Часть III

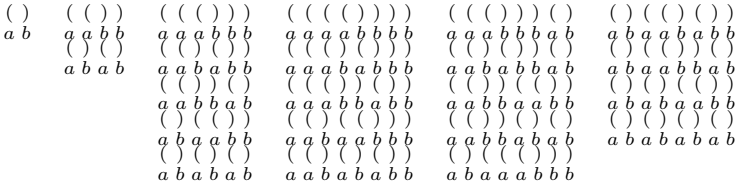
Языки, грамматики, автоматы

Глава 10

Языки и конечные автоматы

10.1 Язык Дика

Как мы знаем, правильные скобочные структуры перечисляются числами Каталана. Выпишем все правильные скобочные структуры до порядка 4:



Если обозначить левую скобку буквой a, а правую — буквой b, то можно переписать правильные скобочные структуры в виде «слов» в алфавите $\{a, b\}$. В приведенной выше таблице под каждой скобочной структурой записано соответствующее ей слово.

При такой записи мы получаем не все слова в алфавите $\{a, b\}$, а только некоторые. Например, слова a, b, aa, ba не соответствуют никаким правильным скобочным структурам.

Определение 10.1.1. Пусть $A = \{a_1, a_2, \ldots, a_k\}$ — произвольный конечный набор различных букв. *Словом* в алфавите A называется произвольная конечная последовательность букв $\alpha_1\alpha_2\ldots\alpha_m$, где $\alpha_i \in A$, $i = 1, \ldots, m$. Число m называется *длиной слова*. *Языком* над алфавитом A называется произвольное (конечное или бесконечное) множество слов в алфавите A.

Пустое слово λ имеет длину 0 и может входить или не входить в язык.

Пример 10.1.2. Язык $\mathcal{F}$ состоит из слов в алфавите $\{a, b\}$, не содержащих двух букв b подряд: $\lambda, a, b, ab, ba, aaa, aab, aba, baa, bab, aaaa, \ldots$.

Множество правильных скобочных структур вместе с пустой структурой образует язык над алфавитом $\{a, b\}$. Этот язык называется *языком*

Рис. 10.1: a) Простой конечный автомат и b) Тот же автомат с выделенным принимающим состоянием

Дика. Конечно, тот же язык мы могли бы рассматривать и над алфавитом $\{(,)\}$; просто символы a, b в нашем восприятии более соответствуют термину «буква».

10.2 Конечные автоматы

Посмотрим на граф с ориентированными ребрами, изображенный на рис. 10.1. Из каждой вершины этого графа выходит два ребра, одно из которых помечено буквой a, а другое — буквой b. Будем называть такой граф автоматом, а его вершины — состояниями. Состояния помечены прописными латинскими буквами.

В любой момент времени автомат находится в одном из состояний — мы будем называть это состояние текущим. При получении на вход буквы автомат переходит из текущего состояния в то, на которое указывает ребро, выходящее из текущего состояния и помеченное этой буквой. Тем самым, каждое слово в алфавите $\{a, b\}$ можно воспринимать как «программу» для автомата: каждая буква в слове является командой, переводящей автомат из одного состояния в другое. Одно из состояний — то, в котором автомат находится до получения первой буквы, — называется начальным. Мы будем всегда помечать его буквой B.

Так, например, выглядит последовательность состояний автомата с рис. 10.1, на вход которого подано слово *baabab*:

$$B \longrightarrow C \longrightarrow B \longrightarrow C \longrightarrow B \longrightarrow C \longrightarrow B.$$

Количество состояний в этой последовательности на единицу больше числа букв во входном слове, а количество стрелок равно этому числу.

Можно пометить некоторые состояния в автомате как принимающие. Принимающие состояния мы будем обводить двойными кружками, см. рис. 10.1 b). Будем говорить, что автомат принимает слово w, если по окончании его обработки он оказывается в принимающем состоянии. В противном случае автомат не принимает слово. Например, автомат с рис. 10.1 b) принимает слово *baabab*, поскольку по окончании его обработки он оказывается в принимающем состоянии B, но не принимает слово a.

Каждому автомату с принимающими состояниями можно сопоставить язык принятых слов. Этот язык состоит из всех слов, принимаемых этим автоматом. Говорят, что автомат над алфавитом A *распознает язык* L над

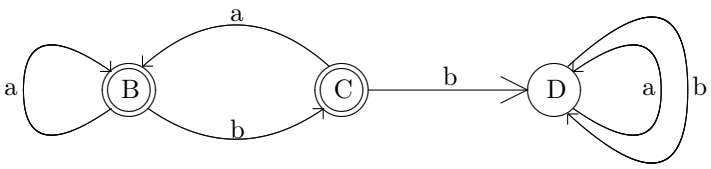


Рис. 10.2: Конечный автомат, распознающий язык $\mathcal{F}$

тем же алфавитом, если он принимает любое слово из языка L и не принимает никакое другое слово.

Например, автомат с рис. 10.1 b) распознает язык, состоящий из всех слов четной длины. Действительно, переходы из состояния B в состояние C и обратно не зависят от очередной буквы во входном слове. Поэтому окончательное состояние зависит лишь от длины слова, точнее — от четности это длины.

Построим автомат, который распознает язык $\mathcal{F}$ из примера 10.1.2 в предыдущем разделе.

Для описания автомата мы должны описать набор его состояний, стрелки, указывающие переходы из одного состояния в другое в зависимости от буквы на входе, а также сказать, какие состояния являются принимающими. Наш автомат будет иметь три состояния:

- начальное состояние B; автомат переходит в это состояние всякий раз, получая букву a, при условии, что в слове еще не встретились две буквы b подряд;

- состояние C, отвечающее ситуации, когда последней обработанной буквой в слове была буква b, при условии, что в слове еще не встретились две буквы b подряд;

- состояние D, отвечающее ситуации, когда в слове уже встретилось две буквы b подряд.

Теперь понятно, как строить и переходы между состояниями. Из состояния B мы переходим обратно в B, если на вход поступила буква a, и в C, если на входе буква b. Находясь в состоянии C, автомат переходит в состояние B, если на входе буква a, и в состояние D, если на вход подана буква b (что означает, что во входном слове встретились две буквы b подряд). Наконец, обе стрелки из состояния D ведут в то же состояние D — если в слове уже встретились две буквы b подряд, то его исправить нельзя.

Состояния B и C являются принимающими — автомат может находиться в одном из этих состояний только, если во входном слове не было подряд идущих букв b. Наоборот, автомат попадает в состояние D только, если во входном слове такие буквы были. Поэтому автомат с рис. 10.2 распознает язык $\mathcal{F}$.

Дадим теперь формальное определение конечного автомата и распознаваемого им языка.

Определение 10.2.1. *Автомат* над данным алфавитом A представляет собой корневой ориентированный граф, ребра которого помечены буквами алфавита, такой, что для каждой буквы алфавита A из каждой вершины графа (*состояния автомата*) выходит ровно одно ребро, помеченное этой буквой. Автомат называется *конечным*, если соответствующий граф конечен. При *обработке* слова над алфавитом A автомат переходит из *начального состояния* (корня графа) в то, куда ведет путь из векторов, помеченных буквами этого слова.

Определение 10.2.2. Автомат над алфавитом A, среди состояний которого выделено подмножество *принимающих состояний*, *распознает язык* L над тем же алфавитом, если конечное его состояние при обработке любого слова языка L является принимающим, а при обработке любого другого слова — нет.

Отметим, что наряду с изображением конечного автомата с помощью графа его легко задавать и таблицей. Строки в такой таблице помечены состояниями автомата, а столбцы — буквами алфавита. Например, автомат, распознающий язык $\mathcal{F}$ задается с помощью таблицы

	a	b
B	B	C
C	B	D
D	D	D

Такая таблица называется *таблицей перехода*. Чтобы информация, представленная в таблице перехода, описывала автомат полностью, необходимо добавить еще одну колонку, в которой про каждое состояние указано, является оно принимающим или нет.

Вовсе не всякий язык можно распознать конечным автоматом. Языки, распознаваемые конечными автоматами, называются *регулярными*. Например, язык $\mathcal{F}$ является регулярным — мы построили распознающий его конечный автомат. Напротив, язык Дика регулярным не является. Для доказательства этого нам понадобится следующее определение.

Определение 10.2.3. Пусть u, v — два слова над алфавитом A. Мы будем говорить, что язык L над тем же алфавитом *различает* слова u и v, если существует слово x над A, такое, что одно из слов ux и vx принадлежит языку L, а другое — нет. Такие два слова u и v будем называть *L-различимыми*.

Очевидно, что все слова над алфавитом A разбиваются на классы попарно L-неразличимых слов. Действительно, если слова u, v L-неразличимы и слова v, w также L-неразличимы, то и слова u, w также L-неразличимы, поэтому неразличимость является отношением эквивалентности. Отметим, что если u — слово языка L, а $v \notin L$, то слова u и v L-различимы: в качестве x можно взять пустое слово.

Теорема 10.2.4. *Язык L над конечным алфавитом A является регулярным в том и только в том случае, если число классов L-неразличимых слов над алфавитом A конечно.*

В одну сторону утверждение теоремы очевидно. Действительно, предположим, что существует конечный автомат, различающий язык L. После обработки любых двух L-различимых слов u, v этот автомат должен оказываться в двух различных состояниях — иначе обработка любых слов ux и vx закончится в одном и том же состоянии. Но это противоречило бы существованию слова x, для которого лишь одно из слов ux и vx принадлежит языку L.

С другой стороны, если количество классов L-неразличимых слов над A конечно, то возьмем эти классы в качестве состояний автомата. Пусть C — одно из состояний. Для любого слова $u \in C$ и любой буквы $a \in A$ направим стрелку a из C в состояние, содержащее слово ua. Понятно, что конечное состояние не зависит от того, с какого слова u мы начинали. Объявим теперь принимающими те состояния, которые содержат слова языка L. Как мы видели выше, эти состояния не могут содержать слов, не входящих в язык. Тем самым мы построили конечный автомат, распознающий язык L. Очевидно, что число его состояний минимально возможное.

Количество классов $\mathcal{D}$-неразличимых слов, где $\mathcal{D}$ — язык Дика, бесконечно. Действительно, любые два слова u и v различной длины, состоящие только из букв a, принадлежат различным классам: если длина слова u равна ℓ и мы припишем к нему слово длины ℓ, состоящее только из букв b, то получим слово из языка Дика, а приписывание того же слова к слову v не даст слова языка Дика. Поэтому язык Дика не распознается конечным автоматом, а значит, нерегулярен.

10.3 Автоматы со стеком

Нельзя ли так подправить понятие автомата, чтобы автомат, распознающий язык Дика, все-таки можно было построить? Оказывается, это сделать несложно. Для этого можно снабдить автомат простой памятью — стеком. Перед началом работы автомата стек пуст. В стек можно класть обрабатываемые автоматом буквы алфавита, однако в любой момент он обеспечивает доступ только к последней положенной букве — вершине стека. Тем самым, стек похож на детскую пирамидку с центральным стержнем, на которую можно сверху класть кружки́ (или на магазин с патронами у боевого автомата). Букву с вершины стека можно снять, тогда верхней становится предыдущая положенная буква — или стек остается пустым. С пустого стека нельзя ничего снять — при попытке сделать это автомат ломается.

Изобразим автомат со стеком, распознающий язык Дика, с помощью таблицы перехода:

	a	b, стек пуст	b, стек непуст
B	B, положить a в стек	D	B, снять верхушку стека
D	D	D	D

В автомате со стеком переходы зависят не только от текущего состояния автомата и входного символа, но и от значения верхушки стека. В нашем случае состояние D — не принимающее. Автомат может попасть в состояние D только, если в некоторой начальной части входного слова оказалось больше закрывающих скобок, чем открывающих. Такое слово не может быть правильной скобочной структурой, каким бы ни был его конец. В любой момент обработки слова в стеке содержится столько букв a, каков скобочный итог (т.е., разность между числом левых и числом правых скобок) обработанной части слова. Состояние B является принимающим только, если стек пустой. Если же конечное состояние автомата это состояние D или B при непустом стеке, то слово не принято.

Рассмотрим еще один пример распознавания языка конечным автоматом со стеком. Пусть $\mathcal{P}'$ — язык двухбуквенных палиндромов над алфавитом $\{a, b, c\}$, т.е слов нечетной длины, средняя буква в которых есть c, других букв c нет, причем слово одинаково читается слева направо и справа налево. Язык $\mathcal{P}'$ начинается со слов

$$c, aca, bcb, aacaa, abcba, bacab, bbcbb, \ldots.$$

Язык $\mathcal{P}'$ можно распознать посредством следующего автомата со стеком:

	a, в стеке a	b, в стеке b	c	a, в стеке не a	b, в стеке не b
B	B, положить a в стек	B, положить b в стек	C	B, положить a в стек	B, положить b в стек
C	C, снять верхушку стека	C, снять верхушку стека	D	D	D
D	D	D	D	D	D

Состояния B и C являются принимающими только, если стек пуст. Во всех остальных случаях они отвергают входное слово, так же, как и состояние D — вне зависимости от состояния стека. Автомат переходит в состояние C, когда во входном слове впервые встретилась буква c. После этого он либо остается в состоянии C, либо переходит в состояние D — если во входном слове обнаружилось нарушение палиндромности.

10.4 Задачи

Задача 10.1. Выпишите последовательность состояний автомата с рис. 10.2 при обработке строки a) *abaabbaab*; b) *baabaaaba*.

Задача 10.2. Для каждого из автоматов с рис. постройте минимальный автомат, принимающий тот же язык.

Задача 10.3. Покажите, что все слова, не являющиеся началами никаких слов из данного языка L, L-неразличимы.

Задача 10.4. Для каждого из следующих языков докажите, что они нерегулярны: a) $\{a^n b a^{2n} | n \geq 0\}$; b) $\{a^j b^j a^k | i + j < k\}$; c) язык слов в алфавите $\{a, b\}$, в которых никакое начало не содержит больше букв b, чем a.

Задача 10.5. Для каждого из следующих языков над алфавитом $\{a, b\}$ решите, регулярен он или нет. Постройте конечные автоматы для распознавания регулярных языков и докажите нерегулярность остальных. a) язык слов, начинающихся с ww, где w — произвольное слово ненулевой длины; b) язык слов, содержащих внутри себя слово ww, где w — произвольное слово ненулевой длины; c) язык слов нечетной длины, средняя буква в которых a; d) язык слов четной длины, средние буквы в которых одинаковы; e) язык слов вида xyx, где x — произвольное слово ненулевой длины; f) язык слов, не являющихся палиндромами.

Задача 10.6. Постройте конечный автомат, распознающий слова, длина которых кратна трем.

Задача 10.7. Выпишите последовательность состояний автомата со стеком, распознающего язык Дика, при обработке строки a) *aababbab*; b) *abaabbbaba*.

Задача 10.8. Постройте конечный автомат со стеком, распознающий язык над алфавитом $\{a, b\}$, состоящим из слов, в которых букв a больше, чем букв b.

Задача 10.9. Выпишите таблицы перехода для автоматов со стеками, распознающих следующие языки над алфавитом$\{a, b\}$: a) язык слов, в которых одинаковое количество букв a и b; b) язык слов, в которых различное количество букв a и b; c) язык слов вида $\{a^n b^{n+m} a^m | m, n \geq 0\}$.