки будет обозначать номер клетки в этой нумерации.

2. Последовательность грамматик $G_i = \langle T_i,\ S_i,\ \mathcal{T}_i,\ \Omega_i\rangle$, $i = 1,\ 2,\ \ldots,\ n$, где $G_n = G$, а грамматика G_{i-1} получена в результате k-исключения вершины t_i из грамматики G_i.

3. Последовательность $T_1^*,\ T_2^*,\ \ldots,\ T_n^*$, где $T_i^*,\ i = 1,\ 2,\ \ldots$ $\ldots,\ n$,— множество клеток, соседних с клеткой t_i в грамматике G_i.

4. Множество $\mathcal{T}^*$, содержащее в качестве своих элементов все различные множества T_i^*, $i = 1,\ 2,\ \ldots,\ n$.

5. Функция $R : T_n \to \mathcal{T}^*$, такая, что $R\,(t_i) = T_i^*$, $i = 1$, $2,\ \ldots,\ n$.

6. Отношения $\rightarrow$, $\Rightarrow$ на множестве $(\mathcal{T}^* \cup T_n) \times T_n$, множества $T_{сл}\,(t)$, $\hat{\mathcal{T}}\,(t)$, $t \in T_n$, и $T_{сл}\,(T^*)$, $\hat{T}\,(T^*)$, $T^* \in \mathcal{T}^*$, такие, как было определено в параграфе 3 предыдущей главы.

Если мы будем говорить, что задано или требуется определить марковское распределение вероятностей P, то это будет означать, что заданы или требуется определить числа $P\,(\{t\}/R\,(t),\ s\,(R\,(t) \cup \{t\}))$ для всех $t \in T_n$ и всех $s\,(R\,(t) \cup \{t\}) \in S^{R(t)\cup\{t\}}$. Совокупность этих чисел будет называться вероятностными параметрами марковского распределения или грамматики. Как показано ранее, любое марковское распределение однозначно определяется своими вероятностными параметрами, а рабочей формулой для вычисления вероятности варианта s является

$$P\,(s) = \prod_{t \in T_n} P\,(\{t\}/R\,(t),\ s\,(R\,(t) \cup \{t\})). \tag{4.29}$$

Примечание. Ранее отмечалось, что задание марковского распределения вероятностными параметрами не самое экономное. Это значит, что хотя любое марковское распределение имеет вид (4.29), но не любое распределение вида (4.29) — марковское. Любое распределение вида (4.29) будет называться марковской аппроксимацией. Конец примечания.

Рассмотрим вопрос о вычислении вероятностных параметров апостериорного марковского распределения $P_{T^H, s^H(T^H)}$, если задано априорное марковское распределение P вариантов и наблюдение $(T^H,\ s^H\,(T^H))$.

Для любого $T' \subset T$ и $s\,(T') \in \hat{s}\,(T')$ через $S\,(G,\ T',\ s\,(T'))$ будем обозначать множество допустимых в G вариантов s^*, таких, что $s^*\,(T') = s\,(T')$. Через $S\,(G,\ T^1,\ s^1\,(T^1),\ T^2,\ s^2\,(T^2), \ldots)$ будем обозначать $S\,(G,\ T^1,\ s^1\,(T^1)) \cap S\,(G,\ T^2,\ s^2\,(T^2)) \cap \ldots$

В соответствии с формулой (4.21), определяющей апостериорное распределение $P_{T^H, s^H(T^H)}$, вероятность $P_{T^H, s^H(T^H)}\,(s)$ для любого варианта s равна либо 0, когда $s \notin S\,(G,\ T^H,$ $s^H\,(T^H))$, либо $c \cdot P\,(s)$, где c — величина, не зависящая от s. В силу этого для любого T' апостериорная вероятность $P_{T^H, s^H(T^H)}\,(T',\ s\,(T'))$ равна сумме $\sum_{s \in S(G, T^H, s^H(T^H), T', s(T'))} P\,(s)$ с точностью до не зависящего от $s\,(T')$ множителя. Поэтому

$$P_{T^H, s^H(T^H)}\,(\{t\} \cup R\,(t),\ s\,(\{t\} \cup R\,(t))) =$$
$$= c \sum_{s' \in S(G, T^H, s^H(T^H), \{t\} \cup R(t), s(\{t\} \cup R(t)))} P\,(s'). \tag{4.30}$$

Множество вариантов, по которому производится суммирование в этом выражении, есть варианты s', такие, что $s'\,(T^H) =$ $= s^H\,(T^H)$, а $s'\,(\{t\} \cup R\,(t)) = s\,(\{t\} \cup R\,(t))$. Это множество очевидно, может быть представлено в виде произведения $S_0 \times S_{сл} \times S_{до}$, где S_0, $S_{сл}$, $S_{до}$ — множества фрагментов, имеющих область определения $R(t)$, $T_{\text{сл}}(R(t))$ и $T \setminus (T_{\text{сл}}(R(t)) \cup R(t))$. Множество S_0, очевидно, содержит не более одного фрагмента, а именно фрагмент $s(R(t))$, если условия $s'(R(t)) = s(R(t))$ и $s'(T^H) = s^H(T^H)$ не являются противоречивыми. Множество $S_{\text{сл}}$ есть множество тех фрагментов $s' : T_{\text{сл}}(R(t)) \to S$, для которых $s'(t') = s^H(t')$ при $t' \in T_{\text{сл}}(R(t)) \cap T^H$, а $s'(t) = s(t)$. Аналогичный вид имеет множество $S_{\text{до}}$. Вероятность $P(s')$, являющаяся слагаемым в правой части (4.30), также может быть представлена в виде произведения трех сомножителей:

$$P(s') = P(R(t),\ s'(R(t)) \times$$
$$\times P(T_{\text{сл}}(R(t))/R(t),\ s'(T_{\text{сл}}(R(t) \cup R(t)) \times$$
$$\times P(T \setminus (T_{\text{сл}}(R(t)) \cup R(t))/R(t),\ s'(T \setminus T_{\text{сл}}(R(t)))).$$

Поэтому равенство (4.30) может быть записано в виде

$$P_{T^H, s^H(T^H)}(\{t\} \cup R(t),\ s(\{t\} \cup R(t)) =$$
$$= c\Big(\sum_{s'(R(t)) \in S_0} P(R(t),\ s'(R(t)))\Big) \times$$
$$\times \sum_{s'(T_{\text{сл}}(R(t)) \cup R(t)) \in S_{\text{сл}}} P(T_{\text{сл}}(R(t))/R(t),\ s'(T_{\text{сл}}(R(t)) \cup R(t))) \times$$
$$\times \sum_{s'(T \setminus T_{\text{сл}}(R(t))) \in S_{\text{до}}} P(T \setminus (T_{\text{сл}}(R(t)) \cup R(t))/R(t),$$
$$s'(T \setminus T_{\text{сл}}(R(t)))).$$

Первая и третья суммы в правой части этого выражения зависят только от $s(R(t))$ и не зависят от $s(t)$, а вторая сумма зависит как от $s(R(t))$, так и от $s(t)$. В силу этого для вычисления вероятностных параметров $P_{T^H, s^H(T^H)}(\{t\}/R(t),\ s(\{t\} \cup R(t))$ достаточно знать только вторую сумму. Действительно,

$$P_{T^H, s^H(T^H)}(\{t\} \cup R(t),\ s(R(t) \cup \{t\})) =$$
$$= c\varphi_1(t,\ s(R(t))) \cdot D_1(t,\ s(\{t\} \cup R(t))) \cdot \varphi_2(t,\ s(R(t))).$$

Следовательно,

$$P_{T^H, s^H(T^H)}(\{t\}/R(t),\ s(\{t\} \cup R(t))) = \frac{D_1(t,\ s(\{t\} \cup R(t)))}{\sum\limits_{s(t) \in S} D_1(t,\ s(\{t\} \cup R(t)))}. \tag{4.31}$$

Таким образом, для вычисления вероятностных параметров апостериорного распределения $P_{T^H, s^H(T^H)}$ необходимо уметь вычислять величины $D_1(t,\ s(\{t\} \cup R(t)))$ для каждой клетки $t \in T$ и каждого фрагмента $s(\{t\} \cup R(t))$, определяемые формулой

$$D_1(t,\ s(\{t\} \cup R(t))) =$$
$$= \sum_{s'(T_{\text{сл}}(R(t)) \cup R(t)) \in S_{\text{сл}}} P(T_{\text{сл}}(R(t))/R(t), s'(T_{\text{сл}}(R(t)) \cup R(t))),$$

где суммирование ведется по всем тем фрагментам, которые «проходят» через фрагмент $s(\{t\} \cup R(t))$ и через наблюдение $s^H(T^H)$, т. е. по тем фрагментам $s' : T_{\text{сл}}(R(t)) \cup R(t) \to S$, для которых $s'(\{t\} \cup R(t)) = s(\{t\} \cup R(t))$, а для всех $t' \in (R(t) \cup T_{\text{сл}}(R(t))) \cap T^H$ выполняется равенство $s'(t') = s^H(t')$.

Для построения практически хорошего алгоритма вычисления этих величин необходимо ввести вспомогательные величины

$$D_2(t,\ s(R(t))) = \sum_{s'(t) \in S} D_1(t,\ s'(t),\ s(R(t)), \quad t \in T, \tag{4.32}$$

$$D_3(T^*, s(T^*)) = \prod_{t \in \hat{T}(T^*)} D_2(t,\ s(T^*)), \quad T^* \in \mathcal{T}^*. \tag{4.33}$$

Значения $D_1(t,\ s(\{t\} \cup R(t)))$ должны вычисляться следующим образом:

$$D_1(t,\ s(\{t\} \cup R(t))) = \Big(\prod_{T^* \in \hat{\mathcal{T}}(t)} D_3(T^*,\ s(T^*))\Big) \times$$
$$\times P(\{t\}/R(t),\ s(\{t\} \cup R(t))), \text{ если } t \notin T_H \text{ или если } s(t) =$$
$$= s^H(t);\ D_1(t,\ s(\{t\} \cup R(t))) = 0 \text{ — в противном случае.} \tag{4.34}$$

Формулы (4.31), (4.32), (4.33), (4.34) являются рабочими формулами для вычисления искомых вероятностных параметров. Схема вычислений здесь в точности совпадает с описанной ранее схемой вычислений полиномиальной характеристики.

4.4. ДВЕ ВЫЧИСЛИТЕЛЬНЫЕ СХЕМЫ ДЛЯ ПРИНЯТИЯ РЕШЕНИЯ О ВАРИАНТЕ

Пусть функция потерь W имеет вид $W(s, s^*) = 0$, если $s = s^*$, и вид $W(s, s^*) = 1$, если $s \neq s^*$. Как известно, в этом случае решением байесовской задачи является вариант s^*, имеющий максимально возможную апостериорную вероятность, т. е.

$$s^* = \arg\max_{s \in S^T} \prod_{t \in T} P(\{t\}/R(t),\ s(\{t\} \cup R(t))). \tag{4.35}$$

Здесь через $P(\{t\}/R(t), s(\{t\} \cup R(t)))$ обозначены параметры апостериорного марковского распределения, которые вычислены по схеме, описанной в предыдущем параграфе.

Решение оптимизационной задачи (4.35) выполняется с помощью рекуррентных соотношений, подобных белмановским [15]. Введем обозначение

$$D_1(t, s(\{t\} \cup R(t))) =$$
$$= \max_{s(T_{\text{сл}}(t)\setminus\{t\})\in \hat{s}(T_{\text{сл}}(t)\setminus\{t\})} \prod_{t'\in T_{\text{сл}}(t)} P(\{t'\}/R(t'), s(\{t'\} \cup R(t'))) \quad (4.36)$$

и введем еще три вспомогательные величины:

$$D_2(t, s(R(t))) = \max_{s'(t)\in S} D_1(t, s'(t), s(R(t))), \quad (4.37)$$

$$s^*(t, s(R(t))) = \arg\max_{s'(t)\in S} D_1(t, s'(t), s(R(t))), \quad (4.38)$$

$$D_3(T^*, s(T^*)) = \prod_{t\in\hat{T}(T^*)} D_2(t, s(R(t))). \quad (4.39)$$

Справедливо следующее равенство:

$$D_1(t, s(\{t\} \cup R(t))) = \left(\prod_{T^*\in\hat{\mathscr{T}}(t)} D_3(T^*, s(T^*))\right) \times$$
$$\times P(\{t\}/R(t), s(\{t\} \cup R(t))). \quad (4.40)$$

Последовательно используя формулы (4.37) — (4.40) по описанной ранее схеме, применяемой для вычисления полиномиальных характеристик, получаем функции $s^*(t, s(R(t)))$ для всех $t \in T$ и всех фрагментов $s(R(t))$. Совокупность этих функций позволяет указать оптимальный вариант s^* по следующей схеме. Значение $s^*(t_1)$ равно $s^*(t_1, \#)$, так как $R(t_1) = \varnothing$, следовательно, $s(R(t_1)) = \#$. Допустим, что уже вычислены значения оптимального варианта для всех клеток t_i, имеющих номер i, меньший i^*. В таком случае известно значение оптимального варианта для всех клеток из $R(t_{i^*})$, так как все эти клетки имеют номер, меньший i^*. Следовательно, можно получить и значение $s^*(t_{i^*})$ по формуле $s^*(t_{i^*}) = s^*(t_{i^*}, s^*(R(t_{i^*})))$.

Описанная схема принятия решения о варианте обеспечивает минимум вероятности ошибочного указания варианта, что является следствием специально выбранной функции потерь. Вероятность ошибочного ответа — излюбленный критерий в распознавании образов. Если количество возможных решений мало, в частности равно 2—3, то этот критерий является действительно наиболее уместным. В наших задачах принимается решение о варианте, т. е. о большом количестве оцениваемых величин, и в этом случае формулировка задачи на минимум вероятности ошибки не всегда отражает те содержательные требования, которые предъявляются к алгоритму распознавания. Так, например, следует подвергнуть сомнению известное решение задачи распознавания последовательности алфавитно-цифровых знаков, т. е. распознавание строк текста. Хотя здесь ясно, что критерием качества функционирования распознающего алгоритма является количество неправильно распознанных знаков, задача распознавания строки ставится и решается как задача минимизации количества неправильно распознанных строк. Такая замена производится при интуитивно понимаемом допущении, что алгоритм, распознающий строки с минимально возможной вероятностью ошибки, будет обеспечивать и минимум количества ошибочно распознанных знаков. Это допущение совершенно неверно.

Пример. Пусть $S = \{s_1, s_2, s_3\}$ — алфавит распознаваемых знаков, а $S \times S$ — множество распознаваемых строк, состоящих из двух букв. И пусть в результате наблюдения изображения, т. е. измерений x, был сделан вывод, что апостериорная вероятность каждой из строк (s_1, s_1), (s_2, s_1), (s_3, s_2), (s_3, s_3) равна $^1/_4$, а апостериорные вероятности всех остальных строк равны нулю. Стремясь определить строку с минимальной вероятностью ошибки, мы должны указать строку с наибольшей апостериорной вероятностью. Таковой будет любая из указанных выше четырех строк. Для каждой из них вероятность того, что она не равна действительной строке, равна $^3/_4$. И, конечно, не следует указывать строку $(s_3, s_1,)$ вероятность которой равна нулю и которая заведомо не равна действительной строке. Однако именно эта строка, имеющая наименьшую возможную, т. е. нулевую вероятность, должна быть указана в качестве ответа, если мы стремимся минимизировать математическое ожидание количества неверно распознанных знаков. При выдаче строки (s_2, s_1) это математическое ожидание равно единице и легко убедиться, что для любого другого ответа оно будет больше. Описание примера закончено.

Действительно, решающее правило, минимизирующее математическое ожидание количества неправильно распознанных структурных элементов варианта, может существенно отличаться от решающего правила, минимизирующего вероятность неправильного распознавания варианта.

Определим функцию $w: T \times S \times S \to R$ и зададим функцию потерь $W: S^T \times S^T \to R$ в виде $W(s, s^*) = \sum_{t\in T} w(t, s(t), s^*(t))$. Для функции w вида

$$w(t, s(t), s^*(t)) = \begin{cases} 1 & \text{при } s(t) \neq s^*(t), \\ 0 & \text{при } s(t) = s^*(t) \end{cases} \quad (4.41)$$

решение байесовской задачи будет обеспечивать минимум математического ожидания количества неправильно распознанных структурных элементов. Эта задача заключается в том, чтобы для известного марковского распределения P и известных потерь $w(t, s(t), s^*(t))$ найти

$$s^* = \arg\min_{s' \in S^T} \sum_{s \in S^T} P(s) \sum_{t \in T} w(t, s(t), s'(t)). \tag{4.42}$$

Поменяем порядок суммирования в правой части (4.42). Представим $P(s)$ в виде $P(\{t\}, s(t)) \cdot P(T \setminus \{t\}/\{t\}, s)$. В результате получим

$$s^* = \arg\min_{s' \in S^T} \sum_{t \in T} \sum_{s(t) \in S} P(\{t\}, s(\{t\}))\, w(t, s(t), s'(t)) \times$$
$$\times \sum_{s(T \setminus \{t\}) \in S^{T \setminus \{t\}}} P(T \setminus \{t\}/\{t\}, s).$$

Сумма $\sum_{s(T \setminus \{t\}) \in S^{T \setminus \{t\}}} P(T/\{t\}/\{t\}, s)$ равна единице при любом значении $s(t)$, поэтому можно записать

$$s^* = \arg\min \sum_{t \in T} \sum_{s(t) \in S} P(\{t\}, s(\{t\}))\, w(t, s(t), s'(t)),$$

откуда следует, что для любого $t \in T$ справедливо равенство

$$s^*(t) = \arg\min_{s'(t) \in S} \sum_{s(t) \in S} P(\{t\}, s(\{t\}))\, w(t, s(t), s'(t)). \tag{4.43}$$

Таким образом, если по марковскому распределению P тем или иным образом вычислить вероятности $P(\{t\}, s(\{t\}))$ для любого $t \in T$ и любого $s(\{t\}) \in S$, то оптимизационная задача (4.42) сводится к решению $|T|$ оптимизационных задач (4.43), каждая из которых является простой, так как требует выполнения $|S|^2$ несложных операций. Основное содержание задачи (4.42), следовательно, заключается в вычислении вероятностей $P(\{t\}, s(\{t\}))$, $t \in T$, $s(\{t\}) \in S$, которые по определению равны $\sum_{s(T \setminus \{t\}) \in S^{T \setminus \{t\}}} P(s)$. Для вычисления этих вероятностей следует использовать три рабочие формулы:

$$P(\{t\} \cup R(t), s(\{t\} \cup R(t))) = P(R(t), s(R(t))) \times$$
$$\times P(\{t\}/R(t), s(\{t\} \cup R(t))), \quad t \in T. \tag{4.44}$$

Второй сомножитель в правой части (4.44) является одним из исходных данных задачи

$$P(\{t\}, s(\{t\})) = \sum_{s(R(t)) \in S^{R(t)}} P(\{t\} \cup R(t), s(\{t\} \cup R(t))), \quad t \in T. \tag{4.45}$$

Для всех $T^* \in \hat{\mathscr{T}}(t)$, $t \in T$ вероятность $P(T^*, s(T^*))$ вычисляется по формуле

$$P(T^*, s(T^*)) = \sum_{s((\{t\} \cup R(t)) \setminus T^*)} P(\{t\} \cup R(t), s(\{t\} \cup R(t))). \tag{4.46}$$

Вычисления по формулам (4.44) — (4.46) должны выполняться по следующей схеме.

До начала вычислений известна вероятность $P(T_1^*, s(T_1^*))$, так как $T_1^* = \varnothing$, а $P(\varnothing, \#) = 1$.

Допустим, что уже вычислены все вероятности $P(T_i^*, s(T_i^*))$ для всех $i \leqslant i^*$ и все вероятности $P(\{t_i\}, s(\{t_i\}))$ для $i \leqslant i^*$. В таком случае вычисляются: вероятности $P(\{t_{i^*}\} \cup R(t_{i^*}), s(\{t_{i^*}\} \cup R(t_{i^*})))$ по формуле (4.44), вероятности $P(\{t_{i^*}\}, s(\{t_{i^*}\}))$ по формуле (4.45) и вероятности $P(T^*, s(T^*))$ для $T^* \in \hat{\mathscr{T}}(t_{i^*})$ по формуле (4.46).

4.5. ЗАДАЧА ОБУЧЕНИЯ I

Применение общей формулировки задачи обучения I для двумерных грамматик приводит к следующей постановке. Для заданной грамматики $G = \langle T, S, \mathscr{T}, \Omega \rangle$ и выборки $\tilde{s}$ вариантов $s_1, s_2, \ldots, s_m$, допустимых в этой грамматике, требуется определить вероятностные параметры $P(\{t\}/R(t), s(\{t\} \cup R(t)))$, максимизирующие функционал $\prod_{j=1}^{m} \prod_{t \in T} P(\{t\}/R(t), s_j(\{t\} \cup R(t)))$. Эта задача оказывается простой. Нетрудно показать, что искомые значения вероятностных параметров определяются формулой

$$P(\{t\}/R(t), s(\{t\} \cup R(t))) = \frac{n(t, s(\{t\} \cup R(t)))}{n'(t, s(R(t)))}, \tag{4.47}$$

где $n(t, s(\{t\} \cup R(t)))$ — количество вариантов в выборке $\tilde{s}$, сужение которых на $\{t\} \cup R(t)$ равно $s(\{t\} \cup R(t))$, а $n'(t, s(R(t)))$ — количество вариантов в выборке $\tilde{s}$, сужение которых на $R(t)$ равно $s(R(t))$.

Формула (4.47) представляет интерес в том смысле, что позволяет дать определение, которое окажется полезным при решении следующих, более сложных задач. Обобщим формулу (4.47) для случая, когда задана не обучающая выборка $\tilde{s}$, а обучающее множество $\hat{s}$ — подмножество допустимых вариантов с заданным на нем распределением вероятностей $P : \hat{s} \to R$. Для любого распределения $P : \hat{s} \to R$ будем считать определенными вероятностные параметры $P^M(\{t\}/R(t), s(\{t\} \cup R(t)))$ в соответствии с формулой

$$P^M(\{t\}/R(t), s(\{t\} \cup R(t))) = \frac{\sum\limits_{s' \in \hat{s}^*(t, s(\{t\} \cup R(t)))} P(s')}{\sum\limits_{s' \in \hat{s}^{**}(t, s(R(t)))} P(s')}, \quad (4.48)$$

где $\hat{s}^*(t, s(\{t\} \cup R(s)))$ — множество вариантов в множестве $\hat{s}$, сужение которых на $\{t\} \cup R(t)$ есть $s(\{t\} \cup R(t))$, а $\hat{s}^{**}(t, s(R(t)))$ — множество вариантов в множестве $\hat{s}$, сужение которых на $R(t)$ есть $s(R(t))$. Совокупность параметров (4.48) определяет распределение вероятностей $P^M: S^T \to R$, такое, что вероятность варианта s определяется формулой

$$P^M(s) = \prod_{t \in T} P^M(\{t\}/R(t), s(\{t\} \cup R(t))).$$

Индекс M в выражении P^M следует понимать как оператор, переводящий функцию $P: \hat{s} \to R$ в функцию $P^M: S^T \to R$. Это значит, что в дальнейшем мы будем считать осмысленными выражения вида $(\alpha P_1 + \beta P_2)^M$ или вида $(\alpha P_1 + P_2)^M(s)$ и т. п. Распределение вероятностей P^M будем называть марковской аппроксимацией распределения вероятностей P. Марковская аппроксимация обладает определенным экстремальным свойством, которое мы будем использовать в дальнейшем.

Пусть $P'(\{t\}/R(t), s(\{t\} \cup R(t)))$ — любые вероятностные параметры, определяющие функцию $P': S^T \to R$ по известной уже формуле

$$P'(s) = \prod_{t \in T} P'(\{t\}/R(t), s(\{t\} \cup R(t))). \quad (4.49)$$

В таком случае не очень трудно доказать, что справедливо неравенство

$$\sum_{s \in \hat{S}} P(s) \log P^M(s) \geqslant \sum_{s \in \hat{S}} P(s) \log P'(s). \quad (4.50)$$

Множество распределений вида (4.49) будем обозначать буквой $\mathcal{P}$.

4.6. ЗАДАЧА ОБУЧЕНИЯ II

Эта задача заключается в том, чтобы для заданного конечного множества $\hat{S}$ вариантов определить параметры $P(\{t\}/R(t), s(\{t\} \cup R(t)))$, $t \in T$, $s(\{t\} \cup R(t)) \in S^{\{t\} \cup R(t)}$, максимизирующие функционал

$$\min_{s \in \hat{S}} \prod_{t \in T} P(\{t\}/R(t), s(\{t\} \cup R(t))). \quad (4.51)$$

Докажем, что искомым решением задачи обучения II является марковская аппроксимация определенного распределения на обучающем множестве $\hat{S}$, обладающего достаточно наглядными экстремальными свойствами.

Теорема 3. *Пусть $P: \hat{S} \to R$ — распределение вероятностей на множестве $\hat{S}$, такое, что неравенство $\sum_{s \in \hat{S}} P(s) \times \log P^M(s) \leqslant \log P^M(s)$ выполняется для всех $s \in \hat{S}$. Тогда P^M есть решение задачи II.*

Доказательство. На основании (4.50) для любого распределения $P' \in \mathcal{P}$ справедливо равенство

$$\sum_{s \in \hat{S}} P(s) \log P'(s) \leqslant \sum_{s \in \hat{S}} P(s) \log P^M(s). \quad (4.52)$$

Кроме того, поскольку $\forall s \in \hat{S} (P(s) \geqslant 0)$ и $\sum_{s \in \hat{S}} P(s) = 1$, имеет место неравенство

$$\sum_{s \in \hat{S}} P(s) \log P'(s) \geqslant \min_{s \in \hat{S}} \log P'(s). \quad (4.53)$$

Используя (4.52) и (4.53), для всех $P' \in \mathcal{P}$ имеем

$$\sum_{s \in \hat{S}} P(s) \log P^M(s) \geqslant \min_{s \in \hat{S}} \log P'(s). \quad (4.54)$$

Из условия теоремы следует, что

$$\sum_{s \in \hat{S}} P(s) \log P^M(s) \leqslant \min_{s \in \hat{S}} \log P^M(s). \quad (4.55)$$

Неравенство

$$\min_{s \in \hat{S}} \log P^M(s) \geqslant \min_{s \in \hat{S}} \log P'(s),$$

справедливость которого для любого $P' \in \mathcal{P}$ следует из (4.54) и (4.55), является доказательством справедливости теоремы. Доказательство теоремы закончено.

10 8-2045

Теорема 4. *Для того чтобы распределение вероятностей $P: \hat{S} \to R$ удовлетворяло условиям теоремы 3, необходимо и достаточно, чтобы значение P обеспечивало минимум функционала* $\sum_{s \in \hat{S}} P(s) \log P^M(s)$

Доказательство состоит из двух частей.

1. Докажем, что если неравенство

$$\sum_{s \in \hat{S}} P(s) \log P^M(s) \leqslant \log P^M(s) \tag{4.56}$$

выполняется для всех $s \in \hat{S}$, то для всех $P': \hat{S} \to R$ справедливо неравенство

$$\sum_{s \in \hat{S}} P(s) \log P^M(s) \leqslant \sum_{s \in \hat{S}} P'(s) \log P'^M(s).$$

Из справедливости (4.56) следует неравенство

$$P'(s) \sum_{s \in \hat{S}} P(s) \log P^M(s) \leqslant P'(s) \log P^M(s) \tag{4.57}$$

для всех $s \in \hat{S}$ и для любых $P'(s) \geqslant 0$.

Суммируя (4.57) для всех $s \in \hat{S}$, получаем

$$\sum_{s' \in \hat{S}} P'(s') \sum_{s \in \hat{S}} P(s) \log P^M(s) \leqslant \sum_{s \in \hat{S}} P'(s) \log P^M(s),$$

а для $P'(s)$, таких, что $\sum_{s \in \hat{S}} P'(s) = 1$,

$$\sum_{s \in \hat{S}} P(s) \log P^M(s) \leqslant \sum_{s \in \hat{S}} P'(s) \log P^M(s). \tag{4.58}$$

На основании неравенства (4.50) для любого P' справедливо неравенство

$$\sum_{s \in \hat{S}} P'(s) \log P^M(s) \leqslant \sum_{s \in \hat{S}} P'(s) \log P'^M(s). \tag{4.59}$$

А из (4.58) и (4.59) для любого P' следует справедливость неравенства

$$\sum_{s \in \hat{S}} P'(s) \log P'^M(s) \geqslant \sum_{s \in \hat{S}} P(s) \log P^M(s).$$

2. Докажем, что если существует такое $s \in \hat{S}$, что

$$\sum_{s \in \hat{S}} P(s) \log P^M(s) > \log P^M(s), \tag{4.60}$$

то существует такое P', что

$$\sum_{s \in \hat{S}} P(s) \log P^M(s) > \sum_{s \in \hat{S}} P'(s) \log P'^M(s). \tag{4.61}$$

Представим множество $\hat{S}$ в виде объединения $\hat{S}_1 \cup \hat{S}_2$, так, что $\hat{S}_1 = \{s : s \in \hat{S},\ \log P^M(s) = \min_{s' \in \hat{S}} P^M(s')\}$, а $\hat{S}_2 = \hat{S} \setminus \hat{S}_1$. Рассмотрим две функции $P_1: \hat{S} \to R$ и $P_2: \hat{S} \to R$, такие, что $P_1(s) = \dfrac{P(s)}{\sum_{s \in \hat{S}_1} P(s)}$, если $s \in \hat{S}_1$, и $P_1(s) = 0$, если $s \in \hat{S}_2$. В то же время $P_2(s) = \dfrac{P(s)}{\sum_{s \in \hat{S}_2} P(s)}$, если $s \in \hat{S}_2$, и $P_2(s) = 0$, если $s \in \hat{S}_1$. Введем обозначения: $k_1 = \sum_{s \in \hat{S}_1} P(s)$, $k_2 = \sum_{s \in \hat{S}_2} P(s)$, $k_1 + k_2 = 1$. Отметим, что $k_2 \neq 0$, так как в противном случае не выполнялось бы неравенство (4.60); k_1 может быть равным и нулю. Если $k_1 = 0$, в качестве P_1 выберем любую функцию, принимающую на $\hat{S}_2$ нулевое значение. Множество $\hat{S}_1$ не является пустым по определению. Множество $\hat{S}_2$ не пусто по той же причине, что и $k_2 \neq 0$, т. е. потому, что это противоречило бы допущению (4.60).

Обозначим через $P^*(\alpha)$ функцию $P^*(\alpha): \hat{S} \to R$, определенную для каждого $\alpha \geqslant k_1$ и $\alpha \leqslant 1$ и равную $P^*(\alpha) = \alpha \cdot P_1 + (1 - \alpha) \cdot P_2$. Очевидно, что $P^*(k_1) = P$.

Для каждого числа α, $k_1 \leqslant \alpha \leqslant 1$, определим число $F(\alpha)$ по следующей формуле:

$$F(\alpha) = \sum_{s \in \hat{S}_1} P_1(s) \log (P^*(\alpha))^M(s) - \sum_{s \in \hat{S}_2} P_2(s) \log (P^*(\alpha))^M(s).$$

Функция F — это непрерывная функция от α. Докажем, что

$$F(k_1) < 0. \tag{4.62}$$

Действительно, $P^*(k_1) = P$, следовательно, для любого $s_1 \in \hat{S}_1$ и любого $s_2 \in \hat{S}_2$ справедливо неравенство $\log (P^*(k_1))^M(s_1) < \log (P^*(k_1))^M(s_2)$. Отсюда следует $\sum_{s \in \hat{S}_1} P_1(s) \times \log (P^*(k_1))^M(s) < \sum_{s \in \hat{S}_2} P_2(s) \log (P^*(k_1))^M(s)$, что доказывает справедливость (4.62).

В результате непрерывности $F(\alpha)$ из (4.62) следует существование такого $\alpha^* > k_1$, что $F(\alpha^*) < 0$, а следовательно, и

$(\alpha^* - k_1)\, F(\alpha^*) < 0$, т. е.

$$(\alpha^* - k_1) \sum_{s\in\hat{S}_1} P_1(s) \log (P^*(\alpha^*))^M (s) -$$

$$- (\alpha^* - k_1) \sum_{s\in\hat{S}_2} P_2(s) \log (P^*(\alpha^*))^M (s) < 0. \qquad (4.63)$$

В силу (4.50) справедливо неравенство

$$\sum_{s\in\hat{S}_1} P(s) \log P^M (s) + \sum_{s\in\hat{S}_2} P(s) \log P^M (s) \geqslant$$

$$\geqslant \sum_{s\in\hat{S}_1} P(s) \log (P^*(\alpha^*))^M (s) + \sum_{s\in\hat{S}_2} P(s) \log (P^*(\alpha^*))^M (s). \quad (4.64)$$

По определению P_1 и P_2 перепишем (4.64), изменив его правую часть:

$$\sum_{s\in\hat{S}_1} P(s) \log P^M (s) + \sum_{s\in\hat{S}_2} P(s) \log P^M (s) \geqslant$$

$$\geqslant k_1 \sum_{s\in\hat{S}_1} P_1(s) \log (P^*(\alpha^*))^M (s) + k_2 \sum_{s\in\hat{S}_2} P_2(s) \log (P^*(\alpha^*))^M (s). \quad (4.65)$$

Сложив левую часть (4.63) и правую часть (4.65), получим

$$\sum_{s\in S} P(s) \log P^M (s) > \alpha^* \sum_{s\in\hat{S}_1} P_1(s) \log (P^*(\alpha^*))^M (s) +$$

$$+ (1 - \alpha^*) \sum_{s\in\hat{S}_2} P_2(s) \log (P^*(\alpha^*))^M (s) =$$

$$= \sum_{s\in\hat{S}} P^*(\alpha^*)(s) \log (P^*(\alpha^*))^M (s),$$

что доказывает справедливость (4.61). Теорема доказана.

Из только что доказанной теоремы следует, что задача II сводится к нахождению распределения вероятностей $P : \hat{S} \to R$, которое максимизирует величину $-\sum_{s\in\hat{S}} P(s) \log P^M (s)$, подобную энтропии, а искомое распределение вероятностей является марковской аппроксимацией P^M найденного распределения P. В ходе доказательства теоремы также выяснено важное свойство функционала $\sum_{s\in\hat{S}} P(s) \log P^M (s)$, а именно его одноэкстремальность. Это значит, что если некоторая функция P^* не обеспечивает минимума функционалу $\sum_{s\in\hat{S}} P(s) \log P^M (s)$, то в любой ненулевой окрестности функции P^* существует функция P^{**}, такая, что $\sum_{s\in\hat{S}} P^{**}(s) \times \log P^{**M} (s) < \sum_{s\in\hat{S}} P^*(s) \log P^{*M} (s)$.

Из сказанного представляется правдоподобным следующий алгоритм решения задачи, который далее будет обоснован.

1. Задать расходящийся ряд положительных чисел α_0, α_1, α_2, ..., α_q, ..., стремящихся к нулю.
2. Задать число $\varepsilon > 0$, определяющее требуемую точность решения задачи.
3. Принять $q = 0$.
4. Принять $P^q(s) = \frac{1}{|\hat{S}|}$ для ксех $s \in \hat{S}$.
5. Найти вариант $s^* \in \hat{S}$, для которого
$$\sum_{s\in\hat{S}} P^q(s) \log P^{qM} (s) - \log P^{qM} (s^*) \geqslant \varepsilon.$$
6. Если такого варианта нет, то «конец».
7. Определить функцию $Q^q : s \to R$, такую что $Q^q(s^*) = 1$, и $Q^q(s) = 0$, если $s \neq s^*$.
8. Определить функцию $P^{q+1} = (1 - \alpha_q) P^q + \alpha_q Q^q$.
9. Принять $q = q + 1$.
10. Перейти на п. 5.

Обоснованием этого алгоритма служат две теоремы.

Теорема 5. *Для любого $\varepsilon > 0$ и любого подмножества $\hat{S}$ допустимых вариантов получаемая алгоритмом последовательность P^q, $q = 0, 1, 2\ldots$, такова, что существует число n, такое, что неравенство $\sum_{s\in\hat{S}} P^n(s) \log P^{nM} (s) - \log P^{nM} (s) < \varepsilon$ выполняется для всех $s \in \hat{S}$.*

Доказательство. Прежде всего докажем, что функционал $\sum_{s\in\hat{S}} P(s) \log P^M (s)$ ограничен снизу на множестве всех возможных распределений $P : \hat{S} \to R$.

При доказательстве теоремы 3 было установлено (формула (4.54)), что для любых $P : \hat{S} \to R$ и любых $P' \in \mathcal{P}$ выполняется неравенство $\sum_{s\in\hat{S}} P(s) \log P^M (s) \geqslant \min_{s\in\hat{S}} \log P'(s)$. Поэтому для доказательства ограниченности функционала $\sum_{s\in\hat{S}} P(s) \log P^M (s)$ снизу достаточно убедиться, что для любого подмножества $\hat{S}$ допустимых вариантов существует такая функция $P' \in \mathcal{P}$, что $P'(s) \neq 0$ для всех $s \in \hat{S}$. Такая функция P' действительно существует, и одной из них может служить, например, марковская аппроксимация распределения $P'' : \hat{S} \to R$, такого, что $P''(s) = \frac{1}{|\hat{S}|}$ для любого $s \in \hat{S}$.

Допустим, теорема неверна, т. е. для любого целого положительного q существует $s \in \hat{S}$, такое, что выполняется неравенство

$$\sum_{s \in \hat{S}} P^q(s) \log P^{qM}(s) - \log P^{qM}(s) \geqslant \varepsilon.$$

Введем обозначения: $F_q = \sum_{s \in \hat{S}} P^q(s) \log P^{qM}(s)$ и $\Phi_q(\alpha) = \sum_{s \in \hat{S}} ((1-\alpha) P^q + \alpha Q^q)(s) \log ((1-\alpha) P^q + \alpha Q^q)^M(s)$. Нас интересует значение производной $d\Phi_q(\alpha)/d\alpha$ при $\alpha = 0$. Для этого представим функцию $\Phi_q(\alpha)$ в виде сложной функции, зависящей от промежуточных переменных $\alpha_1(\alpha)$ и $\alpha_2(\alpha)$, где $\alpha_1(\alpha) = \alpha$ и $\alpha_2(\alpha) = \alpha$. С учетом введенных обозначений

$$\Phi_q(\alpha) = \Psi_q(\alpha_1(\alpha), \alpha_2(\alpha)) = \sum_{s \in \hat{S}} ((1-\alpha_1(\alpha)) P^q + \alpha_1(\alpha) Q^q)(s) \log ((1-\alpha_2(\alpha)) P^q + \alpha_2(\alpha) Q^q)^M(s), \tag{4.66}$$

а искомая производная будет равна

$$\frac{d\Phi_q}{d\alpha} = \frac{\partial \Psi(\alpha_1(\alpha), \alpha_2(\alpha))}{\partial \alpha_1} \frac{d\alpha_1}{d\alpha} + \frac{\partial \Psi(\alpha_1(\alpha), \alpha_2(\alpha))}{\partial \alpha_2} \frac{d\alpha_2}{d\alpha}.$$

При $\alpha = 0$ производная $\frac{\partial \Psi}{\partial \alpha_1} = \sum_{s \in \hat{S}} (-P^q + Q^q)(s) \log P^{qM}(s)$, а производная $\frac{\partial \Psi}{\partial \alpha_2} = 0$, так как в силу (4.50) функционал $\sum_{s \in \hat{S}} P^q(s) \log ((1-\alpha_2) P^q + \alpha_2 Q^q)^M(s)$ при $\alpha_2 = 0$ достигает своего глобального максимума. Таким образом, при $\alpha = 0$ имеем производную

$$\frac{d\Phi_q}{d\alpha} = \sum_{s \in \hat{S}} Q^{(q)}(s) \log P^{qM}(s) - \sum_{s \in \hat{S}} P^q(s) \log P^{qM}(s).$$

Функция $Q^{(q)}$ равна нулю для всех $s \in \hat{S}$, кроме одного, а именно s^*, для которого выполняется неравенство

$$\sum_{s \in \hat{S}} P^q(s) \log P^{qM}(s) - \log P^{qM}(s^*) \geqslant \varepsilon,$$

левую часть которого, как нетрудно заметить, образует величина $-\frac{d\Phi_q}{d\alpha}$. Следовательно, при $\alpha = 0$ выполняется неравенство $\frac{d\Phi_q}{d\alpha} \leqslant -\varepsilon$. Поскольку $\Phi_{q(0)} = F_q$, а $\Phi_q(\alpha_q) = F_{q+1}$, имеем $F_q - F_{q-1} \geqslant \alpha_q \cdot \varepsilon + \delta_q$, где δ_q стремится к нулю, и к тому же так, что δ_q/α_q также стремится к нулю. Поскольку α_q, $q = 1, 2, \ldots$, есть по условию расходящийся ряд, из последнего неравенства следует, что при достаточно больших значениях q_1 и q_2 разность $F_{q_1} - F_{q_2}$ может быть сколь угодно большой, а это невозможно в силу доказанной ранее ограниченности функционала F снизу. Теорема доказана.

Теорема 6. *Если для некоторого $P : \hat{S} \to R$ и $\varepsilon > 0$ неравенство*

$$\sum_{s \in \hat{S}} P(s) \log P^M(s) - \log P^M(s) \leqslant \varepsilon \tag{4.67}$$

выполняется для всех $s \in \hat{S}$, то справедливо и неравенство

$$\max_{P' \in \mathcal{P}} \min_{s \in \hat{S}} \log P'(s) - \min_{s \in \hat{S}} \log P^M(s) \leqslant \varepsilon.$$

Доказательство. Из двух доказанных ранее теорем следует, что

$$\min_{P \in \mathcal{P}} \sum_{s \in \hat{S}'} P(s) \log P^M(s) = \max_{P' \in \mathcal{P}} \min_{s \in \hat{S}} \log P'(s). \tag{4.68}$$

Очевидно также неравенство

$$\sum_{s \in \hat{S}} P(s) \log P^M(s) \geqslant \min_{P \in \mathcal{P}} \sum_{s \in \hat{S}} P(s) \log P^M(s),$$

из которого с учетом (4.68) получаем неравенство

$$\max_{P' \in \mathcal{P}} \min_{s \in \hat{S}} P'(s) - \sum_{s \in \hat{S}} P(s) \log P^M(s) \leqslant 0. \tag{4.69}$$

Из условия (4.67) следует неравенство

$$\sum_{s \in \hat{S}} P(s) \log P^M(s) - \min_{s \in \hat{S}} \log P^M(s) \leqslant \varepsilon. \tag{4.70}$$

Складывая (4.69) и (4.70), получаем

$$\max_{P' \in \mathscr{P}} \min_{s \in \hat{S}} \log P'(s) - \min_{s \in \hat{S}} \log P^M(s) \leqslant \varepsilon.$$

Теорема доказана.

Две последние теоремы доказывают, что сформулированный ранее алгоритм гарантированно выйдет на команду останова п. 6. При этом функция P^{qM}, полученная алгоритмом на момент останова, такова, что значение $\min_{s \in \hat{S}} \log P^{qM}(s)$ отличается от своего максимально возможного значения $\max_{P \in \mathscr{P}} \min_{s \in \hat{S}} \log P(s)$ не более чем на заданную величину ε. Именно в этом смысле доказанные две теоремы являются обоснованием предложенного алгоритма, который решает задачу обучения II с любой заранее заданной точностью.

В то же время при доказательстве теорем были обнаружены определенные свойства решаемой задачи, которые позволяют несколько видоизменить алгоритм так, что его практически можно использовать и не задаваясь требуемой точностью решения. Для этого вместо приведенных ранее действий алгоритма по п. 5 следует отыскивать вариант $s^* = \arg\min_{s \in \hat{S}} p^{qM}(s)$, а п. 6 не выполнять вообще. Строго говоря, после такого видоизменения алгоритм уже перестал быть алгоритмом, так как в нем отсутствует команда останова. И тем не менее такой «алгоритм» удобен для практического использования. Прежде всего представляется правдоподобным (и в действительности верным), что в таком «алгоритме» $\lim_{q \to \infty} \min_{s \in \hat{S}} P^{qM}(s) = \max_{P \in \mathscr{P}} \min_{s \in \hat{S}} P(s)$, что само по себе уже является в определенной мере его обоснованием. Отсутствие же команды останова не препятствует практическому использованию «алгоритма» при условии наблюдения его действий пользователем.

В этом случае совместная работа должна быть организована так, что на каждый q-й итерации алгоритма вычисляются значения $\max_{i \leqslant q} \min_{s \in \hat{S}} \log P^{iM}(s)$ и $\min_{i \leqslant q} \sum_{s \in \hat{S}} P^i(s) \log P^{iM}(s)$. Первое из этих значений означает качество решения задачи обучения, достигнутое на данный момент времени, а второе — то предельно возможное качество, которое нельзя превзойти даже при неограниченном продолжении обучения. Как правило, наблюдая изменение этих двух величин в процессе работы алгоритма, пользователь достаточно уверенно принимает

решение о целесообразности прекращения работы алгоритма. Возможны и другие практически полезные видоизменения алгоритма.

И в завершение рассмотрения задачи обучения II следует сделать еще одно важное замечание. Нетрудно заметить, что теоремы 3—6 доказаны в достаточно общем виде и справедливы не только для того случая, когда $\mathscr{P}$ — множество марковских аппроксимаций на множествах, порождаемых двумерными грамматиками. Фактически конкретный вид множества $\mathscr{P}$ был использован лишь при доказательстве ограниченности функционала $\sum_{s \in \hat{S}} P(s) \log P^M(s)$ на множестве всех возможных функций вида $\hat{S} \to R$. Естественно, что это свойство имеет место не только для рассматриваемой модели, но и для очень многих других моделей. Для всех этих моделей, следовательно, оказываются справедливыми доказанные теоремы, обнаруживающие взаимосвязь двух классов задач обучения и позволяющие сводить конструирование алгоритма для решения задачи обучения II к решению задачи обучения I, которая зачастую оказывается значительно проще.

4.7. ЗАДАЧА ОБУЧЕНИЯ III

Пусть задана двумерная грамматика $G = \langle T, S, \mathscr{T}, \Omega \rangle$. По-прежнему через $\mathscr{P}$ будем обозначать множество таких распределений вероятностей $P : S^T \to R$, что для любого варианта $s \in S(G)$ выполняются равенства

$$P(s) = \prod_{t \in T} P(\{t\}/R(t),\ s(\{t\} \cup R(t))),\quad \sum_{s \in S(G)} P(s) = 1.$$

В одном из предыдущих параграфов мы для каждой грамматики G и распределения вероятностей $P \in \mathscr{P}$ определили алгоритм $A(G, P)$. Этот алгоритм для каждого наблюдения $T^H \subset T$, $s^H(T^H)$ определяет наиболее вероятный вариант s^*, такой, что $s^*(T^H) = s^H(T^H)$. Через $\mathcal{A}(G)$ обозначим класс таких алгоритмов, т. е. $\mathcal{A}(G) = \{A(G, P) : P \in \mathscr{P}\}$.

Пусть задано обучающее множество $\hat{S}$ в виде множества пар $\{(s_1, T_1), (s_2, T_2), \ldots, (s_m, T_m)\}$. Задача обучения III, или синтеза распознающего алгоритма из класса $\mathcal{A}(G)$ по заданной выборке $\hat{S}$, заключается в нахождении алгоритма $A \in \mathcal{A}(G)$, такого, что равенство $A(T^H, s(T^H)) = s$ выполняется для всех $(s, T^H) \in \hat{S}$.

Для каждой пары $(s, T^H) \in \hat{S}$ через $\hat{S}'(s, T^H)$ обозначим множество допустимых вариантов, совпадающих на T^H с вариантом s, но не равных s. Таким образом,

$$\hat{S}'(s,T^H)=\{s':s'\in S(G),\ s'\neq s,\ s'(T^H)=s(T^H)\}.$$

Синтез алгоритма $A\in\mathcal{A}(G)$, очевидно, сводится к поиску функции $P\in\mathcal{P}$, такой, что неравенство $P(s)>P(s')$ выполняется для каждого $(s,T^H)\in\hat{S}$ и $s'\in\hat{S}(s,T^H)$. Эта задача сводится, в свою очередь, к поиску таких чисел $P(\{t\}/R(t),\ s(\{t\}\cup R(t)))$, $t\in T$, $s(\{t\}\cup R(t))\in S^{\{t\}\cup R(t)}$, что неравенство

$$\prod_{t\in T}P(\{t\}/R(t),\ s(\{t\}\cup R(t)))>\prod_{t\in T}P(\{t\}/R(t),\ s'(\{t\}\cup R(t))) \tag{4.71}$$

выполняется для всех $(s,T^H)\in\hat{S}$ и $s'\in\hat{S}'(s,T^H)$. Введя обозначение $\alpha(t,\ s(\{t\}\cup R(t)))=\log P(\{t\}/R(t),\ s(\{t\}\cup R(t)))$, мы сводим нашу задачу к решению системы линейных неравенств

$$\sum_{t\in T}\alpha(t,\ s(\{t\}\cup R(t)))>$$

$$>\sum_{t\in T}\alpha(t,\ s'(\{t\}\cup R(t))),\ s'\in\hat{S}'(s,T^H),(s,T^H)\in\hat{S}, \tag{4.72}$$

относительно переменных $\alpha(t,\ s(\{t\}\cup R(t)))$. Дальнейший ход рассуждений весьма близок к обоснованию подобного алгоритма, который разработал и обосновал Б. Н. Козинец [60], конечно, для других целей.

Числа $\alpha(t,s(\{t\}\cup R(t)))$, $t\in T$, $s(\{t\}\cup R(t))\in S^{\{t\}\cup R(t)}$ будем считать компонентами вектора α в конечномерном линейном пространстве. Будем, таким образом, считать осмысленными выражения: $\gamma\cdot\alpha$ — для произведения вектора α на число γ; $\alpha_1+\alpha_2$ — для суммы двух векторов, $\alpha_1\cdot\alpha_2$ — для скалярного произведения и α^2 — для $\alpha\cdot\alpha$. Каждому варианту $s'\in S^T$ поставим в соответствие двоичный вектор $x(s')$ в том же пространстве, имеющий в качестве своих компонент числа $x(s')(t,\ s(\{t\}\cup R(t)))$, определенные для всех $t\in T$ и $s(\{t\}\cup R(t))\in S^{\{t\}\cup R(t)}$, так, что компонента $x(s')(t,\ s(\{t\}\cup R(t)))$ равна единице, если $s'(\{t\}\cup R(t))=s(\{t\}\cup R(t))$, и равна нулю в противном случае. С учетом принятых обозначений запишем

$$\sum_{t\in T}\alpha(t,\ s(\{t\}\cup R(t)))=\alpha\cdot x(s).$$

Обозначим через Δ выпуклое замыкание множества

$$\{x(s)-x(s'):(s,\ T^H)\in\hat{S},s'\in\hat{S}'(s,\ T^H)\}.$$

Теорема 7. *Если система неравенств (4.72) имеет решение, то $\alpha^*=\arg\min_{\alpha\in\Delta}\alpha^2$ и есть такое решение.*

Доказательство. Прежде всего отметим, что из непротиворечивости системы неравенств (4.72) следует $\min_{\alpha\in\Delta}\alpha^2\neq 0$. Мы докажем, что если $\alpha^*=\arg\min_{\alpha\in\Delta}\alpha^2$, то для любой пары вариантов $(s,\ T^H)\in\hat{S}$ и $\hat{S}'\in S(s,\ T^H)$ справедливо неравенство

$$\alpha^*\cdot(x(s)-x(s'))\geqslant\alpha^{*2}>0. \tag{4.73}$$

Допустим, что неравенство (4.73) несправедливо, т. е. существует пара вариантов s, s', $(s,\ T^H)\in\hat{S}$ и $s'\in\hat{S}'(s,\ T^H)$, такая что

$$\alpha^{*2}>\alpha^*\cdot(x(s)-x(s')). \tag{4.74}$$

Для вещественного числа γ введем обозначение

$$F(\gamma)=(\alpha^*\cdot(1-\gamma)+(x(s)-x(s'))\cdot\gamma)^2$$

и убедимся в том, что функция F — непрерывна и, кроме того, $F(0)=\alpha^{*2}$, а $F'(0)=-2\alpha^{*2}+2(x(s)-x(s'))\cdot\alpha^*$. Правая часть последнего выражения по предположению (4.74) меньше нуля. Поэтому существует такое значение γ^*, $0<\gamma^*\leqslant 1$, что $(\alpha^*\cdot(1-\gamma^*)+(x(s)-x(s'))\cdot\gamma^*)^2<\alpha^{*2}$. Вектор α^* по предположению входит в Δ, как и вектор $(x(s)-x(s'))$. Следовательно, и вектор $\alpha^*\cdot(1-\gamma^*)+(x(s)-x(s'))\cdot\gamma^*$ также входит в Δ, причем длина его меньше, чем α^{*2}.

Мы доказали, таким образом, что для $\arg\min_{\alpha\in\Delta}\alpha^2$ выполняется (4.73), а следовательно, и (4.72). Теорема доказана.

Введем в рассмотрение некоторую функцию f, обладающую следующими свойствами. Пусть для $\alpha\in\Delta$ существует такая пара вариантов $s_1\in\hat{S}$ и $s_2\in\hat{S}(s_1,\ T^H)$, что выполняется неравенство $\alpha\cdot x(s_1)\leqslant\alpha\cdot x(s_2)$, т. е. нарушается одно из неравенств в системе (4.72). В таком случае $f(\alpha)$ определяется выражением $\alpha\cdot(1-\gamma_0)+(x(s_1)-x(s_2))\cdot\gamma_0$, где $\gamma_0=\arg\min_{\gamma}[\alpha\cdot(1-\gamma_0)+(x(s_1)-x(s_2))\cdot\gamma]$. Если же такой пары вариантов s_1 и s_2 не существует, то $f(\alpha)=\alpha$. Обозначим $f(\alpha)$ через $f^1(\alpha)$, а $f(f^1(\alpha))$ — через $f^2(\alpha)$ и вообще $f(f^{n-1}(\alpha))$ — через $f^n(\alpha)$. Естественно, сказанное не определяет однозначно функций f, f^1 и f^n, так как для заданного $\alpha\in\Delta$ могут существовать несколько различных пар s_1 и s_2, нарушающих систему неравенств (4.72), и, следовательно, по-разному может быть определено значение $f(\alpha)$. Однако независимо от того, как доопределить функцию f до однозначности, справедливой оказывается теорема, которая фактически определяет алгоритм решения задачи обучения III.

Теорема 8. *Если система неравенств (4.72) имеет решение, то для любого $\alpha \in \Delta$ существует такое целое число n, что $f^n(\alpha)$ есть решение системы неравенств (4.72).*

Доказательство. Обратимся к рис. 4.1, на котором представлено двумерное подпространство, содержащее векторы α, $x(s(\alpha)) - x(s'(\alpha))$ и $f(\alpha)$. Поскольку $(f(\alpha))^2 = \min\limits_{\gamma} (\alpha \cdot (1-\gamma) + (x(s(\alpha)) - x(s'(\alpha))) \cdot (\gamma))^2$, очевидно, что конец вектора $f(\alpha)$ является основанием перпендикуляра, опущенного из начала координат на линию, проходящую через концы векторов α и $(x(s(\alpha)) - x(s'(\alpha))$. Кроме того, поскольку по определению $\alpha \cdot (x(s(\alpha))) \leqslant \alpha \times x(s'(\alpha))$, угол между векторами α и $x(s(\alpha)) - x(s'(\alpha))$ — неострый. Не выходя за рамки элементарной геометрии, получаем $(f(\alpha))^2/\alpha^2 = w^2/(w^2+\alpha^2) = 1/(1+\alpha^2/w^2)$ и $w^2 \leqslant (x(s) - x(s'))^2$. Кроме того, $|x(s) - x(s')|^2 \leqslant 2|T|$, поэтому

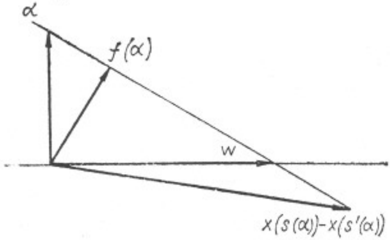


Рис. 4.1. К доказательству теоремы 8 гл. 4

$$(f(\alpha))^2/\alpha^2 \leqslant 1/(1+\alpha^2/(2|T|)).$$

Используя неравенство $\alpha^2 \geqslant \varepsilon$, преобразуем последнее соотношение к виду $(f(\alpha))^2/\alpha^2 \leqslant 1/(1+\varepsilon/(2|T|))$. Таким образом, для любого $\alpha \in \Delta$ последовательность α^2, $(f(\alpha))^2$, $(f^2(\alpha))^2$, ..., $(f^q(\alpha))^2$, ... меньше некоторой геометрической прогрессии с показателем $2|T|/(2|T|+\varepsilon)$. Следовательно, если только для некоторого значения α при всех $i \leqslant n$ $f^n(\alpha) \neq f^{n-1}(\alpha)$, то $(f^n(\alpha))^2 \leqslant \alpha^2 \cdot (2|T|/(2|T|+\varepsilon))^n$. Однако $(f^n(\alpha))^2$ не может быть меньше ε, поэтому условие $\alpha^2 \times (2|T|/2|T|+\varepsilon)^n = \varepsilon$ определяет максимально возможное количество итераций алгоритма, необходимое для нахождения решения, если перед началом первой итерации состояние алгоритма определялось набором α. Теорема доказана.

4.8. ЗАДАЧИ САМООБУЧЕНИЯ РАСПОЗНАВАНИЮ В ДВУМЕРНЫХ ГРАММАТИКАХ

1. Эвристические мотивировки. Рассмотренные в предыдущем параграфе процедуры обучения представляют собой уточнение модели, описывающей определенный объект, на основании экспериментального материала, представленного обучающей выборкой, т. е. множеством вариантов. При этом уже при формулировках задач обучения делалось молчаливое допущение, что на стадии формирования обучающей выборки доступны для измерения все параметры, включенные в модель, иными словами, допускалось, что в результате наблюдения экспериментатор получает вариант, т. е. исчерпывающие сведения о состоянии модели в момент наблюдения.

Задачи самообучения, анализу которых посвящен данный параграф, ставятся тогда, когда в каждый момент наблюдения измерению доступна лишь часть параметров, с помощью которых описывается модель. При этом в каком-то одном состоянии объекта может быть измерена одна часть параметров, в другом, вообще говоря, другая часть, а некоторая совокупность параметров, возможно, недоступна для измерения никогда. Это значит, что оценка параметров модели должна быть выполнена по выборке не вариантов, а фрагментов вариантов, и по последней следует определить зависимость наблюдаемых величин от некоторых ненаблюдаемых и, более того, зависимость даже между величинами, которые непосредственно не наблюдались никогда.

Здесь, естественно, должен возникнуть вопрос о принципиальной возможности такого рода процедур. Мы постараемся дать положительный ответ на этот вопрос, а пока лишь отметим, что процедуры подобного рода являются неотъемлемой частью интеллектуальной деятельности любого существа, претендующего на интеллектуальность, в частности искусственно созданного. Поэтому если иногда распознавание образов рассматривают как совокупность исследований, примыкающих к искусственному интеллекту, то именно проблематика самообучения распознаванию может служить тому подтверждением.

Мы рассмотрим пример, иллюстрирующий различие между задачами обучения и самообучения, и возможность разумной постановки последней. Предлагаемый ниже пример иллюстрирует лишь определенные эвристические соображения, а не практические рекомендации.

Пример. Допустим, что человек хочет сконструировать систему, понимающую его замысел по изображениям, которые он нарисовал. Допустим также, что замысел человека случаен и принимает лишь два значения с вероятностями

α_1 и α_2. Изображение, которое отражает замысел, также случайное и принимает значение x из огромного, но конечного множества X значений с вероятностью $p_1(x)$ при первом замысле и с вероятностью $p_2(x)$ при втором замысле. Если человеку известны вероятности α_1, α_2 и функции p_1, p_2, то он может сконструировать систему, имеющую входной канал для получения информации x и принимающую решение в пользу первого замысла, если $\alpha_1 \cdot p_1(x) > \alpha_2 \cdot p_2(x)$, и принимающую решение в пользу второго замысла в противном случае. Такая конструкция системы вполне разумна, а с некоторой точки зрения и оптимальна.

Если человеку неизвестны вероятности α_1, α_2 и функции p_1, p_2 хотя бы потому, что время от времени они меняются, то он может сконструировать систему, имеющую два входа: первый — для получения информации об x, а второй — для непосредственного сообщения системе величины $k \in \{1, 2\}$ о своем замысле. В результате подачи на вход системы достаточно длинной обучающей последовательности $k_1, x_1, k_2, x_2, \dots$ $\dots, k_i, x_i \dots$ система в принципе может восстановить вероятности α_1, α_2 и функции p_1, p_2, а затем их использовать для принятия разумного решения о замысле k человека, наблюдая лишь изображение x. Мы здесь, естественно, пренебрегаем тем обстоятельством, что при любой длине обучающей последовательности вероятности α_1, α_2 и функции p_1, p_2 будут восстановлены неточно, так как сейчас речь идет не об этом.

Допустим теперь, что по тем или иным причинам у человека нет возможности непосредственно сообщать системе о своем замысле k в период обучения хотя бы потому, что в системе не предусмотрен специальный канал для этого. Однако в системе имеется канал, по которому можно подавать на ее вход сообщение y, представляющее собой речевой сигнал из очень большого, но конечного множества Y.

Допустим, что речевой сигнал также случаен и имеет распределение вероятностей $q_1(y)$ при первом замысле и $q_2(y)$ — при втором. Кроме того, допустим, что существуют два идеальных речевых сигнала y_1 и y_2, таких, что $q_1(y_1) \neq 0$, $q_2(y_1) = 0$, $q_1(y_2) = 0$, $q_2(y_2) \neq 0$, хотя и априори неизвестных. И, наконец, предположим, что при наличии у человека i-го, $i = 1,2$, замысла он сообщает системе сигнал y и показывает системе изображение x с вероятностью $p_i(x) \cdot q_i(y)$.

Допустим, что на вход системы подается достаточно длинная последовательность $U = x_1, y_1, x_2, y_2, \dots, x_i, y_i, \dots$ Если бы система уже была обучена распознавать замысел k по речевому сигналу y, то последовательность U могла бы заменить обучающую последовательность и быть использованной для обучения распознаванию изображений.

В то же время если бы система уже была обучена распознавать замысел по изображению, то эта обученность могла бы быть использована для обучения распознаванию и по речевым сигналам. Мы же ставим вопрос о том, может ли система, наблюдающая последовательность U, обучиться распознаванию как изображений, так и речевых сигналов, если до наблюдения этой последовательности система не умела распознавать ни то, ни другое. В совершенно нестрогой форме — это вопрос о том, можно ли человеку или какому-либо другому существу объяснить, что означают слова на некотором непонятном ему языке, давая объяснение на другом, но тоже непонятном ему языке. Мы намерены дать на этот вопрос положительный ответ.

Наблюдая последовательность U, в принципе можно восстановить множество функций $\mathcal{P} = \{(p_y : X \to R) : y \in Y\}$, где $p_y(x)$ — условная вероятность изображения x на зрительном входе системы при условии, что на речевом входе имеется сигнал y. Здесь, как и прежде, мы пренебрегаем тем обстоятельством , что функции p_y даже при очень длинной последовательности U могут быть восстановлены лишь с определенной погрешностью. Множество $\{(p_y : X \to R) : y \in Y\}$ одномерно в том смысле, что для любых p', p'', p''', таких, что, $p' \in \mathcal{P}$, $p'' \in \mathcal{P}$, $p''' \in \mathcal{P}$, $p' \neq p''$, существует такое число γ, что $p''' = \gamma \cdot p' + (1 - \gamma) \cdot p''$. Это утверждение справедливо, так как для любого $y \in Y$ $p_y = \alpha_1(y) \cdot p_1 + \alpha_2(y) \cdot p_2$, где $\alpha_1(y)$ и $\alpha_2(y)$ — апостериорные вероятности событий $k = 1$ и $k = 2$ при наблюдении y:

$$\alpha_i(y) = \frac{\alpha_i \cdot q_i(y)}{\sum\limits_{i=1}^{2} \alpha_i \cdot q_i(y)}, \quad i = 1, 2.$$

В силу справедливости этого утверждения мы можем предложить следующую процедуру восстановления функций p_1 и p_2. Выберем две любые функции p' и p'' из $\mathcal{P}$, такие, что $p' \neq p''$, и затем упорядочим все функции из $\mathcal{P}$ таким образом, что если $p^* = \gamma^* \cdot p' + (1 - \gamma^*) \cdot p''$, а $p^{**} = \gamma^{**} p' + (1 - \gamma^{**}) \cdot p''$ и $\gamma^* < \gamma^{**}$, то функция p^* будет считаться левее функции p^{**}. Найдем самую левую функцию из $\mathcal{P}$ и самую правую функцию из $\mathcal{P}$. Одна из них будет равна p_1, а другая — p_2. Зная функции p_1 и p_2, мы можем восстановить числа α_1 и α_2 — вероятности событий $k = 1$ и $k = 2$. Для этого прежде всего нужно вычислить функцию $p_{ср} = \frac{1}{N} \sum\limits_{i=1}^{N} p_{y_i}$, где y_i — i-й элемент в обучающей последовательности U, а N — ее длина. Для $p_{ср}$ при достаточно большом значении N справедливо равенство $p_{\text{ср}} = \alpha_1 \cdot p_1 + \alpha_2 \cdot p_2$. По этому равенству, зная функции p_1 и p_2 (мы их уже восстановили) и зная, что $\alpha_1 + \alpha_2 = 1$, мы легко определим α_1 и α_2.

Для того чтобы восстановить функции q_1 и q_2, рассмотрим множество функций $Q = \{(Q_x : y \to R) : x \in X\}$, где $Q_x(y)$ — условная вероятность речевого сигнала y на входе системы при условии, что на зрительном входе имеется изображение x. Множество Q также является одномерным в том же смысле, что и множество $\mathcal{P}$.

Рассмотрим два изображения x_1 и x_2, таких, что $Q_{x_1} \neq Q_{x_2}$. Справедлива следующая система уравнений:

$$\begin{aligned} Q_{x_1} &= \alpha_1(x_1) \cdot q_1 + \alpha_2(x_1) \cdot q_2, \\ Q_{x_2} &= \alpha_1(x_2) \cdot q_1 + \alpha_2(x_2) \cdot q_2, \end{aligned} \tag{4.75}$$

где $\alpha_j(x_j)$ апостериорная вероятность события $k = i$ при наблюдении x_j, которую мы можем вычислить, так как α_1 и α_2, p_1 и p_2 уже восстановлены.

В системе уравнений (4.75) левые части известны. Эта система имеет решение относительно функций q_1 и q_2, так как $\alpha_1(x_1) \neq \alpha_1(x_2)$). Поэтому $\alpha_1(x_1) \cdot \alpha_2(x_2) \neq \alpha_1(x_2) \cdot \alpha_2(x_1)$.

После того как восстановлены вероятности α_1, α_2 и функции p_1, p_2 и q_1, q_2, можно принять разумное решение k_i для каждого элемента x_i, y_i из последовательности U. Например, если $\alpha_1 \cdot p_1(x_i) \cdot q_1(y_i) > \alpha_2 \cdot p_2(x_i) \cdot q_2(x_i)$, то $k_i = 1$, в противном случае $k_i = 2$. Описание примера закончено.

Конкретизация высказанных эвристических соображений до уровня недвусмысленной формулировки задачи, равно как и анализ этой задачи, существенно выходит за рамки проблематики обработки изображений и двумерных грамматик, являющихся основным предметом исследования. Поэтому в следующем параграфе мы рассмотрим в общей формулировке основные теоремы о самообучении, имеющие самостоятельное значение, и лишь затем применим их для решения задач самообучения применительно к двумерным грамматикам.

2. Три теоремы о самообучении. Пусть X — некоторое абстрактное конечное множество, называемое множеством изображений; K — конечное множество образов. Будем считать, что каждому образу $k \in K$ соответствует некоторое распределение вероятностей $P_k : X \to R$ на множестве X, а также допустим существование некоторого распределения вероятностей $Q : K \to R$ образов. Функции P_k, $k \in K$, и Q априори неизвестны. Известны, однако, множества $\mathcal{P}_k$, $k \in K$, такие, что $P_k \in \mathcal{P}_k$, $k \in K$. Множества $\mathcal{P}_k$, $k \in K$, не обязательно разные.

Допустим, что задана выборка $U = (k_1, x_1), (k_2, x_2), \dots, (k_m, x_m)$ случайных независимых пар (k_i, x_i), имеющих распределение вероятностей $Q(k_i) \cdot P_k(x_i)$.

Задачей обучения по выборке $U = (k_1, x_1), (k_2, x_2), \dots, (k_m, x_m)$ мы будем называть отыскание таких функций Q и P_k, $k \in K$, которые максимизируют вероятность выборки U, т. е. максимизируют функционал

$$\sum_{i=1}^{m} \log Q(k_i) \cdot P_{k_i}(x_i) \tag{4.76}$$

при условиях $P_k \in \mathcal{P}_k$, $k \in K$, $\sum_{k \in K} Q(k) = 1$. Представим выборку U в виде множества $\hat{X}$ и функции $\alpha : (\hat{X} \times K) \to N$ (N — множество целых чисел), так, чтобы изображение $x \in X$ входило в множество $\hat{X}$ тогда и только тогда, когда x входит в выборку U, а число $\alpha(x, k)$ равнялось количеству вхождений пары (x, k) в выборку U. Очевидно, что максимизация (4.76) в этом случае сводится к максимизации

$$\sum_{k \in K} \sum_{x \in \hat{X}} \alpha(x, k) \log Q(k) \cdot P_k(x),$$

которая вырождается в независимые максимизации $|K|$ функционалов

$$\sum_{x \in \hat{X}} \alpha(x, k) \log P_k(x) \tag{4.77}$$

при условии $P_k \in \mathcal{P}$, $k \in K$, и функционала

$$\sum_{k \in K} \sum_{x \in X} \alpha(x, k) \log Q(k) \tag{4.78}$$

при условии $\sum_{k \in K} Q(k) = 1$.

Пусть теперь задана выборка $U = x_1, x_2, \dots, x_m$ случайных независимых изображений, имеющих распределение вероятностей $\sum_{k \in K} Q(k) \cdot P_k(x)$. От прежде рассмотренной выборки она отличается тем, что в ней отсутствуют сведения о номере образа, которому соответствует изображение x_i, $i = 1, 2, \dots, m$.

Задачей самообучения по выборке $U = (x_1, x_2, \dots, x_m)$ будем называть отыскание таких функций Q и P_k, $k \in K$, которые максимизируют вероятность выборки U, т. е. максимизируют функционал

$$\mathcal{L}(A) = \sum_{i=1}^{m} \log \sum_{k \in K} Q(k) P_k(x_i) \tag{4.79}$$

при условии $P_k \in \mathcal{P}_k$, $k \in K$, $\sum_{k \in K} Q(k) = 1$. В выражении (4.79) через A обозначена совокупность функций Q и P_k, $k \in K$.

Покажем, что если известен алгоритм обучения, т. е. алгоритм отыскания максимума функционалов (4.77) и (4.78), то можно сформулировать и алгоритм самообучения, т. е. максимизации (4.79), причем алгоритм самообучения будет включать в себя алгоритм обучения как подпрограмму.

Предлагаемый нами алгоритм самообучения заключается в следующем. Пусть $A^0 = (Q^0, P_k^0, k \in K)$ — некоторые начальные значения искомых функций, обладающие тем свойством, что для любого изображения x_i из обучающей выборки по крайней мере для одного $k \in K$ вероятность $Q(k) \cdot P_k(x_i)$ не равна нулю.

Алгоритм самообучения представляет собой получение последовательности $A^0, A^1, \ldots, A^j, \ldots$ Одна итерация, т. е. получение функций $A^{(j+1)} = (Q^{(j+1)}, P_k^{(j+1)}, k \in K)$ из функций $A^{(j)} = (Q^{(j)}, P_k^{(j)}, k \in K)$, выполняется в два этапа.

На первом этапе выполняется как бы распознавание изображений из обучающей выборки, т. е. вычисление величин

$$\alpha^{(j)}(x_i, k) = \frac{Q^{(j)}(k) \cdot P_k^{(j)}(x_i)}{\sum_{k \in K} Q^{(j)}(k) \cdot P_k^{(j)}(x_i)} \tag{4.80}$$

для всех $k \in K$ и для всех x_i из обучающей выборки.

Величины $\alpha^{(j)}(x_i, k)$ подобны апостериорным вероятностям принадлежности изображения x_i к классу k, однако не являются таковыми, поскольку распределения вероятностей $Q^{(j)}$ и $P_k^{(j)}$ совсем не обязательно равны действительным распределениям. И, тем не менее, на втором этапе должна быть выполнена процедура обучения, т. е. максимизация функционалов

$$\sum_{i=1}^{m} \alpha^{(j)}(x_i, k) \log P_k(x_i) \tag{4.81}$$

для всех $k \in K$ при условии $P_k \in \mathcal{P}$ и функционала

$$\sum_{k \in K} \sum_{i=1}^{m} \alpha^{(j)}(x_i, k) \log Q(k) \tag{4.82}$$

при условии $\sum_{k \in K} Q(k) = 1$.

Функции P_k^*, $k \in K$, и Q, максимизирующие соответственно (4.81) и (4.82), принимаются в качестве функций $P_k^{(j+1)}$ и

$Q^{(j+1)}$ и служат исходными для следующей итерации алгоритма.

В приведенном алгоритме выражена глубокая взаимосвязь между тремя очень обширными подклассами задач распознавания, а именно задач собственно распознавания, обучения и самообучения. Как мы увидим далее из обоснования приведенного алгоритма, эта взаимосвязь справедлива практически при любом виде распределений P_k, Q и множеств $\mathcal{P}_k$. Если еще учесть, что эта взаимосвязь выражена в краткой и легко запоминающейся форме, то, конечно, обнаружение ее следует считать, наряду с алгоритмом вычеркивания и разработкой двумерных грамматик, одним из основных результатов данной работы.

Для обоснования алгоритма нам необходимо использовать свойства функции $\sum_{i=1}^{n} \alpha_i \log P_i$, которые сформулированы в следующей лемме.

Лемма 3. *Пусть α_i, $i = 1, 2, \ldots, n$, — положительные числа; P_i, $i = 1, 2, \ldots, n$,— положительные числа, такие, что $\sum_{i=1}^{n} P_i = 1$. В таком случае $\sum_{i=1}^{n} \alpha_i \log \frac{\alpha_i}{\sum_{i=1}^{n} \alpha_i} \geqslant \sum_{i=1}^{n} \alpha_i \log P_i$, причем равенство выполняется лишь тогда, когда для всех $i = 1, 2, \ldots, n$ выполняется равенство $P_i = \frac{\alpha_i}{\sum_{i=1}^{n} \alpha_i}$.*

Доказательство. Максимум функции $\sum_{i=1}^{n} \alpha_i \times \log P_i$ по переменным $P_1, P_2, \ldots, P_n$ при $\sum_{i=1}^{n} P_i = 1$ достигается при величинах $P_1^*, P_2^*, \ldots, P_n^*$, являющихся решением системы уравнений

$$\frac{\alpha_i}{P_i} - \lambda = 0, \quad i = 1, 2, \ldots, n,$$

$$\sum_{i=1}^{n} P_i = 1.$$

Эта система имеет следующее единственное решение:

$$P_i^* = \frac{\alpha_i}{\sum_{i=1}^{n} \alpha_i}, \quad i = 1, 2, \ldots, n.$$

Лемма доказана.

Обозначим через $\mathscr{L}(A)$ максимизируемый функционал

$$\sum_{i=1}^{m} \log \sum_{k\in K} Q(k) P_k(x_i).$$

Теорема 9. *Пусть $Q^{(j)}$, $P_k^{(j)}$, $k\in K$, и $Q^{(j+1)}$, $P_k^{(j+1)}$, $k\in K$,— два набора функций, получаемые в результате $(j-1)$-й и j-й итераций алгоритма самообучения. В таком случае*

$$\sum_{i=1}^{m} \log \sum_{k\in K} Q^{(j+1)}(k)\cdot P_k^{(j+1)}(x_i) \geqslant \sum_{i=1}^{m} \log \sum_{k\in K} Q^{(j)}(k) P_k^{(j)}(x_i). \quad (4.83)$$

При этом если хотя бы для одного $i=1, 2, \ldots, m$ и для одного $k\in K$ выполняется неравенство

$$\frac{Q^{(j+1)}(k)\cdot P_k^{(j+1)}(x_i)}{\sum_{k\in K} Q^{(j+1)}(k)\cdot P_k^{(j+1)}(x_i)} \neq \frac{Q^{(j)}(k)\cdot P_k^{(j)}(x_i)}{\sum_{k\in K} Q^{(j)}(k)\cdot P_k^{(j)}(x_i)},$$

то неравенство (4.83) выполняется строго.

Доказательство. Поскольку в соответствии с (4.80) для всех i и j справедливо равенство $\sum_{k\in K}\alpha^{(j)}(x_i, k) = 1$, выражение для $\mathscr{L}(A)$ можно записать в виде

$$\mathscr{L}(A^{(j)}) = \sum_{i=1}^{m} \log \sum_{k\in K} Q^{(j)}(k) P_k^{(j)}(x_i) =$$

$$= \sum_{k\in K}\sum_{i=1}^{m} \alpha^{(j)}(x_i, k) \log \sum_{k\in K} Q^{(j)}(k) P_k^{(j)}(x_i) =$$

$$= \sum_{k\in K}\sum_{i=1}^{m} \alpha^{(j)}(x_i, k)\log Q^{(j)}(k) + \sum_{k\in K}\sum_{i=1}^{m} \alpha^{(j)}(x_i, k)\log P_k^{(j)}(x_i) -$$

$$- \sum_{i=1}^{m}\sum_{k\in K} \alpha^{(j)}(x_i, k) \log \frac{Q^{(j)}(k) P_k^{(j)}(x_i)}{\sum_{k\in K} Q^{(j)}(k)\cdot P_k^{(j)}(x_i)} \quad (4.84)$$

Аналогично можно записать выражение для $\mathscr{L}(A^{(j+1)})$:

$$\mathscr{L}(A^{(j+1)}) = \sum_{k\in K}\sum_{i=1}^{m} \alpha^{(j)}(x_i, k)\log Q^{(j+1)}(k) +$$

$$+ \sum_{k\in K}\sum_{i=1}^{m} \alpha^{(j)}(x_i, k)\log P_k^{(j+1)}(x_i) -$$

$$- \sum_{i=1}^{m}\sum_{k\in K} \alpha^{(j)}(x_i, k) \log \frac{Q^{(j+k)}(k) P_k^{(j+1)}(x_i)}{\sum_{k\in K} Q^{(j+1)}(k) P_k^{(j+1)}(x_i)} \quad (4.85)$$

Справедливы следующие неравенства

$$\sum_{i=1}^{m} \alpha^{(j)}(x_i, k)\log P_k^{(j)}(x_i) \leqslant \sum_{i=1}^{m} \alpha^{(j)}(x_i, k)\log P_k^{(j+1)}(x_i),\ k\in K; \quad (4.86)$$

$$\sum_{k\in K}\sum_{i=1}^{m} \alpha^{(j)}(x_i, k)\log Q^{(j)}(k) \leqslant \sum_{k\in K}\sum_{i=1}^{m} \alpha^{(j)}(x_i, k)\log Q^{(j+1)}(k); \quad (4.87)$$

$$\sum_{k\in K} \alpha^{(j)}(x_i, k)\log \frac{Q^{(j)}(k)\cdot P_k^{(j)}(x_i)}{\sum_{k\in K} Q^{(j)}(k)\cdot P_k^{(j)}(x_i)} \geqslant$$

$$\geqslant \sum_{k\in K} \alpha^{(j)}(x_i, k)\log \frac{Q^{(j)}(k)\cdot P_k^{(j)}(x_i)}{\sum_{k\in K} Q^{(j)}(k)\cdot P_k^{(j)}(x_i)}, \quad i = 1, 2, \ldots, m. \quad (4.88)$$

Неравенства (4.86) и (4.87) справедливы по определению второго этапа j-й итерации алгоритма, потому что на этом этапе происходит выбор в качестве $Q^{(j+1)}$ и $P_k^{(j+1)}$ именно таких функций, которые максимизируют (4.82) и (4.81).

Неравенства (4.88) справедливы по определению первого этапа j-й итерации алгоритма, так как величины $\alpha^{(j)}(x_i, k)$ выбираются равными $\frac{Q^{(j)}(k) P_k^{(j)}(x_i)}{\sum_{k\in K} Q^{(j)}(k) P_k^{(j)}(x_i)}$. Поэтому в силу леммы 3 неравенства (4.88) выполняются, причем если хотя бы для одного $i = 1, 2, \ldots, m$; $k\in K$ справедливо неравенство

$$\frac{Q^{(j+1)}(k)\cdot P_k^{(j+1)}(x_i)}{\sum_{k\in K} Q^{(j+1)}(k)\cdot P_k^{(j+1)}(x_i)} \neq \alpha^{(j)}(x_i, k),$$

то хотя бы одно из неравенств (4.88) выполняется строго.

Из неравенств (4.86) следует, что

$$\sum_{k\in K}\sum_{i=1}^{m} \alpha^{(j)}(x_i, k)\log P_k^{(j)}(x_i) \leqslant \sum_{k\in K}\sum_{i=1}^{m} \alpha^{(j)}(x_i, k)\log P_k^{(j+1)}(x_i), \quad (4.89)$$

а из неравенств (4.88),— что

$$\sum_{i=1}^{m}\sum_{k\in K} \alpha^{(j)}(x_i, k)\log \frac{Q^{(j)}(k)\cdot P_k^{(j)}(x_i)}{\sum_{k\in K} Q^{(j)}(k)\cdot P_k^{(j)}(x_i)} \geqslant$$

$$\geqslant \sum_{i=1}^{m}\sum_{k\in K} \alpha^{(j)}(x_i, k)\log \frac{Q^{(j+1)}(k)\cdot P_k^{(j+1)}(x_i)}{\sum_{k\in K} Q^{(j+1)}(k)\cdot P_k^{(j+1)}(x_i)}, \quad (4.90)$$

с учетом условий, когда неравенство (4.90) выполняется строго.

Из неравенств (4.87), (4.89), (4.90) и с учетом (4.84), (4.85) сразу же следует неравенство $\mathcal{L}(A^{(j+1)}) \geqslant \mathcal{L}(A^{(j)})$ и справедливость условий, указанных в теореме, при которых $\mathcal{L}(A^{(j+1)}) > \mathcal{L}(A^{(j)})$. Теорема доказана.

Алгоритмы самообучения, доказанные только что и ниже теоремы, становятся достаточно ясными при рассмотрении наглядного описания, приведенного на рис. 4.2.

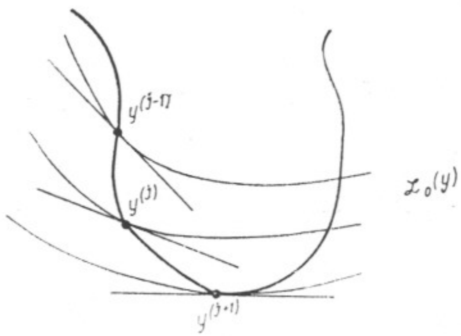


Рис. 4.2. Интерпретация алгоритма самообучения

Отобразим множество всех возможных функций вида $Q \cdot P_k$ на $|K| \times m$-мерное евклидово пространство. Набору функций $(Q,\ P_k,\ k \in K)$ соответствует в этом пространстве точка y с координатами y_{ik}, определяемыми формулой

$$y_{ik} = \log Q(k) \cdot P_k(x_i).$$

Определим функцию $\mathcal{L}_0 : R^{|k| \times m} \to R$ так, что

$$\mathcal{L}_0(y) = \sum_{i=1}^{m} \log \sum_{k \in K} Q(k) \cdot P_k(x_i) = \sum_{i=1}^{m} \log \sum_{k \in K} e^{y_{ik}}.$$

На рисунке тонкими кривыми представлены так называемые линии уровня, т. е. множества точек y, для которых $\mathcal{L}_0(y) = \theta$. Разным линиям соответствуют разные значения θ. Линии уровня нарисованы так, что каждая из них делит поле рисунка на две области, одна из которых является выпуклой, и тому есть свои причины, так как для любого значения θ множество $\{y : \mathcal{L}(y) \leqslant \theta\}$ является выпуклым.

Чтобы доказать это, рассмотрим некоторую точку y^*, такую, что $\mathcal{L}_0(y^*) = \theta$, и функцию $C(y) = (y - y^*) \times \operatorname{grad}_{y=y^*} \mathcal{L}_0(y) + \theta$. В точке $y = y^*$ функции C и $\mathcal{L}_0$ равны между собой, так как $C(y^*) = \theta$, $\mathcal{L}_0(y^*) = \theta$. Множество точек y, для которых $C(y) = \theta$, представлено на рисунке прямой. Очевидно, что это касательная к линии уровня θ в точке y^*. Эта прямая делит плоскость чертежа на две части, в одной из которых $C(y) < 0$, а в другой $C(y) > 0$. Мы докажем, что множество $\{y : \mathcal{L}_0(y) \leqslant \theta\}$ выпукло, если докажем, что для любой точки y^* на линии уровня θ вся линия лежит по одну сторону от касательной к этой линии в точке y^*, или, иными словами, что из неравенства $\mathcal{L}_0(y) \leqslant \theta$ следует $C(y) \leqslant \theta$.

Градиент функции $\mathcal{L}_0(y)$ в точке y^* есть вектор с компонентами $\alpha_{ik} = \dfrac{e^{y_{ik}}}{\sum_k e^{y_{ik}}}$. Так как $\sum_{k \in K} \alpha_{ik} = 1$, а все $\alpha_{ik} \geqslant 0$, то пользуясь тем же приемом, что и при доказательстве теоремы 9, неравенство $\mathcal{L}_0(y) \leqslant \theta$, т. е.

$$\sum_{i=1}^{m} \log \sum_{k \in K} e^{y_{ik}} \leqslant \sum_{i=1}^{m} \log \sum_{k \in K} e^{y^*_{ik}}, \tag{4.91}$$

можно записать в виде

$$\sum_{k \in K} \sum_{i=1}^{m} \alpha_{ik} \cdot y_{ik} - \sum_{i=1}^{m} \sum_{k \in K} \alpha_{ik} \log \frac{e^{y_{ik}}}{\sum_{k \in K} e^{y_{ik}}} \leqslant$$

$$\leqslant \sum_{k \in K} \sum_{i=1}^{m} \alpha_{ik} \cdot y^*_{ik} - \sum_{i=1}^{m} \sum_{k \in K} \alpha_{ik} \log \alpha_{ik}; \quad \alpha_{ik} = \frac{e^{y^*_{ik}}}{\sum_{k \in K} e^{y^*_{ik}}}. \tag{4.92}$$

В силу леммы справедливо неравенство $\sum_{i=1}^{m} \sum_{k \in K} \alpha_{ik} \log \alpha_{ik} \geqslant \sum_{i=1}^{m} \sum_{k \in K} \alpha_{ik} \log \dfrac{e^{y_{ik}}}{\sum_{k \in K} e^{y_{ik}}}$, поэтому из (4.92) следует $\sum_{k \in K} \sum_{i=1}^{m} (y_{ik} - y^*_{ik}) \cdot \alpha_{ik} \leqslant 0$. Запишем последнее неравенство в векторной форме $(y - y^*) \cdot \operatorname{grad}_{(y=y^*)} \mathcal{L}_0(y) \leqslant 0$ и получим, таким образом, что $C(y) \leqslant \theta$, т. е. множество $\{y : \mathcal{L}_0(y) \leqslant \theta\}$ выпукло. На рисунке утолщенной линией представлено также множество Y — образ множества A допустимых функций $Q \cdot P_k$.

Одна итерация алгоритма в приведенном представлении заключается в следующем.

Пусть к началу j-й итерации имелась точка $y^{(j-1)}$. Первый этап итерации заключается в построении касательной к линии уровня в точке $y^{(j-1)}$, второй — в отыскании точки $y^{(j)} \in Y$, расположенной в «положительную» сторону от касательной и максимально удаленной от нее. На рисунке представлены точки $y^{(j-1)}$, $y^{(j)}$ и $y^{(j+1)}$. Последняя точка, по-видимому, неподвижна, т. е. такая, что $y^{(j+2)}$ окажется равным $y^{(j+1)}$. Весьма правдоподобно, что последовательность точек, строящаяся таким способом, имеет предел, однако, вообще, это не так. В действительности процесс самообучения сходится при определенных условиях, причем условия сходимости последовательности $A^1, A^2, \ldots, A^{(j)}, \ldots$ более жесткие, чем условия сходимости апостериорных вероятностей $\alpha^{(j)}(x_i, k)$, $i = 1, 2, \ldots, m$; $k \in K$, хотя с точки зрения практики ни те, ни другие условия не слишком обременительны.

Совокупность чисел $\alpha(x_i, k)$, $i = 1, 2, \ldots, m$; $k \in K$, будем рассматривать как вектор в $|k| \times m$-мерном нормированном пространстве с нормой $\sqrt{\sum_{i=1}^{m}\sum_{k\in K}(\alpha(x_i, k))^2}$. Этот вектор будем обозначать через α, возможно, с верхним индексом. Через $\mathcal{L}_0(\alpha)$ обозначим число

$$\max_{A\in\mathcal{A}}\left[\sum_{k\in K}\sum_{i=1}^{m}\alpha(x_i, k)\log Q(k)\cdot P_k(x_i)\right] - \sum_{i=1}^{m}\sum_{k\in K}\alpha(x_i, k)\times$$
$$\times\log\alpha(x_i, k).$$

Через $F(\alpha)$ обозначим вектор α', получаемый из α в результате одной итерации алгоритма самообучения. Решение уравнения $\alpha = F(\alpha)$ будем называть неподвижной точкой алгоритма.

Теорема 10. *Если $\mathcal{L}_0(\alpha)$ есть непрерывная функция от α, а количество неподвижных точек конечно, то для любого начального значения $\alpha^{(1)}$ существует неподвижная точка α^*, такая, что последовательность $\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(j)}, \ldots$, получаемая алгоритмом самообучения, имеет предел α^*.*

Доказательство. 1. Поскольку последовательность $\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(j)}, \ldots$ является бесконечной последовательностью в ограниченной области конечномерного пространства, существует бесконечная подпоследовательность $\alpha^{(j(1))}$ $\alpha^{(j(2))}, \ldots, \alpha^{(j(t))}, \ldots$, такая, что $j(t) > j(t-1)$, и имеющая предел. Обоначим этот предел через α'. Докажем, что α' есть неподвижная точка алгоритма.

Последовательность чисел $\mathcal{L}_0(\alpha^{(1)})$, $\mathcal{L}_0(\alpha^{(2)})$, ..., $\mathcal{L}_0(\alpha^{(j)})$, ... в силу теоремы 9 — монотонно неубывающая, причем ограниченная сверху, так как $\mathcal{L}_0(\alpha) \leqslant 0$ для всех α. Поэтому она имеет предел, который мы обозначим через $\mathcal{L}^*$. Очевидно, что $\mathcal{L}^*$ является пределом и последовательности $\mathcal{L}_0(\alpha^{(j(t))})$, $t = 1, 2, \ldots$, и последовательности $\mathcal{L}_0(\alpha^{(j(t)+1)})$, $t = 1, 2, \ldots$ Сказанное запишем в виде трех равенств:

$$\lim_{t\to\infty}\mathcal{L}_0(\alpha^{(j(t))}) = \mathcal{L}^*, \tag{4.93}$$

$$\lim_{t\to\infty}\mathcal{L}_0(\alpha^{(j(t)+1)}) = \mathcal{L}^*, \tag{4.94}$$

$$\lim_{t\to\infty}\alpha^{(j(t))} = \alpha'. \tag{4.95}$$

С учетом непрерывности $\mathcal{L}_0(\alpha)$ по α равенства (4.93) и (4.94) запишем в виде $\lim\limits_{t\to\infty}\mathcal{L}_0(\alpha^{(j(t))}) = \mathcal{L}_0(\alpha')$ и $\lim\limits_{t\to\infty}\mathcal{L}_0(\alpha^{(j(t)+1)}) = \mathcal{L}_0(F(\alpha'))$, откуда следует, что $\mathcal{L}_0(\alpha') = \mathcal{L}_0(F(\alpha'))$. В силу теоремы 9 это равенство возможно лишь в том случае, если $\alpha' = F(\alpha')$, так как при $\alpha' \neq F(\alpha')$ выполнимо неравенство $\mathcal{L}(\alpha') < \mathcal{L}(F(\alpha'))$.

2. Пусть Ω — множество неподвижных точек алгоритма. Докажем, что $\lim\limits_{j\to\infty}\min\limits_{\alpha'\in\Omega}|\alpha^{(j)} - \alpha'| = 0$, т. е. что

$$\forall\,\varepsilon > 0\,(\exists N\,(\forall j > N\,(\min_{\alpha'\in\Omega}|\alpha^{(j)} - \alpha'| < \varepsilon))).$$

Допустим, что это не так, т. е. что справедливо неравенство

$$\exists\,\varepsilon > 0\,(\forall N\,(\exists j > N\,(\min_{\alpha'\in\Omega}|\alpha^{(j)} - \alpha'| \geqslant \varepsilon))).$$

Из последнего утверждения следует существование такой бесконечной последовательности $\alpha^{(j(1))}, \alpha^{(j(2))}, \ldots, \alpha^{(j(t))}, \ldots$, что для всех t и $\alpha' \in \Omega$ справедливо неравенство $|\alpha^{(j(t)} - \alpha'| \geqslant \varepsilon$. В этой последовательности можно выделить сходящуюся подпоследовательность, предел которой α', очевидно, не принадлежит Ω. Однако это невозможно в силу доказанного в п. 1. Таким образом, имеет место равенство

$$\lim_{j\to\infty}\min_{\alpha'\in\Omega}|\alpha^{(j)} - \alpha'| = 0. \tag{4.96}$$

3. По лемме известно, что $\sum\limits_{k\in K}\alpha_k\log\frac{\alpha_k}{x_k} \geqslant 0$ для всех $0 \leqslant x_k$, $k \in K$, таких, что $\sum\limits_{k\in K}x_k = 1$. Докажем теперь, что к тому же

$$\sum_{k\in K}\alpha_k\log\frac{\alpha_k}{x_k} \geqslant \frac{1}{2}\sum_{k\in K}(\alpha_k - x_k)^2. \tag{4.97}$$

Введем обозначение $\Delta x_k = x_k - \alpha_k$, $k \in K$, и рассмотрим две функции, $\varphi(\gamma) = \sum\limits_{k\in K}\alpha_k\log\frac{\alpha_k}{\alpha_k + \gamma\Delta x_k}$, $\psi(\gamma) = \sum\limits_{k\in K}(\gamma\cdot\Delta x_k)^2$, зависящие от переменной γ, $0 \leqslant \gamma \leqslant 1$. Понятно, что $\varphi(0) = \psi(0) = 0$ и что при $0 \leqslant \gamma \leqslant 1$ для всех $k \in K$ выполняется неравенство $0 \leqslant \alpha_k + \gamma\cdot\Delta x_k \leqslant 1$. Производные по γ для введенных функций будут: $\frac{d\varphi}{d\gamma} = -\sum\limits_{k\in K}\alpha_k\frac{\Delta x_k}{\alpha_k + \gamma\cdot\Delta x_k}$, $\frac{d\psi}{d\gamma} = 2\cdot\sum\limits_{k\in K}\gamma(\Delta x_k)^2$.

Поскольку $\sum\limits_{k\in K}\Delta x_k = 0$, имеем $\frac{d\varphi}{d\gamma} = \frac{d\varphi}{d\gamma} + \sum\limits_{k\in K}\Delta x_k$, и можно записать новое выражение для производной в виде $\frac{d\varphi}{d\gamma} =$

$= \sum_{k \in K} \frac{\gamma \cdot (\Delta x_k)^2}{\alpha_k + \gamma \Delta x_k}$. Знаменатель в каждом слагаемом в этом выражении не превышает 1 при любом значении γ из интервала $0 \leqslant \gamma \leqslant 1$, поэтому при любом значении γ из этого интервала справедливо неравенство $2 \frac{d\varphi}{d\gamma} \geqslant \frac{d\psi}{d\gamma}$. Поскольку $2 \cdot \varphi(0) = \psi(0)$, из последнего неравенства следует, что при любом значении γ, в том числе и при $\gamma = 1$, справедливо неравенство $2 \cdot \varphi(\gamma) \geqslant \psi(\gamma)$. Мы имеем, таким образом, $2 \cdot \sum_{k \in K} \alpha_k \log \frac{\alpha_k}{\alpha_k + \Delta x_k} \geqslant \sum_{k \in K} (\Delta x_k)^2$, что является лишь иной формой записи неравенства (4.97).

4. Поскольку $\lim_{j \to \infty} \mathscr{L}_0(\alpha^{(j)}) = \mathscr{L}^*$, имеем

$$\lim_{j \to \infty} (\mathscr{L}_0(\alpha^{(j)}) - \mathscr{L}_0(\alpha^{(j+1)})) = 0. \tag{4.98}$$

При доказательстве теоремы 9 было выяснено, что

$$\mathscr{L}_0(\alpha^{(j+1)}) - \mathscr{L}_0(\alpha^{(j)}) \geqslant \sum_{i=1}^{m} \sum_{k \in K} \alpha^{(j)}(x_i, k) \log \frac{\alpha^{(j)}(x_i, k)}{\alpha^{(j+1)}(x_i, k)}. \tag{4.99}$$

Из неравенств (4.97), (4.99) и равенства (4.98) следует

$$\lim_{j \to \infty} |\alpha^{(j)} - \alpha^{(j+1)}| = 0. \tag{4.100}$$

5. Введем обозначение $\Delta = \min_{\alpha' \in \Omega} \min_{\alpha'' \in \Omega \setminus \{\alpha'\}} |\alpha' - \alpha''|$, где Ω — множество неподвижных точек алгоритма. Поскольку Ω по предположению содержит конечное количество точек, имеем $\Delta > 0$.

Докажем, что в последовательности $\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(j)}, \ldots$ есть лишь конечное количество номеров j, таких, что $\arg \min_{\alpha' \in \Omega} |\alpha^{(j)} - \alpha'| \neq \arg \min_{\alpha' \in \Omega} |\alpha^{(j+1)} - \alpha'|$. Это действительно так, потому что в противном случае в силу (4.96) для некоторого $\delta > 0$, $\delta < \Delta/2$ существовало бы бесконечное количество номеров j таких, что $|\alpha^{(j)} - \alpha'| < \delta$, $|\alpha^{(j+1)} - \alpha''| < \delta$, $\alpha' \in \Omega$, $\alpha'' \in \Omega$, а $\alpha' \neq \alpha''$. Отсюда сдедовало бы существование бесконечного количества номеров j, для которых, $|\alpha^{(j)} - \alpha^{(j+1)}| > \Delta - 2\delta > 0$, что невозможно в силу (4.100). Таким образом, последовательность $\arg \min_{\alpha' \in \Omega} |\alpha^{(j)} - \alpha'|$, $j = 1, 2, \ldots$, имеет предел, который мы обозначим через

$$\alpha^* = \lim_{j \to \infty} \arg \min_{\alpha' \in \Omega} |\alpha^{(j)} - \alpha'|. \tag{4.101}$$

Подставляя (4.101) в доказанное ранее (4.96), получаем $\lim_{j \to \infty} |\alpha^{(j)} - \alpha^*| = 0$, или, что то же, $\lim_{j \to \infty} \alpha^{(j)} = \alpha^*$, $\alpha^* \in \Omega$. Теорема доказана.

Из доказанного существования предела для последовательности $\alpha^{(j)}$, $j = 1, 2, \ldots$, конечно, не вытекает, что последовательность $A^{(j)}$, $j = 1, 2, \ldots$, искомых функций имеет предел, однако это уже не имеет большого значения. Действительно, в каком-либо прагматическом смысле функции Q и P_k, $k \in K$, нужны только для того, чтобы в конечном итоге классифицировать выборку, а эта классификация зависит только от α. Поэтому если для двух разных наборов функций $A^1 = (Q^1, P_k^1, k \in K)$, и $A^2 = (Q^2, P_k^2, k \in K)$, соответствующие апостериорные вероятности α^1 и α^2 равны, то и наборы $A^{(1)}$ и $A^{(2)}$ можно считать эквивалентными.

В то же время ясно, что последовательность $A^{(j)}$, $j = 1, 2, \ldots$ также имеет предел, если значение $A^{(j)}$, вычисленное после j-й итерации алгоритма, является в определенном смысле непрерывной функцией от набора $\alpha^{(j-1)}$, вычисленного к началу этой итерации. Во многих случаях это условие выполняется, однако мы повторяем, что даже если это условие не выполняется, то это не должно служить препятствием для применения алгоритма.

Выясним свойство неподвижных точек алгоритма самообучения, т. е. свойство тех точек, которые будут выданы в качестве результата.

Для любого набора $A = (Q, P_k, k \in K)$ определим число $\Phi(A) = \sqrt{\sum_{i=k}^{m} \sum_{k \in K} (Q(k) \cdot P_k(x_i))^2}$. Разностью двух наборов $A^{(1)}$ и $A^{(2)}$ будем называть набор $A' = (Q', P_k')$, такой, что для всех $x \in X$, $k \in K$ справедливо равенство $Q'(k) \cdot P_k'(x) = Q^{(1)}(k) P_k^{(1)}(x) - Q^{(2)}(k) P_k^{(2)}(x)$.

Пусть R^+ — множество всех возможных неотрицательных вещественных чисел. Кривой в $\mathscr{A}$, начинающейся в точке A^*, будем называть функцию $\overline{A} : R^+ \to \mathscr{A}$, такую, что $\overline{A}(0) = A^*$, а $\Phi\left(\frac{d\overline{A}}{dt}\right) = 1$; через $\left.\frac{d\overline{A}}{dt}\right|_{(t=0)}$ обозначим $\lim_{\substack{t \to 0 \\ t > 0}} \frac{\overline{A}(t) - \overline{A}(0)}{t}$.

По-прежнему через $\mathscr{L}(A)$ обозначим $\sum_{i=1}^{m} \log \sum_{k \in K} Q(k) \times P_k(x_i)$, а через $\mathscr{L}_1(A, \alpha)$ обозначим $\sum_{k \in K} \sum_{i=1}^{m} \alpha(x_i, k) \log Q(k) \times P_k(x_i)$.

Очевидно, что в точке $A^* = \arg\max_{A \in \mathcal{A}} \mathcal{L}(A)$ для любого $i = 1, 2, \ldots, m$ имеем $\sum_{k \in K} Q^*(k) P_k^*(x_i) \neq 0$, поэтому для любой кривой $\overline{A}$, начинающейся в A^*, производная $\left.\frac{d\mathcal{L}(\overline{A}(t))}{dt}\right|_{t=0}$ справа существует и принимает неположительное значение.

Очевидно, что для любого значения α в точке $A^* = \arg\max_{A \in \mathcal{A}} \mathcal{L}_1(A, \alpha)$ для всех $i = 1, 2, \ldots, m$; $k \in K$, таких, что $\alpha(x_i, k) \neq 0$, справедливо неравенство $Q^*(k) \cdot P_k^*(x_i) \neq 0$, поэтому для любой кривой $\overline{A}$, начинающейся в A^*, производная $\left.\frac{d\mathcal{L}_1(\overline{A}(t), \alpha)}{dt}\right|_{t=0}$ справа существует и принимает неположительное значение.

Если в точке A^* для любой кривой $\overline{A}$, начинающейся в A^*, производная $\left.\frac{d\mathcal{L}(\overline{A}(t))}{dt}\right|_{t=0}$ справа неположительна, то такую точку будем называть локальным экстремумом функционала $\mathcal{L}(A)$.

Теорема 11. *Если α^* есть неподвижная точка алгоритма самообучения, то $A^* = \arg\max_{A \in \mathcal{A}} \mathcal{L}_1(A, \alpha^*)$ есть локальный экстремум функционала $\mathcal{L}(A)$.*

Доказательство. Обозначим через $\mathcal{L}_2(A, \alpha)$ выражение $\sum_{i=1}^{m} \sum_{k \in K} \alpha(x_i, k) \log \frac{Q(k) P_k(x_i)}{\sum_{k \in K} Q(k) P_k(x_i)}$, а через $\alpha_{ik}(A)$ выражение

$$\alpha_{ik}(A) = \frac{Q(k)\ P_k(x_i)}{\sum_{k \in K} Q(k) P_k(x_i)}.$$

Представим функцию $\mathcal{L}(A)$ так же, как и при доказательстве теоремы 9, но с учетом принятых обозначений:

$$\mathcal{L}(A) = \mathcal{L}_1(A, \alpha^*) - \mathcal{L}_2(A, \alpha^*). \tag{4.102}$$

Для любой кривой $\overline{A}$, начинающейся в $A^* \in \mathcal{A}$,

$$\left.\frac{d\mathcal{L}_1(\overline{A}(t), \alpha^*)}{dt}\right|_{t=0} \leqslant 0. \tag{4.103}$$

Рассмотрим производную $\left.\frac{d\mathcal{L}_2(\overline{A}(t), \alpha^*)}{dt}\right|_{t=0}$ для кривой $\overline{A}$, начинающейся в A^*. Эта производная равна $\left.\sum_{i=1}^{m} \sum_{k \in K} \frac{d\alpha_{ik}(\overline{A}(t))}{dt}\right|_{t=0}$. Поскольку для любых $t \geqslant 0$ и $i = 1, 2, \ldots, m$ справедливо

равенство $\sum_{k \in K} \alpha_{ik}(\overline{A}(0)) = \sum_{k \in K} \alpha_{ik}(\overline{A}(t)) = 1$, имеем $\sum_{k \in K} [\alpha_{ik}(\overline{A}(0)) - \alpha_{ik}(\overline{A}(t))] = 0$ и $\sum_{i=1}^{m} \sum_{k \in K} \frac{d\alpha_{ik}(\overline{A}(t))}{dt} = 0$.

Мы получили, таким образом, что для любой кривой $\overline{A}$, начинающейся в A^*,

$$\left.\frac{d\mathcal{L}_2(\overline{A}(t), \alpha^*)}{dt}\right|_{t=0} = 0. \tag{4.104}$$

В связи с выражениями (4.102), (4.103), (4.104) получаем, что для любой кривой $\overline{A}$, начинающейся в точке $A^* = \arg\max \mathcal{L}_1(A, \alpha^*)$, где α — неподвижная точка, выполняется условие $\left.\frac{d\mathcal{L}(\overline{A}(t))}{dt}\right|_{t=0} \leqslant 0$, и точка A^* есть, таким образом, локальный экстремум функции $\mathcal{L}(A)$. Теорема доказана.

В заключение отметим, что не любой локальный экстремум есть неподвижная точка алгоритма. Так, на рис. 4.3 представлен пример с точками, которые являются локальными экстремумами, и лишь одна из них, обозначенная крестиком, неподвижна.

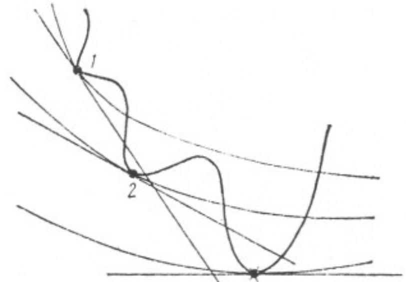


Рис. 4.3. Локальные экстремумы (точки 1 и 2), не являющиеся неподвижными точками

3. Самообучение в двумерных грамматиках. Пусть задана двумерная грамматика $G = \langle T, S, \mathcal{T}, \Omega \rangle$. Как было отмечено в параграфе 4.2, задание грамматики G есть еще и задание последовательностей $t_1, t_2, \ldots, t_{|T|}$ и $T_1^*, T_2^*, \ldots, T_{|T|}^*$, а также функции R.

На множестве S^T существует распределение вероятностей $P: S^T \to R$, которое неизвестно, но известно, что оно имеет вид

$$P(s) = \prod_{t \in T} P(\{t\}/R(t), s(\{t\} \cup R(t))), \quad s \in S^T, \tag{4.105}$$

т. е. известно множество $\mathcal{P}$, содержащее P.

Существует также некоторая случайная выборка $s_1, s_2, \ldots, s_m$ допустимых в G вариантов, взятых из генеральной совокупности, имеющей распределение вероятностей P. Эта выборка известна не полностью, а известно лишь сужение $s_i(T_i^H)$ каждого варианта на поле наблюдения $T_i^H \subset T$, свое для каждого варианта. Обучающей выборкой, таким образом, является последовательность $(T_1^H,\ s_1(T_1^H)),\ (T_2^H,\ s_2(T_2^H)),\ \ldots$ $\ldots,(T_m^H,\ s_m(T_m^H))$.

Задача самообучения заключается в вычислении оценок наибольшего правдоподобия для вероятностных параметров $P(\{t\}/R(t),\ s(\{t\}\cup R(t)))$, т. е. в отыскании распределения вероятностей $P\in\mathcal{P}$, максимизирующего функционал $\sum_{i=1}^{m}\log P(T_i^H,\ s_i(T_i^H))$. Напомнив обозначение $\hat{S}(T',\ s(T')) = \{s' : s'\in S^T,\ s'(T') = s(T')\}$ и учтя, что $P(T_i^H,\ s_i(T_i^H)) = \sum_{s\in\hat{S}(T_i^H,s_i(T_i^H))} P(s)$, функция P имеет вид (4.105), задачу самообучения можно переформулировать как отыскание значений вероятностных параметров распределения $P\in\mathcal{P}$, т. е. чисел $P(\{t\}/R(t),\ s(\{t\}\cup R(t)))$, $t\in T$, $s(\{t\}\cup R(t))\in S^{\{t\}\cup R(t)}$, максимизирующих функционал

$$\mathcal{L}(A) = \sum_{i=1}^{m}\log\sum_{s\in\hat{S}(T_i^H,s_i(T_i^H))}\prod_{t\in T}P(\{t\}/R(t),\ s(\{t\}\cup R(t))). \tag{4.106}$$

Через A в последней формуле обозначена совокупность вероятностных параметров.

Преобразуем выражение (4.106) так, как это было сделано при рассмотрении общей задачи самообучения. Пусть числа $\alpha_i(s)$, $i = 1, 2 \ldots, m$; $s\in S^T$, таковы, что $\alpha_i(s)\geqslant 0$ для любых i и s, а для любого i, кроме того, справедливо условие $\sum_{s\in\hat{S}(T_i^H,s_i(T_i^H))}\alpha_i(s) = 1$. В таком случае

$$\mathcal{L}(A) = \sum_{i=1}^{m}\sum_{s\in\hat{S}(T_i^H,s_i(T_i^H))}\alpha_i(s)\log P(s) -$$

$$-\sum_{i=1}^{m}\sum_{s\in\hat{S}(T_i^H,s_i(T_i^H))}\alpha_i(s)\log\frac{P(s)}{\sum_{s\in\hat{S}(T_i^H,s_i(T_i^H))}P(s)}. \tag{4.107}$$

Если в выражении (4.107) выбрать $\alpha_i(s)$ так, что

$$\alpha_i(s) = \frac{P(s)}{\sum_{s\in\hat{S}(T_i^H,s_i(T_i^H))}P(s)}\quad \text{при}\ s\in\hat{S}(T_i^H,\ s_i(T_i^H)); \tag{4.108}$$

$\alpha_i(s) = 0$ — в противном случае, то любое изменение функции P, увеличивающее первое слагаемое в правой части (4.107),

непременно увеличивает и $\mathcal{L}(A)$. Следовательно, оптимизации функционала (4.107), может выполняться по следующей принципиальной вычислительной схеме. Начиная с некоторого $P^0\in\mathcal{P}$, необходимо получить последовательность $P_1\in\mathcal{P}$, $P_2\in\mathcal{P}$, ..., $P_j\in\mathcal{P}$, ..., причем вычисление $P^{(j+1)}\in\mathcal{P}$ по $P^{(j)}\in\mathcal{P}$ производится в два этапа.

На первом этапе вычисляются величины $\alpha_i^j(s)$ для всех i и s по формуле (4.108). Вычисленное значение $\alpha_i^j(s)$ оказывается равным апостериорной вероятности варианта s по наблюдению $(T_i^H,\ s_i(T_i^H))$ при условии что априорное распределение вероятностей вариантов есть P^j. В действительности эти числа не равны апостериорным вероятностям вариантов, так как значение P^j не обязательно равно действительному априорному распределению. И тем не менее именно эти числа используются на втором этапе для уточнения оценки P, т. е. для вычисления

$$P^{(j+1)} = \arg\max_{P\in\mathcal{P}}\sum_{i=1}^{m}\sum_{s\in\hat{S}(T_i^H,s_i(T_i^H))}\alpha_i(s)\log P(s). \tag{4.109}$$

Для описанной схемы справедливы три теоремы, доказанные для общего алгоритма самообучения. Однако указанная схема является всего лишь принципиальной схемой, а не практической рекомендацией, так как вычисление чисел $\alpha_i(s)$ для всех вариантов s невозможно в силу их чрезвычайно большого количества. Для приведения указанной схемы к практически рекомендуемому виду рассмотрим подробнее выражение (4.108).

Функционал, максимизируемый на втором этапе каждой итерации самообучения, обозначим через $\mathcal{L}_1(A)$ и выполним его преобразование:

$$\mathcal{L}_1(A) = \sum_{i=1}^{m}\sum_{s\in\hat{S}(T_i^H,s_i(T_i^H))}\alpha_i(s)\log P(s) =$$

$$= \sum_{i=1}^{m}\sum_{s\in S^T}\alpha_i(s)\log\prod_{t\in T}P(\{t\}/R(t),\ s(\{t\}\cup R(t))) =$$

$$= \sum_{t\in T}\sum_{i=1}^{m}\sum_{s\in S^T}\alpha_i(s)\log P(\{t\}/R(t),\ s(\{t\}\cup R(t))) =$$

$$= \sum_{t\in T}\sum_{s(\{t\}\cup R(t))}\left(\sum_{i=1}^{m}\sum_{s(T\setminus(\{t\}\cup R(t)))}\alpha_i(s)\right)\log P(\{t\}/R(t),\ s(\{t\}\cup R(t))).$$

Обозначим $\sum\limits_{s(T \setminus (\{t\} \cup R(t)))} \alpha_i(s)$ через $\alpha_i(\{t\} \cup R(t)$, $s(\{t\} \cup R(t)))$ и придем к выводу, что величины $P^*(\{t\}/R(t)$, $s(\{t\} \cup R(t)))$, максимизирующие $\mathcal{L}_1(A)$, должны вычисляться по формуле

$$P^*(\{t\}/R(t),\ s(\{t\} \cup R(t))) = c \sum_{i=1}^{m} \alpha_i(\{t\} \cup R(t),\ s(\{t\} \cup R(t))),$$

где c — нормирующий множитель, выбираемый из условия $\sum\limits_{s(t) \in S} P^*(\{t\}/R(t),\ s(\{t\} \cup R(t))) = 1$. Для реализации второго этапа каждой итерации самообучения нет необходимости, таким образом, знать апостериорную вероятность $\alpha_i(s)$ каждого варианта, а достаточно знать числа $\alpha_i(\{t\} \cup R(t),\ s(\{t\} \cup R(t)))$ для всех i, t и $s(\{t\} \cup R(t))$, где каждое из этих чисел есть апостериорная вероятность фрагмента $s(\{t\} \cup R(t))$ по наблюдению $(T_i^H,\ s_i(T_i^H))$. Процедура вычисления этих апостериорных вероятностей была нами рассмотрена при рассмотрении вероятностной модели наблюдений и схем принятия решений о варианте.

Самообучение в двумерных грамматиках, таким образом, заключается в многократном улучшении (в смысле величины $\mathcal{L}(A)$) вероятностных параметров грамматики, каждое из которых состоит из двух этапов.

Пусть к началу очередной, j-й итерации самообучения получены числа $P^j(\{t\}/R(t),\ s(\{t\} \cup R(t)))$ для всех $t \in T$ и $s(\{t\} \cup R(t))$. В таком случае на первом этапе вычисляются числа $\alpha_i(\{t\} \cup R(t),\ s(\{t\} \cup R(t)))$, $i = 1, 2, \ldots, m$; $t \in T$, $s(\{t\} \cup R(t)) \in S^{\{t\} \cup R(t)}$, каждое из которых равно апостериорной вероятности фрагмента $s(\{t\} \cup R(t))$ по наблюдению $(T_i^H,\ s_i(T_i^H))$. На втором этапе производится вычисление новых значений вероятностных параметров по формуле

$$P^{j+1}(\{t\}/R(t),\ s(\{t\} \cup R(t))) = \frac{\sum\limits_{i=1}^{m} \alpha_i(\{t\} \cup R(t),\ s(\{t\} \cup R(t)))}{\sum\limits_{s(t) \in S} \sum\limits_{i=1}^{m} \alpha_i(\{t\} \cup R(t),\ s(\{t\} \cup R(t)))}.$$

ЗАКЛЮЧЕНИЕ

Укажем те черты, которые отличают известные направления в распознавании образов от направления, которое предложено в этой книге. Любое распознающее устройство производит отображение $k: \mathcal{V} \to \mathcal{K}$ множества $\mathcal{V}$ распознаваемых объектов в множество $\mathcal{K}$ результатов распознавания. Любому результату θ распознавания соответствует подмножество $\mathcal{V}(\theta) = \{v \in \mathcal{V} : k(v) = \theta\}$, называемое классом распознаваемых объектов. Различные направления в распознавании образов отличаются друг от друга тем, что так или иначе вводятся ограничения на функцию k, т. е. постулируется, какого рода отображения будут считаться распознаванием, или накладываются ограничения на классы $\mathcal{V}(\theta)$, т. е. постулируется, какого рода подмножества могут считаться классом объектов.

Описанное в данной монографии направление основано на задании множеств распознаваемых объектов с помощью двумерных грамматик, что определяет его жесткую ориентацию на определенный класс задач обработки. Наличие этой жесткой ориентации определяет наиболее существенные отличия предлагаемого подхода от других известных подходов, где такая жесткая привязка к предметной области отсутствует.

Наиболее характерно для большинства моделей то, что они ориентированы на решение задач распознавания при очень малых априорных знаниях о классе распознаваемых объектов. Это и определяет их привлекательность и сферу применимости, которая оказывается достаточно широкой. Менее заметно, однако, что в рамках известных моделей достаточно затруднительно использование априорных сведений, если они все-таки имеются. Поэтому этап обучения, т. е. настройки параметров алгоритма по заданному материалу обучения, является необходимой и неотъемлемой частью процедуры распознавания. Поэтому исследование процессов обучения составляет основное научное содержание в известных направлениях в распознавании образов.

В разработанном аппарате двумерных грамматик при достаточно ясном понимании цели распознавания в том или ином конкретном приложении двумерная грамматика может быть вполне определенно сформулирована, так, что необходимость в дальнейшем обучении системы просто не возникает. Разработанный формализм, следовательно, достигает цели, которая во введении была сформулирована как развитие теории распознавания в направлении, допускающем формальное задание всех априорных сведений о классе распознаваемых объектов так, чтобы требуемая функция $k : v \to \mathcal{K}$ однозначно следовала из этого задания.

Таким образом, если в известных подходах настройка распознающего алгоритма на ту или другую прикладную задачу осуществляется через процесс обучения, то в разработанном подходе настройка может осуществляться и прямым указанием автора о своем замысле. выраженном средствами двумерных грамматик. В этом случае исчезают романтичный ореол и загадочность, которые характерны для процедур обучения, взамен чего приобретается достаточно прозаичная уверенность в правильном последующем решении задачи. В то же время применение двумерных грамматик совсем не исключает возможности обучения, т. е. настройки определенных параметров под заданный материал обучения. Однако, в отличие от известных подходов, исследованием этих процедур не исчерпывается научная проблематика двумерных грамматик. Даже если двумерная грамматика, порождающая требуемое множество изображений, полностью и недвусмысленно задана, т. е. полностью определена функция, которая должна быть реализована алгоритмом распознавания, построение алгоритма, реализующего эту функцию, не является тривиальной задачей. В рамках двумерно-грамматического подхода, таким образом, вскрыт тот факт, что функции, которые должны реализовать алгоритм распознавания изображений, значительно сложнее тех функций, которые до сих пор использовались в известных моделях распознавания. Это открывает сферу исследований собственно проблемы распознавания, лежащей вне теории обучения распознаванию.

Оправдано ли такое, как мы видим, усложнение модели распознавания, при котором обнаруживаются новые трудные математические проблемы?

Да!

Не следует отождествлять отсутствие трудностей и их незнание. Преодолевать и даже обходить можно только известные трудности.

ПРИЛОЖЕНИЕ

СИСТЕМА ВВОДА И ОБРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

Наиболее наглядной и удобной формой представления сведений о проектируемом изделии для конструктора является эскиз или чертеж. В то же время в системах автоматизированного проектирования (САПР) данные о схеме или чертеже необходимо представить в памяти ЭВМ в некотором специфическом закодированном виде. Поэтому САПР наряду с устройствами отображения и вывода чертежной информации должны быть снабжены быстрыми и удобными средствами ввода этой информации.

Существующие в настоящее время полуавтоматические устройства кодирования чертежно-графической информации все же требуют больших затрат малопроизводительного и утомительного труда человека — кодировщика. Поэтому ввод изображений в ЭВМ является наиболее узким местом в технологическом цикле автоматизированного проектирования, что тормозит широкое и эффективное использование САПР.

В данной работе описываются назначение, функциональные возможности и условия применения системы автоматического ввода и обработки чертежно-графической информации с бумажного носителя (рис. 4.4), созданной в Институте кибернетики имени В. М. Глушкова АН УССР.

Назначение системы. Система предназначена для ввода в ЭВМ широкого класса чертежно-графических документов (в виде чертежей, эскизов, графиков непосредственно на бумажном носителе) и представление этих документов в виде, удобном для хранения на машинном носителе, человеко-машинного редактирования, автоматической обработки и воспроизведения на графических выводных устройствах векторного вида.

Система предназначена также для автоматического распознавания и содержательной интерпретации чертежей и эскизов печатных плат и схем устройств вычислительной техники.

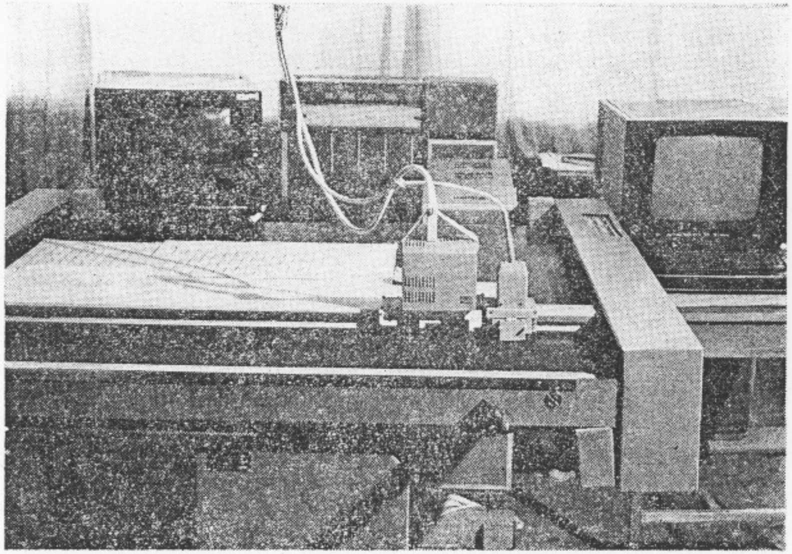


Рис. 4.4. Система ввода в ЭВМ и обработки графической информации

Областью применения системы является автоматизация проектных, конструкторских, картографических работ, автоматизация создания банков чертежей, автоматизация научно-исследовательских работ.

Состав системы и назначение составных частей. В систему входят следующие аппаратурные и программные средства:

УВК СМ-4;

графопостроитель АП-7251;

считывающая головка ЕОАК;

контроллер считывающей головки;

графический дисплей ЭПГ/СМ или ЭПГ-400 или УПГИ;

операционная система ОС РВ;

программное обеспечение.

Графопостроитель АП-7251 является в системе устройством для механического перемещения считывающей головки ЕОАК, укрепленной на его траверсе. Кроме того, на этой же траверсе рядом со считывающей головкой укреплена световая указка, с помощью которой система указывает оператору то или иное место на чертеже или очерчивает ту или иную область на чертеже (например, очерчивает края части поля зрения, подлежащей считыванию). Считывающая головка и световая указка укреплены на графопостроителе дополнительно к пишущей головке, а не взамен ее, так что графопостроитель может быть использован в системе и по своему прямому первоначальному назначению.

Действие **считывающей головки ЕОАК** основано на использовании двух линеек ПЗС по 250 светочувствительных элементов в каждой. Одна из линеек чувствительна к синему цвету, а вторая — к красному. Две линейки в совокупности позволяют, таким образом, считывать цветную черно-бело-красно-синюю графическую информацию.

При фиксированном положении головки проекция светочувствительного поля обеих линеек на плоскость чертежа образует прямоугольник $0{,}1 \times 25$ мм.

При движении головки в направлении, перпендикулярном направлению линейки ПЗС, происходит считывание растровой информации через каждые 0,1 мм. Таким образом, считывающая головка вместе с графопостроителем образует сканирующее устройство, имеющее на выходе растровую информацию о чертеже, где размер одного элемента растра равен $0{,}1 \times 0{,}1$ мм^2.

Растровая информация с двух линеек ПЗС по двум каналам поступает в контроллер считывающей головки. В контроллере производится поразрядное преобразование пары изображений, поступающих со считывающей головки с целью получения двух других изображений, являющихся компиляцией двух исходных. Эта часть контроллера построена как универсальная, т. е. реализует любое поразрядное преобразование изображения, код которого поступает на контроллер от центральной ЭВМ. Затем каждое из двух новых полученных изображений экономно кодируется и в режиме прямого доступа записывается в оперативную память ЭВМ СМ-4. Контроллер выполнен в виде автономного конструктивного блока ЭВМ СМ-4.

С точки зрения программиста графопостроитель, считывающая головка и контроллер представлены восемью программно адресуемыми регистрами, находящимися на общей шине. Один из них является регистром команд и состояний, другой регистром данных графопостроителя. Остальные шесть регистров являются дополнительными регистрами, управляющими собственно процессом считывания.

Графопостроитель, считывающая головка и контроллер образуют считывающее устройство со следующими эксплуатационными характеристиками:

размер поля зрения — управляется от центральной ЭВМ;

максимальный размер поля зрения — 700×1000 мм^2;

цвет считываемой информации — управляется от ЭВМ и выбирается из синего, черного, красного и белого;

размер одной точки растра — $0{,}1 \times 0{,}1$ мм^2;

максимальная скорость сканирования — 250 тыс. точек растра в секунду;

средняя скорость сканирования больших площадей с учетом холостых обратных ходов — 6 мин на чертеже размером 700×1000 мм².

Программное обеспечение состоит из базового программного обеспечения, выполняющего обработку широкого класса графических документов и специализированного обеспечения, выполняющего обработку чертежей и эскизов печатных плат и принципиальных схем устройств вычислительной техники.

Функциональные возможности системы. Система выполняет следующую совокупность преобразований изображения:

экономное кодирование;

сглаживание контуров;

загрубление изображения;

устранение помех;

выделение осевых линий (скелета) и пятен;

спрямление осевых линий и контуров пятен;

сшивание отдельных фрагментов изображения и человекомашинное редактирование информации об изображении в памяти ЭВМ.

Опишем каждый из этих этапов в отдельности.

Экономное кодирование. Размер поля зрения считывающего устройства определяется размером рабочего поля графопостроителя и равен 700×1000 мм². Элементом поля зрения является квадрат размером $0{,}1 \times 0{,}1$ мм². Каждый такой элемент описывается двумя битами, первый из которых определяет, отражает ли данный квадратик свет в синей полосе спектра, а второй — имеется ли отраженный свет в красной полосе. Таким образом, исчерпывающее описание обрабатываемого изображения составляет $1{,}4 \times 10^8$ бит. Такой объем памяти превышает возможности современных миникомпъютеров типа СМ-4. Поэтому первым этапом обработки является экономное кодирование изображения. В данной системе принято кодирование так называемыми особыми местами. Сущность этого кодирования состоит в следующем.

Пусть i — координата y центра квадратика, являющегося элементом поля зрения, а j — координата x центра этого квадратика; $x(i, j)$ — величина, равная 0 или 1 и определяющая наличие или отсутствие отраженного света в этом квадратике. Пара (i, j) целых чисел называется особым местом поля зрения, если

$$\| x(i, j) - x(i+1, j) | - | x(i, j+1) - (i+1, j+1) \| = 1.$$

Экономное кодирование изображения заключается в составлении списка особых мест изображения.

Совокупность особых мест является таким же исчерпывающим описанием изображения, как и совокупность величин $x(i, j)$, т. е. растровое описание, однако для чертежнографической информации является в десятки или сотни раз экономнее.

Представление изображения совокупностью особых мест выполняется аппаратурно с помощью контроллера. Растровое представление изображения, таким образом, отсутствует на любом последующем этапе. Выбор указанного метода кодирования явился опеделяющим в данной системе. В основном именно благодаря выбранному методу кодирования система оказалась реализуемой на миниЭВМ и показывает исключительно высокое быстродействие по сравнению с известными подобными системами.

Сглаживание контуров. Считывающее устройство, применяемое в системе, является устройством дискретного действия. Это неизбежно приводит к тому, что бинарная информация о свете, отраженном от квадратика, находящегося на границе светлого и темного участков изображения является почти произвольной.

Это является причиной того, что края дискретизированного изображения неизбежно являются очень «изрезанными», «рваными» даже в том случае, если исходный графический материал был идеальным. Понимание того, что изрезанность краев не является отображением действительного чертежа, а возникает лишь как следствие его дискретизации, позволяет дополнительно уменьшить объем памяти, необходимый для хранения чертежа.

Задача, решаемая на этом этапе, заключается в том, что смещением некоторых точек контуров изображения в пределах 0,1 мм следует добиться, чтобы особых мест на этом деформированном изображении оказалось как можно меньше. Хотя задача здесь сформулирована именно как задача минимизации памяти, чисто внешний эффект оказывается таким, как будто изображение улучшилось. Улучшение контуров на реальных изображениях дает приблизительно десятикратное уменьшение необходимой памяти.

Загрубление изображения. Загрубление изображения заключается в том, что координаты всех особых мест изменяются на 0,1 мм так, чтобы все они стали кратными 0,2 мм. Процедура загрубления сокращает объем необходимой памяти в 3—4 раза, однако ее применение может привести к искажению исходного чертежа. Поэтому процедура загрубления выполнена в системе в трех вариантах.

В первом варианте загрубление выполняется одновременно со специальными мерами, благодаря которым ни один темный элемент поля зрения не превращается в светлый. Таким образом, загрубление в этом варианте в принципе не может привести к разрывам линий, но может привести к исчезновению белых промежутков между линиями. Применение этого варианта загрубления рекомендуется, если толщина линий чертежа меньше 0,2 мм, а расстояние между краями различных линий более 0,3 мм. Если толщина линий больше 0,3 мм, а расстояние между линиями меньше 0,2 мм, то рекомендуется второй вариант загрубления, при котором ни один светлый элемент изображения не превращается в темный, благодаря чему исключается слияние темных участков.

Если толщина как линий, так и светлых промежутков между линиями больше 0,3 мм, то следует использовать третий вариант загрубления, являющийся более быстродействующим.

Устранение помех. Практически на любом изображении возможны помехи, т. е. небольшие изолированные светлые или темные пятна. Для подавляющего большинства приложений информация об этих пятнах является ненужной и подлежит исключению. Устранение помех приводит к большему или меньшему сокращению необходимой памяти в зависимости от качества исходного чертежа.

Выделение скелета изображения и пятен. Большая часть графического документа представляет собой линии, т. е. графические элементы, толщина которых меньше заданной величины. Кроме линий на чертеже имеются еще и пятна. Примерами пятен являются, например, изображения металлизованных поверхностей на фотошаблоне печатной платы.

Выделение осевых линий, или скелета, представляет собой важный этап обработки изображений и обозначает восстановление траектории движения пишущего инструмента в процессе рисования линий на чертеже.

В основу выделения скелета изображения взято следующее определение. Некоторая точка R в поле зрения принадлежит осевой линии с толщиной δ, если являются темными все точки квадрата Q с центром в точке R и со стороной δ, а любой квадрат $Q_1 \neq Q$, включающий в себя Q, содержит и светлые точки. Точки, принадлежащие осевым тех линий, толщина которых не превышает величины, указанной пользователем, образуют линии скелета. Остальные графические элементы образуют пятна.

Алгоритм выделения осевой подробно изложен в работе [57].

Спрямление осевых линий и пятен. В большинстве своем линии на чертеже содержательно представляют прямолинейные отрезки. Это значит, что автор документа при проведении той или иной линии имел в виду прямолинейный отрезок. Однако в результате небрежности рисования (например, при изготовлении эскиза от руки) даже те линии, которые по

содержанию чертежа должны быть прямыми, в действительности прямыми не являются.

Для исключения упомянутых (и неизбежных) дефектов изготовления чертежа применяется процедура спрямления. При этом, естественно, спрямлению подвергаются не только те линии, которые должны быть прямыми, но и те, которые по своему содержательному назначению являются кривыми, например окружностями. Последние в результате процедуры спрямления будут представлены своей кусочно-линейной аппроксимацией.

Процедура спрямления заключается в замене некоторого криволинейного участка прямолинейным отрезком, концы которого совпадают с концами криволинейного участка. Эта замена происходит только тогда, когда все точки исходной кривой удалены от аппроксимирующего отрезка не более, чем на величину, указанную пользователем. Алгоритм спрямления осевых линий и контуров пятен описан в [54].

Сшивание фрагментов изображения. Вся предшествующая обработка изображения выполняется не одновременно для всего изображения, а для отдельных его частей, совокупность которых, конечно, покрывает все изображение. Это обусловлено как принципом действия считывающего устройства, производящего сканирование чертежа по полосам шириной 25 мм, так и необходимостью выполнения всей указанной обработки на ЭВМ с малым объемом ОЗУ. Так, улучшение контуров, устранение помех и загрубление выполняются отдельно для полос изображения шириной 25 мм. Выделение осевых и спрямление выполняется для еще меньших участков, а именно для квадратов со стороной приблизительно 25 мм. Для представления чертежа в виде единого массива или файла служит процедура сшивания результатов, полученных для отдельных кадров. После выполнения сшивания дополнительно производится процедура спрямления.

Человеко-машинное редактирование изображения. Информация об изображении, введенная в ЭВМ и представленная в векторном виде, может быть отредактирована. Набор операций по изменению непосредственно чертежа включает в себя устранение линий из чертежа, внесение новых линий в чертеж и изменение координат отдельных точек на чертеже. Процесс редактирования отображается на экране графического дисплея и снабжен сервисом по просмотру на экране и редактированию отдельных, даже очень малых фрагментов чертежа в увеличенном виде.

13 8-2045

ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ВОЗМОЖНОСТЕЙ СПЕЦИАЛИЗИРОВАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Обработка чертежей и эскизов печатных плат. На эскизе должен быть представлен в условном виде слой печатной платы, содержащий проводники различной толщины, контактные площадки и отверстия различных типов, метализованные поверхности с различным наполнением и групповые элементы различных типов. Все элементы слоя печатной платы должны быть представлены на эскизе в условных, заранее оговоренных обозначениях.

После обработки эскиза печатной платы базовым программным обеспечением векторное представление этого изображения подается на вход специализированного обеспечения, которое выполняет следующие функции:

1. Обнаружение на эскизе условных обозначений контактных площадок и отверстий и распознавание их типов.
2. Обнаружение на эскизе условных обозначений проводников различных толщин и металлизованных поверхностей.
3. Привязки координат элементов печатной платы к узлам технологической решетки с заданным шагом.

Помимо этих возможностей содержится обширный сервис по контролю и редактированию полученных результатов. После контроля и при необходимости редактирования информация о слое печатной платы представляется на языке ЯГТИ.

Обработка чертежей и эскизов принципиальных схем. Схема, представленная на эскизе, может содержать следующие элементы:

изображения микросхем;

связи;

электрические элементы — диоды, резисторы, конденсаторы и места заземлений;

жгуты;

стрелки (обозначение входных и выходных сигналов);

цифробуквенные надписи, обозначающие тип микросхемы, наименования входных и выходных сигналов, номера контактов на схеме и номера входов в жгуты.

После обработки изображения принципиальной схемы базовым программным обеспечением векторное представление этого изображения поступает на вход специализированного обеспечения, которое выполняет следующие функции:

1. Обнаружение и распознавание надписей;
2. Обнаружение и распознавание условных обозначений микросхем и электрических элементов;
3. Привязка надписей к элементам электрической схемы;
4. Составление таблицы элементов схемы и таблицы связей.

Так же, как и при распознавании эскизов печатных плат, в программное обеспечение включен разнообразный сервис для контроля и, в случае необходимости, редактирования конечных результатов обработки.

Режимы обработки изображений. Указанная выше последовательность преобразований может быть выполнена в различных режимах. Нужный режим обработки определяется оператором в процессе диалога с системой. Прежде всего оператор по запросу системы указывает положение в поле зрения участка поля зрения, подлежащего считыванию. Это положение оператор указывает, посылая в ЭВМ с терминала координаты выбранного прямоугольного участка. Эти координаты оператор чаще всего определяет на глаз. Поэтому если оператор желает, система по его указанию с помощью световой указки прочерчивает границы выбранного участка. Наблюдая этот процесс прочерчивания, оператор может скорректировать выбранные координаты. Оператору представлены большие возможности по использованию цветного ввода. Например, оператор может дать указание вводить в ЭВМ только информацию, представленную на бумажном носителе синим цветом. В этом случае все, что нарисовано на бумаге черным или красным цветом, будет системой игнорироваться. Это удобно в том случае, когда на чертеже представлена не только информация, подлежащая вводу в ЭВМ, но и другая вспомогательная информация (всевозможные конструкторские пометки, шаблоны, миллиметровая сетка и т. п.). Оператор может указать не только один цвет, но и два, и три. Например, он может дать указание ввести все, что представлено синим или черным цветом. В этом случае будет игнорироваться лишь то, что нарисовано красным цветом. И, наконец, оператор может дать указание на ввод как бы двух изображений одновременно, однако представленных на одном бумажном носителе. Например, можно дать указание сформировать два изображения, первое из которых сформировано без информации, представленной синим цветом, а второе — без информации, представленной красным цветом, и запомнить в отдельных файлах эти два изображения. Такой режим является удобным для того случая, когда на чертеже представлены две содержательно различные серии объектов, но обе эти серии подлежат вводу, например сеть железных и шоссейных дорог, две стороны одной печатной платы и т. п. Оператор указывает нужное множество преобразований, которые должны быть выполнены над изображением, и конкретные параметры этих преобразований. Так, он может дать указание не спрямлять изображение, не улучшать контура, а только убрать помехи. В число параметров, указанных оператором, входят помехи, Таблица

Характеристика	«Фуджицу»	«Теремки»
1. Размеры чертежа	420×594 мм	700×1000 мм
2. Способ рисования	По технологической решетке с заданным шагом	Произвольный с примерным соблюдением наклонов с шагом 45°
3. Выполняемые функции	Ввод и распознавание логических элементов проводников и символов. Не предусмотрены жгуты	Ввод и распознавание эскизов печатных плат и принципиальных схем. Преобразование в векторный вид широкого класса чертежно-графических изображений
4. Способ реализации	Аппаратурный. Специализированное распознающее устройство. Состав: устройство ввода барабанного типа; блок распознавания символов из читающего автомата; оперативная память 4 Мбайта; 8-битовый и 16-битовый микропроцессоры.	Аппаратурно-программный. Использует универсальную мини ЭВМ СМ-4 с объемом ОЗУ 256 кбайт, считающую головку и контроллер считывающей головки (6 печатных плат)
5. Время обработки одного изображения размером 420×594 мм	10—15 мин	15—25 мин

наибольшая толщина линии, точность спрямления осевых и контуров. Оператору представлена возможность записи результатов отдельных этапов на диск. Таким образом, оператор может прекратить работу на том или ином этапе и возобновить работу с этого этапа при очередном выходе на ЭВМ.

Оператору представлена также возможность наблюдать на экране графического дисплея результаты отдельных этапов обработки и оценить правильность выбранных им параметров. В любой момент времени оператор может прервать работу системы и возобновить диалог с целью изменения тех или иных своих указаний.

Если оператору необходимо ввести большое количество однотипных графических документов, имеющих одинаковые размеры, и требуется выполнить с ними одинаковые процедуры обработки, то нет необходимости вводить все параметры обработки отдельно для каждого изображения. В этом случае достаточно лишь указать программе, что данное изображение нужно обрабатывать с теми же параметрами, что и предыдущие.

Эксплуатационные параметры системы. Параметры комплекса «Теремки» рассмотрены в сравнении с параметрами системы распознавания принципиальных схем, разработанной фирмой «Фуджицу» (Япония) *Nagata S., Matsuura T., Endo H.* Automatic recognition system for logic circuit diagrams // FUJITSU Sci. Tech. J.— 1985.— 21, N 14.— P. 408—420. Параметры обеих систем приведены в таблице.

Анализируя таблицу, необходимо отметить следующее:

1. Комплекс «Теремки» позволяет кодировать изображения большего размера.

2. Комплекс «Теремки» выполняет более глубокий смысловой анализ изображений, причем изображения структурно более сложные.

3. Комплекс «Теремки» выполнен на базе универсальной ЭВМ, что позволяет использовать его в режиме разделения времени с другими используемыми задачами. Кроме того, он использует минимальное число специализированных устройств в результате чего является по-видимому значительно более дешевым.

4. Система «Фуджицу» предназначена для решения единственной задачи — кодирования принципиальных логических схем. Базовый набор «Теремков», как было показано выше, предназначен для ввода и предварительной обработки широкого класса графических изображений.

5. Комплекс «Теремки» уступает по быстродействию, что объясняется аппаратурной реализацией в японской системе всех процедур обработки.

СПИСОК ЛИТЕРАТУРЫ

1. *Аверинцев М. Б.* Об одном способе описания случайных полей с дискретным аргументом // Пробл. передачи информ.— 1970.— 6, вып. 2.— С. 100—108.
2. *Аверинцев М. Б.* Описание марковских случайных полей при помощи гиббсовских условных вероятностей // Теория вероятностей и ее применения.— 1972.— 17, вып. 1.— С. 21—34.
3. *Автоматизация* анализа и распознавания изображений : Методы и средства. / Под ред. Г. Г. Громова.— Рига : Зинатне, 1979.— Вып. 1.— 260 с.
4. *Айвазян С. А., Бежаева З. И., Староверов О. В.* Классификация многомерных наблюдений.— М. : Статистика, 1974.— 240 с.
5. *Айзерман М. А.* Опыты по обучению машин распознаванию зрительных образов // Биологические аспекты кибернетики.— М., 1962.— С. 174—183.
6. *Айзерман М. А., Браверман Э. М., Розоноэр Л. И.* Методы потенциальных функций в теории обучения машин.— М. : Наука, 1970.— 384 с.
7. *Алгоритмы* обучения распознаванию образов / Под ред. В. Н. Вапника — М. : Сов. радио, 1973.— 815 с.
8. *Алгоритмы* и программы восстановления зависимостей / Под ред. В. Н. Вапника — М. : Наука, 1984.— 815 с.
9. *Апраушева Н. Н.* Исследование одного алгоритма расщепления смеси нормально распределенных классов // Многомерный статистический анализ в социально-экономических исследованиях.— М., 1974.— С. 135—149.
10. *Ахо А., Ульман Дж.* Теория синтаксического анализа, перевода и компиляции.— М. : Мир, 1978.— Т. 1.
11. *Барабанюк Т. Н., Шлезингер М. И.* Об одном алгоритме обучения и оценка качества обучения // Статист. пробл. упр.— 1976.— Вып. 14.— С. 93—104.
12. *Барабанюк Т. Н. и др.* Система автоматического кодирования изображений печатных плат / Т. Н. Барабанюк, Г. К. Карапетян, В. В. Мацелло, С. Я. Тырский, В. М. Шарыпанов, М. И. Шлезингер, Г. А. Шульдешов // Автоматизированные системы обработки изображений : (АСОИз-81) : Тез. докл. I Всесоюз. конф. (9—11 июня 1981 г.).— М., 1981.— С. 109.
13. *Барабанюк Т. Н. и др.* Система автоматического кодирования эскизов схем логических устройств / Т. Н. Барабанюк, В. М. Кийко, В. В. Мацелло, В. М. Шарыпанов, М. И. Шлезингер // Там же.— С. 103.
14. *Башкиров О. А.* Ввод и обработка растровой метрической информации в подсистеме РАСТР-1// Методы и средства обработки сложноструктурированной, семантически насыщенной графической информации : Тез. докл. I Всесоюз. конф. (Горький, сент. 1983 г.).— Горький, 1983.— С. 10.
15. *Беллман Р., Дрейфус С.* Прикладные задачи динамического программирования.— М. : Наука, 1965.— 455 с.
16. *Бокач М. А. и др.* Устройство для оптического чтения и сортировки документов «Бланк-4» / М. А. Бокач, С. А. Зенкина, В. А. Ключевич и др. // Автоматизация ввода письменных знаков в ЦВМ : Тез. докл. IV Всесоюз. конф. (Каунас, 26—27 мая 1977 г.).— Каунас, 1977.— С. 187—190.
17. *Бонгард М. М.* Моделирование процесса обучения узнаванию на универсальной вычислительной машине // Биологические аспекты кибернетики.— М., 1962.— С. 184—191.
18. *Бонгард М. М.* Проблема узнавания.— М. : Наука, 1967.— 320 с.
19. *Вапник В. Н., Лернер А. Я., Червоненкис А. Я.* Системы обучения распознаванию образов при помощи обобщенных портретов // Изв. АН СССР. Техн. кибернетика.— 1965.— № 1.— С. 67—71.
20. *Вапник В. Н., Червоненкис А. Я.* Теория распознавания образов.— М. : Наука, 1974.— 415 с.
21. *Вапник В. Н.* Восстановление зависимостей по эмпирическим данным.— М. : Наука, 1979.— 447 с.
22. *Васильев В. И.* Распознающие системы . Справочник.— Киев : Наук. думка, 1983.— 422 с.
23. *Васин Ю. Г. и др.* Автоматизированная подсистема растрового ввода-вывода и обработки графической информации (РАСТР-1) / Ю. Г. Васин, О. А. Башкиров, Б. М. Чудинович и др. // Методы и средства обработки сложноструктурированной семантически насыщенной графической информации : Тез. докл. I Всесоюз. конф. (Горький, сент. 1963 г.).— Горький, 1983.— С. 10.
24. *Верхаген К. и др.* Распознавание образов : состояние и перспективы / К. Верхаген, Р. Дейн, Ф. Грун и др.— М. : Радио и связь, 1985.— 104 с.
25. *Винцюк Т. К.* Распознавание рукописных знаков методами динамического программирования // Кибернетика и вычисл. техника : Распознавание образов.— 1969.— Вып. 3.— С. 52—77.
26. *Возиянов А. Ф. и др.* Синтез оптимальных эталонов для читающих автоматов и исследование их эффективности / А. Ф. Возиянов, В. Г. Галочкин, В. Г. Калмыков, В. А. Ковалевский, Ю. Л. Провалов, М. И. Шлезингер // Автоматизация ввода письменных знаков в ЦВМ.— 1973.— 1.— С. 135—140.
27. *Волошин Г. Я., Косенкова С. Т.* Метод распознавания, основанный на аппроксимации выборок смесью нормальных законов // Вычисл. системы.— 1975.— Вып. 61.— С. 60—68.
28. *Гимельфарб Г. Л.* Автоматизированная межотраслевая обработка снимков земной поверхности, получаемых с ИСЗ серии Landsat // Зарубеж. радиоэлектрон. — 1983.— № 8.— С. 56—84.
29. *Глазкова Т. Г. и др.* Возможности применения ЭВМ для клинико-рентгенологической дифференциальной диагностики рака и доброкачественного поражения пищевода / Т. Г. Глазкова, К. Н. Гурарий, С. И. Даниленко и др. // Вестн. радиологии и рентгенологии.— 1971.— № 2.— С. 25—28.
30. *Гребнев В. А., Шлезингер М. И.* Самонастраивающийся алгоритм для распознавания машинописных изображений // Распознавание образов и конструирование читающих автоматов.— 1967.— Вып. 2.— С. 74—82.
31. *Глушков В. М.* К вопросу о самообучении в персептроне // Журн. вычисл. математики и мат. физики.— 1962.— № 6.— С. 1102—1110.

32. *Глушков В. М.* Теория обучения одного класса дискретных персептронов // Там же.— № 2.— С. 317—335.
33. *Губерман Ш. А., Извекова М. Л., Хургин Я. И.* Применение методов распознавания образов при интерпретации геофизических данных // Самообучающиеся автоматические системы.— М., 1966.— С. 48—51.
34. *Добрушин Р. Л.* Описание случайного поля при помощи условных вероятностей и условия его регулярности // Теория вероятностей и ее применения.— 1968.— **13**, вып. 2.— С. 201—229.
35. *Добрушин Р. Л.* Задание систем случайных величин при помощи условных распределений // Там же.— 1970.— **15**, вып. 3.— С. 469—497.
36. *Есин Н. А., Шлезингер М. И.* Синтез вероятностной автоматной грамматики, описывающей заданное множество последовательностей // Кибернетика.— 1977.— № 4.— С. 116—120.
37. *Журавлев Ю. И.* Экстремальные задачи, возникающие при обосновании эвристических процедур // Пробл. прикл. математики и механики.— М., 1971.— С. 67—74.
38. *Журавлев Ю. И., Никифоров В. В.* Алгоритмы распознавания, основанные на вычислении оценок // Кибернетика.— 1971.— № 3.— С. 1—11.
39. *Журавлев Ю. И.* Непараметрические задачи распознавания образов // Там же.— 1976.— № 6.— С. 93—103.
40. *Журавлев Ю. И.* Корректные алгебры над множеством некорректных (эвристических) алгоритмов // Там же.— 1977.— № 4.— С. 14—21; № 6.— С. 21—27; 1978.— № 2.— С. 35—43.
41. *Журавлев Ю. И.* Об алгебраическом подходе к решению задач распознавания или классификации // Пробл. кибернетики.— 1978.— Вып. 33.— С. 5—68.
42. *Журавлев Ю. И., Исаев И. В.* Построение алгоритмов распознавания корректных для заданной контрольной выборки // Журн. вычисл. математики и мат. физики.— 1979.— **19**, № 3.— С. 726—738.
43. *Журавлев Ю. И. и др.* Задачи распознавания и классификации со стандартной обучающей информацией / Ю. И. Журавлев, А. А. Зенкин, А. И. Зенкина и др. // Там же.— 1980.— **20**, № 5.— С. 1294—1309.
44. *Загоруйко Н. Г.* Методы распознавания и их применение.— М. : Сов. радио, 1972.— 137 с.
45. *Загоруйко Н. Г., Елкина В. М., Тимеркаев В. С.* Алгоритм заполнения пропусков в эмпирических таблицах // Вычисл. системы.— 1975.— Вып. 61.— С. 3—27.
46. *Загоруйко Н. Г.* Эмпирическое предсказание и распознавание образов. — Новосибирск : Наука. Сиб. отд-ние, 1980.— 137 с.
47. *Загоруйко Н. Г.* Классификация задач прогнозирования на таблицах «объект — свойство» // Машинные методы обнаружения закономерностей : Вычисл. системы.— 1981.— Вып. 88.— С. 3—7.
48. *Закревский А. Д.* Выявление импликативных закономерностей в булевом пространстве признаков и распознавание образов // Кибернетика.— 1982.— № 1.— С. 1—6.
49. *Ивахненко А. Г.* Самообучающиеся системы распознавания и автоматического управления.— Киев : Техніка, 1969.— 512 с.
50. *Ивахненко А. Г.* Эвристическая самоорганизация — основное направление развития технической кибернетики // Вестн. АН УССР.— 1970.— № 7.— С. 23—37.
51. *Ивахненко А. Г.* Индуктивный метод самоорганизации моделей сложных систем.— Киев : Наук. думка, 1982.— 296 с.
52. *Калмыков В. Г. и др.* Сравнительный анализ алгоритмов синтеза линейного решающего правила для проверки сложных гипотез / В. Г. Калмыков, А. А. Сухоруков, М. И. Шлезингер и др. // Автоматика.— 1981.— № 1.— С. 3—9.

53. *Карапетян Г. К., Мацелло В. В.* Структурный алгоритм распознавания графических изображений // Распознавание образов и автоматизация проектирования робототехнических зрительных систем.— Киев, 1982.— С. 44—54.
54. *Карапетян Г. К., Шульдешов Г. А.* Однопроходный алгоритм линейной аппроксимации последовательностей точек на плоскости // Кибернетика.— 1984.— № 1.— С. 114—117.
55. *Кийко В. М., Шлезингер М. И.* Анализ изображений, порождаемых блочными двумерными грамматиками // Обработка и распознавание сигналов.— Киев, 1975.— С. 19—37.
56. *Кийко В. М.* Алгоритм распознавания изображений, порождаемых двумерными блочными грамматиками // Распознавание образов.— Киев, 1977.— С. 3—23.
57. *Кийко В. М., Шлезингер М. И.* Алгоритм выделения отрезков осевой линии на графических изображениях.— Киев, 1983.— с. (Препр. / АН УССР. Ин-т кибернетики; № 83-19).
58. *Кийко В. М., Шлезингер М. И.* Алгоритм кодирования графических изображений на основе выделения отрезков их осевых линий // Методы и средства обработки сложноструктурированной, семантически насыщенной графической информации : Тез. докл. I Всесоюз. конф. (Горький, сент. 1983 г.).— Горький, 1983.— С. 10.
59. *Кийко В. М., Шлезингер М. И.* Выделение и кодирование линий в системе ввода в ЭВМ графических изображений // Управляющие системы и машины.— 1984.— № 1.— С. 56—59.
60. *Козинец Б. Н.* Рекуррентный алгоритм разделения выпуклых оболочек двух множеств // Алгоритмы обучения распознаванию образов.— М., 1973.— С. 43—50.
61. *Ковалевский В. А.* Методы оптимальных решений в распознавании изображений.— М. : Наука, 1976.— 328 с.
62. *Коваль В. К., Шлезингер М. И.* Двумерное программирование в задачах анализа изображений // Автоматика и телемеханика.— 1976.— № 8.— С. 149—168.
63. *Коваль В. К., Шлезингер М. И.* Оптимизация функций на множестве, заданном локально-конъюнктивным предикатом // Распознавание образов.— Киев, 1976.— С. 3—13.
64. *Коваль В. К.* Распознавание по частям // Там же.— Киев, 1977.— С. 41—53.
65. *Крупников Г. П. и др.* Зарубежные серийно выпускаемые анализаторы изображений / Г. П. Крупников, И. А. Марков, Н. М. Подвысоцкая и др. // Автоматизация анализа и распознавания изображений: Методы и средства. / Под ред. Г. Г. Громова.— 1979.— Вып. 1.— С. 191—216.
66. *Лбов Г. С.* Методы обработки разнотипных экспериментальных данных. — Новосибирск : Наука, 1981.— 160 с.
67. *Лебедев Д. Г., Лебедев Д. С.* Дискретизация изображений посредством выделения и квантования контуров // Изв. АН СССР. Техн. кибернетика.— 1965.— № 1.— С. 88—92.
68. *Лебедев Д. С. и др.* Синтез известных изображений Марса по фотоматериалам, полученным с космического аппарата «Марс-5» / Д. С. Лебедев, Л. П. Ярославский, М. К. Нараева и др. // Докл. АН СССР.— 1975.— **225**, № 6.— С. 1288—1291.
69. *Лохвицкий М. С., Шахгильдян В. В.* Методы адаптивного приема сигналов.— М. : Связь, 1974.— 159 с.
70. *Матросов В. Л.* Корректные алгебры ограниченной емкости над множеством некорректных алгоритмов // Докл. АН СССР.— 1980.— **253**, № 2.— С. 25—30.

71. *Матросов В. Л.* Корректные алгебры ограниченной емкости над множеством алгоритмов вычисления оценок // Журн. вычисл. математики и мат. физики.— 1981.— **21**, № 5.— С. 1276—1291.
72. *Мацелло В. В., Шлезингер М. И.* Синтаксический анализ изображений в процессе построчного сканирования // Изв. АН СССР. Техн. кибернетика.— 1982.— № 3.— С. 173—179.
73. *Мацелло В. В. и др.* Базовый набор программных средств системы обработки графических изображений / В. В. Мацелло, С. Я. Тырский, М. И. Шлезингер и др. // Методы и средства обработки сложноструктурированной, семантически насыщенной графической информации : Тез. докл. I Всесоюз. конф. (Горький, сент. 1983 г.).— Горький, 1983. — С. 38—39.
74. *Мацелло В. В. и др.* Система ввода и обработки графической информации / В. В. Мацелло, Ю. Л. Провалов, В. М. Шарыпанов, М. И. Шлезингер // Управляющие системы и машины.— 1986.— № 6 — С. 56—60.
75. *Миленький А. В.* Классификация сигналов в условиях неопределенности. Статистические методы самообучения в распознавании образов.— М. : Сов. радио, 1971.— 328 с.
76. *Минский М., Пэйперт С.* Персептроны.— М. : 1971.— 262 с.
77. *Морозов В. С., Морозова В. А.* Прогнозирование индивидуального срока службы электронных приборов с помощью метода обобщенных портретов // Электрон. техника. Сер. I: Электрон. СВЧ.— 1969.— Вып. 9.— С. 75—81.
78. *Надь Дж.* Цифровая обработка изображений, полученных при дистанционном исследовании природных ресурсов // Распознавание образов при помощи цифровых вычислительных машин.— М., 1974.— С. 92—124.
79. *Нильсон Н.* Обучающиеся машины.— М. : Мир, 1967.— 180 с.
80. Персептрон — система распознавания образов / Под общ. ред. А. Г. Ивахненко.— Киев : Наук. думка, 1975.— 430 с.
81. *Прэтт У.* Цифровая обработка изображений : Пер. с англ.— М. : Мир, 1982.— Кн. 2.— 430 с.
82. *Рамачандран Е.* Метод кодирования для векторного представления технических чертежей // Тр. инж. ин-та по электротехнике и радиоэлектронике.— 1980.— **68**, № 7.— С. 68—73.
83. *Распознавание* образов при помощи цифровых вычислительных машин / Под ред. Л. Д. Хармона.— М. : Мир, 1974.— 163 с.
84. *Раудис Ш. Ю.* О количестве априорной информации при построении алгоритма классификации // Изв. АН СССР. Техн. кибернетика.— 1972.— № 4.— С. 168—174.
85. *Раудис Ш. Ю.* Оценка вероятности ошибки классификации // Статист. пробл. упр.— 1973.— Вып. 5.— С. 9—14.
86. *Раудис Ш. Ю.* Определение числа полезных признаков в задаче классификации // Там же.— 1976.— Вып. 14.— С. 137—149.
87. *Раудис Ш. Ю., Юшкавичус К.* Вычисление ожидаемой ошибки классификации для кусочно-линейного решающего правила // Там же. — С 117—126.
88. *Раудис Ш. Ю.* Ограниченность выборок в задачах классификации // Там же.— Вып. 18.— С. 1—180.
89. *Розенблатт Ф.* Принципы нейродинамики.— М. : Мир, 1965.— 480 с.
90. *Рудаков К. В.* О корректности алгоритмов распознавания типа потенциальных функций // Журн. вычисл. математики и мат. физики.— 1980.— **20**, № 3.— С. 737—744.
91. *Рудаков К. В.* О категорном подходе к исследованию алгебраических расширений семейств алгоритмов распознавания графической информации // Методы и средства обработки сложноструктурированной, семантически насыщенной графической информации : Тез. докл. I Всесоюз. конф. (Горький, сент. 1983 г.).— Горький, 1983.—С. 184.

92. *Святогор Л. А., Шлезингер М. И.* Решение задачи распознавания машинописных знаков с учетом пространственной дискретизации // Кибернетика и вычислительная техника. Распознавание образов.— Киев, 1969.— Вып. 3.— 145—157.
93. *Святогор Л. А.* Разделяющая поверхность // Энцикл. кибернетики.— Киев, 1973.— **2**,— С. 362.
94. *Семенков О. И., Леонович Э. Н., Апарин Г. П.* Анализ некоторых подсистем ввода графических данных в информационно-вычислительной системе проектирования // Автоматизация проектирования сложных систем.— 1976.— Вып. 2.— С. 154—163.
95. *Семенков О. И., Абламейко С. В.* Методы и алгоритмы обработки растровой графической информации.— Минск : Ин-т техн. кибернетики АН БССР, 1984.— 115 с.
96. *Стрэнд Р.* Распознавание оптических образов при экспериментах на трековых камерах с частицами высоких энергий // Распознавание образов при помощи цифровых вычислительных машин.— М., 1974.— С. 15—37.
97. *Ткачев И. И.* Об эквивалентности некоторых моделей для распознавания образов // Кибернетика.— 1980.— № 3.— С. 147.
98. *Ткачев И. И.* Эквивалентные преобразования в одном классе распознающих систем // Там же.— 1981.— № 1.— С. 27—32.
99. *Ту Дж., Гансалес Р.* Принципы распознавания образов.— М. : Мир, 1978.— 411 с.
100. *Фу К. С.* Последовательные методы в распознавании образов и обучении машин.— М. : Наука, 1971.— 256 с.
101. *Фу К. С.* Структурные методы в распознавании образов.— М. : Мир, 1977.— 320 с.
102. *Халмош П. Р.* Как писать математические тексты // Успехи мат. наук.— 1971.— **26**, вып. 5.— С. 243—269.
103. *Хармон Л. Д.* Автоматическое распознавание печатных и рукописных письменных знаков // Распознавание образов при помощи цифровых вычислительных машин.— М., 1974.— С. 76—91.
104. *Цыпкин Я. З.* Адаптация и обучение в автоматических системах.— М. : Наука, 1968.— 400 с.
105. *Шлезингер М. И.* О самопроизвольном различении образов // Читающие автоматы.— Киев, 1965.— С. 38—45.
106. *Шлезингер М. И.* Самообучение распознаванию образов как задача статистической аппроксимации // II симпоз по кибернетике, Тбилиси, 1965 : (Тез., аннотации, рефераты).— Тбилиси, 1965.— С. 147.
107. *Шлезингер М. И.* Экспериментальная проверка алгоритма самообучения // Распознавание образов и конструирование читающих автоматов.— 1967.— Вып. 2.— С. 64—73.
108. *Шлезингер М. И., Святогор Л. А.* О построении эталонов для корреляционных читающих автоматов // Тр. III Всесоюз. конф. по информационно-поисковым системам и автоматизированной обработке науч.-техн. информ. . В 4 т.— М., 1967.— Т. 3.— С. 129—139.
109. *Шлезингер М. И.* Алгоритмы самообучения распознаванию образов // Конф. по теории автоматов и искусств. мышлению, Ташкент, 27—31 мая 1968 г. : Аннотации, докл. и программа.— М., 1968.— С. 101.
110. *Шлезингер М. И.* Взаимосвязь обучения и самообучения в распознавании образов // Кибернетика.— 1968.— № 2.— С. 81—88.
111. *Шлезингер М. И.* Самообучение распознаванию образов — теоретические вопросы и алгоритмы // IV Всесоюз. шк.-семинар (28 сент. 1968 г.) Автоматическое распознавание слуховых образов : (АРСО-IV).— Киев, 1968.— С. 42—58.
112. *Шлезингер М. И.* Последовательные алгоритмы самообучения // Распознавание образов и конструирование читающих автоматов.— 1969.— Вып. 1.— С. 3—11.

113. *Шлезингер М. И.* Исследование одного класса распознающих устройств, решающих задачу проверки сложных гипотез // Автоматика.— 1972.— № 2.— С. 38—42.
114. *Шлезингер М. И.* Синтез линейного решающего правила для одного класса задач распознавания образов // Изв. АН СССР. Техн. кибернетика.— 1972.— № 5.— С. 157—160.
115. *Шлезингер М. И.* Распознавание процессов // Энцикл. кибернетики. — 1973.— **2**.— С. 371—373.
116. *Шлезингер М. И.* Самообучение распознаванию образов // Там же.— С. 381—382.
117. *Шлезингер М. И.* Оптимизация устройств, реализующих кусочно-линейные решающие правила // Автоматизация ввода письменных знаков в ЦВМ. Вопросы синтеза читающих автоматов : Материалы III Всесоюз. конф. (12—14 июня 1973).— Каунас, 1973.— С. 126—131.
118. *Шлезингер М. И.* Синтаксический анализ двумерных зрительных сигналов в условиях помех // Кибернетика.— 1976.— № 4.— С. 113—130.
119. *Шлезингер М. И.* Распознавание изображений // Укр. Сов. энцикл.— 1983.— Т. 9.— С. 182.
120. *Шлезингер М. И.* Особенности кодирования изображений в системах АРМ // Техника, экономика, информация. Сер. Автоматизация проектирования.— 1983.— Вып. 1.— С. 74—80.
121. *Шлезингер М. И. и др.* Принципы построения базового обеспечения системы обработки растровой графической информации / М. И. Шлезингер, В. В. Мацелло, А. Д. Предельский и др. // Техника, экономика, информация. Сер. Автоматизация проектирования.— 1983.— Вып. 2.— С. 30—39.
122. *Эндрюс Г.* Применение вычислительных машин для обработки изображений.— М. : Энергия, 1977.— 160 с.
123. *Якубович В. А.* Рекурентные конечно сходящиеся алгоритмы решения систем неравенств // Докл. АН СССР — 1966.— **166**, № 6.— С. 1308—1311.
124. *Якубович В. А.* Конечно сходящиеся алгоритмы решения счетных систем неравенств и их применение в задачах синтеза адаптивных систем // Там же.— 1969.— **189**, № 3.— С. 495—498.
125. *Ярославский Л. П.* Устройства ввода-вывода изображений для ЭЦВМ.— М. : Энергия, 1968.—88 с.
126. *Ярославский Л. П.* Введение в цифровую обработку изображений.— М. : Сов. радио, 1979.— 311 с.
127. *Coak C.* A Chomsky Hierarchy of Isotonic Array Grammars and Languages // Comput. Graph. and Image Process.— 1978.— **8**, N 1.— P. 144—152.
128. *Haralick R. M.* Structural Pattern Recognition Homomorphisms and Arrangements // Pattern Recogn.— 1978.— **10**, N 3.— P. 223—236.
129. *Haralick R. M., Mohammed J. L., Zucker S. W.* Compatibilities and the Fixed Points of Arithmetic Relaxation Process // Comput. Graph. and Image Process.— 1980.— **13**, N 3.— P. 242—256.
130. *Janssens D., Rozenberg G., Verraedt R.* On Sequential and Parallel Node-Rewriting Graph Grammars // Ibid.— 1982.— **18**, N 3.— P. 279—304.
131. *Martelli A.* Edge Detection Using Heuristic Search Methods // Ibid.— 1972.— **1**, N 2.— P. 131—146.
132. *Montanari U.* Separable Graphs, Planar Graphs and Web Grammars // Inform. and Contr.— 1970.— **16**, N 3.— P. 243—268.
133. *Muralidhar K. H. and others.* Automatic Digitization of Printed Circuit Layouts / K. H. Muralidhar C. N. Ajit, S. Rajaram, Dr. A. Prabhakar // Comput. Industry.— 1982.— N 3.— P. 199—206.

134. *Ota P.* Mosaic Grammars // Pattern Recogn.— 1975.— **7**, N 1/2.— P. 61—65.
135. *Rosenfeld A., Hummel R., Zucker S.* Scene Labeling by Relaxation Operation // IEEE Trans. Syst. Man, and Cybern.— 1976.— **6**, N 6.— P. 420—433.
136. *Rothstein I., Weiman C.* Parallel and Sequential Specification of Context Sensitive Language for Straight Lines on Grids // Comput. Graph. and Image Process.— 1976.— **5**, N 1.— P. 106—124.
137. *Siromoney G., Siromoney R., Krithivasan K.* Abstract Families of Matrices and Picture Languages // Ibid.— 1972.— **1**, N 3.— P. 284—307.
138. *Siromoney G., Siromoney R., Krithivasan K.* Array Grammars and Kolam // Ibid.— 1974.— **3**, N 1.— P. 63—82.
139. *Siromoney G., Siromoney R., Subramanian K.* Stochastic Table Arrays // Ibid.— 1982.— **18**, N 2.— P. 202—211.
140. *Wang F.* Parallel Context — Free Array Grammars Normal Forms // Ibid.— 1961.— **15**, N 3.— P. 296—300.