Глава 3

КОНЕЧНЫЕ АВТОМАТЫ И РЕГУЛЯРНЫЕ ГРАММАТИКИ

§ 3.1. Конечный автомат

В гл. 2 мы познакомились со схемой порождения — грамматиками. Грамматики являются конечными описаниями языков. В этой главе мы рассмотрим другой метод конечного описания бесконечных языков — с помощью *распознавателей*, наипростейшим примером которых являются *конечные автоматы*.

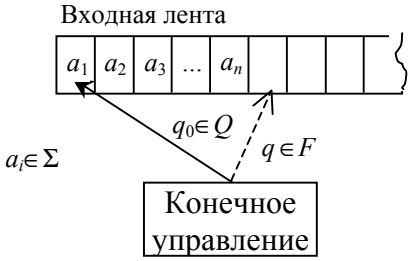

Рис.3.1.

Конечный автомат (рис. 3.1) состоит из конечного управления и входной ленты, разбитой на ячейки. В каждой ячейке записан один символ из входного алфавита Σ, и все они образуют конечную входную цепочку. Конечное управление первоначально находится в состоянии q_0 и сканирует крайнюю левую ячейку ленты. По мере чтения входной цепочки слева направо автомат переходит в другие состояния из множества Q. Если, прочитав входную цепочку, автомат оказывается в некотором конечном состоянии из множества F, то говорят, что он принял ее. Множество цепочек, принимаемых конечным автоматом, называется *языком*, *распознаваемым данным конечным автоматом*.

Конечные автоматы не могут распознавать все языки, порождаемые грамматиками, но языки, распознаваемые ими, являются в точности языками, порождаемыми грамматиками типа 3. В последующих главах мы познакомимся с распознавателями для языков типа 0, 1 и 2. В дальнейшем вместо термина “конечный автомат” будем использовать аббревиатуру fa — *finite automaton*.

Здесь мы определим конечный автомат как формальную систему и выясним его возможности как распознающего устройства.

***Определение* 3.1.** *Конечным автоматом* называется формальная система $M = (Q, \Sigma, \delta, q_0, F)$, где Q — конечное непустое множество *состояний*; Σ — конечный *входной алфавит*; δ — *отображение* типа $Q \times \Sigma \to Q$; $q_0 \in Q$ — *начальное состояние*; $F \subseteq Q$ — множество *конечных состояний*.

Запись $\delta(q, a) = p$, где $q, p \in Q$ и $a \in \Sigma$, означает, что конечный автомат M в состоянии q, сканируя входной символ a, продвигает свою входную головку на одну ячейку вправо и переходит в состояние p.

Область определения отображения δ можно расширить до $Q \times \Sigma^*$ следующим образом: $\delta'(q, \varepsilon) = q$, $\delta'(q, xa) = \delta(\delta'(q, x), a)$ для любого $x \in \Sigma^*$ и $a \in \Sigma$. Таким образом, запись $\delta'(q, x) = p$ означает, что fa M, начиная в состоянии $q \in Q$ чтение цепочки $x \in \Sigma^*$, записанной на входной ленте, оказывается в состоянии $p \in Q$, когда его входная головка продвинется правее цепочки x.

Далее мы будем использовать одно и то же обозначение δ для обоих отображений, так как это не приведет к путанице.

Определенная таким образом модель конечного автомата называется *детерминированной*. Для обозначения детерминированного автомата часто используют аббревиатуру dfa.

***Определение* 3.2.** Цепочка $x \in \Sigma^*$ *принимается конечным автоматом* M, если $\delta(q_0, x) = p$ для некоторого $p \in F$.

Множество всех цепочек $x \in \Sigma^*$, принимаемых конечным автоматом M, называется *языком*, *распознаваемым конечным автоматом* M, и обозначается как $T(M)$, т. е.

$$T(M) = \{x \in \Sigma^* \mid \delta(q_0, x) = p \text{ при некотором } p \in F\}.$$

Любое множество цепочек, принимаемых конечным автоматом, называется *регулярным*.

Пример 3.1. Рассмотрим диаграмму состояний конечного автомата. Пусть задан конечный автомат $M = (Q, \Sigma, \delta, q_0, F)$, где $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$, $F = \{q_0\}$, $\delta(q_0, 0) = q_2$, $\delta(q_0, 1) = q_1$, $\delta(q_1, 0) = q_3$, $\delta(q_1, 1) = q_0$,
$\delta(q_2, 0) = q_0$, $\delta(q_2, 1) = q_3$, $\delta(q_3, 0) = q_1$, $\delta(q_3, 1) = q_2$.

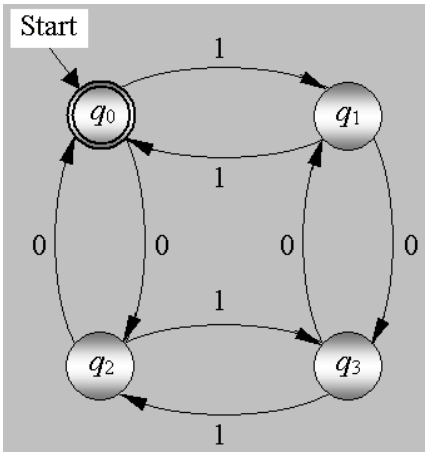

Рис. 3.2.

Диаграмма состояний конечного автомата состоит из узлов, представляющих состояния, и из ориентированных дуг, определяющих возможные переходы, которые зависят от входных символов. Так, если $\delta(q, a) = p$, то из узла, представляющего состояние q, в узел, представляющий состояние p, проводится дуга, помеченная входным символом a. На рис. 3.2 дана диаграмма состояний

конечного автомата M. Двойным кружком выделено единственное в данном примере конечное состояние, которое является одновременно и начальным.

Предположим, что на входе автомата находится цепочка 110101. Поскольку $\delta(q_0, 1) = q_1$, а $\delta(q_1, 1) = q_0$ и $q_0 \in F$, то цепочка 11 находится в языке, распознаваемом данным конечным автоматом, т.е. в $T(M)$, но мы интересуемся всей входной цепочкой. Сканируя остаток 0101 входной цепочки, автомат переходит последовательно в состояния q_2, q_3, q_1, q_0. Поэтому $\delta(q_0, 110101) = q_0$ и потому цепочка 110101 тоже находится в $T(M)$.

Легко показать, что $T(M)$ есть множество всех цепочек из $\{0, 1\}^*$, содержащих четное число нулей и четное число единиц. В частности, $\varepsilon \in T(M)$.

§ 3.2. Отношения эквивалентности и конечные автоматы

Напомним несколько понятий, относящихся к бинарным отношениям, которые потребуются в дальнейшем для описания свойств конечных автоматов.

***Определение* 3.3.** *Бинарное отношение* R на множестве S есть множество пар элементов S.

Если $(a, b) \in R$, то по-другому это записывают как aRb. Нас будут интересовать отношения на множествах цепочек над конечным алфавитом.

***Определение* 3.4.** Говорят, что бинарное отношение R на множестве S *рефлексивно*, если для каждого $s \in S$ имеет место sRs; *симметрично*, если для $s, t \in S$ sRt влечет tRs; *транзитивно*, если для $s, t, u \in S$ из sRt и tRu следует sRu.

Отношение, которое рефлексивно, симметрично и транзитивно, называется *отношением эквивалентности*.

Следующая теорема говорит о важном свойстве отношений эквивалентности.

Теорема 3.1. *Если R — отношение эквивалентности на множестве S, то S можно разбить на k непересекающихся подмножеств, называемых классами эквивалентности, так что aRb тогда и только тогда, когда a и b находятся в одном и том же подмножестве.*

Доказательство. Определим $[a]$ как $\{b \mid aRb\}$. Для любых a и b из множества S имеет место одно из двух соотношений: либо $[a] = [b]$, либо $[a] \cap [b] = \varnothing$. Начнем доказательство от противного.

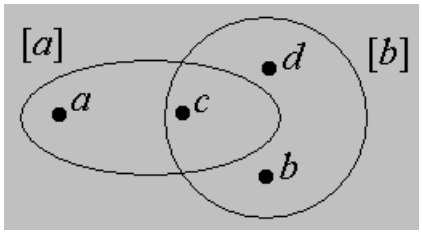

Рис. 3.3.

Пусть $[a] \neq [b]$ и $[a] \cap [b] \neq \varnothing$. Из $[a] \neq [b]$ следует, что существует $d \in S$, такое, что $d \notin [a]$ и $d \in [b]$ или (что то же самое) $d \,\overline{R}\, a$ и dRb.

Кроме того, из $[a] \cap [b] \neq \varnothing$ следует, что существует $c \in S$, такое, что $c \in [a]$ и $c \in [b]$ (рис. 3.3) или (что то же самое) cRa и cRb.

Итак, имеем dRb, cRb или по симметричности — bRc и, наконец, cRa. Из dRb, bRc и cRa по транзитивности получаем dRa. Но это противоречит предыдущему выводу: $d\overline{R}a$.

Отдельные множества $[a]$ для $a \in S$ являются *классами эквивалентности*. Ясно, что элементы a и b находятся в одном и том же множестве — классе эквивалентности тогда и только тогда, когда они эквивалентны, т.е. когда aRb.

***Определение* 3.5.** *Индекс отношения эквивалентности* R, заданного на множестве S, есть число образуемых им классов эквивалентности.

Замечание 3.1. Очевидно, что если S — конечно, то индекс отношения эквивалентности k не может быть бесконечным. В общем случае, когда S — бесконечно, k может быть как конечным, так и бесконечным.

Рассмотрим конечный автомат, приведенный в примере 3.1. Определим отношение R на множестве $\{0, 1\}^*$ следующим образом: $(x, y) \in R$ тогда и только тогда, когда $\delta(q_0, x) = \delta(q_0, y)$. Отношение R рефлексивно, симметрично и транзитивно, т.е. R — отношение эквивалентности. Отношение R делит множество $\{0, 1\}^*$ на четыре класса эквивалентности, соответствующие четырем состояниям автомата. Кроме того, если xRy, то для всех $z \in \{0,1\}^*$ имеет место $xzRyz$, поскольку $\delta(q_0, xz) = \delta(\delta(q_0, x), z) = \delta(\delta(q_0, y), z) = \delta(q_0, yz)$. Такое отношение эквивалентности называется *правоинвариантным*.

Теорема 3.2. *Следующие три утверждения эквивалентны*:

1) *язык* $L \subseteq \Sigma^*$ *принимается некоторым конечным автоматом*;

2) *язык* L *есть объединение некоторых классов эквивалентности правоинвариантного отношения эквивалентности конечного индекса*;

3) *пусть отношение эквивалентности* R *определяется следующим образом*: xRy *тогда и только тогда, когда для всех* $z \in \Sigma^*$ $xz \in L$ *точно тогда, когда* $yz \in L$. *Тогда* R *имеет конечный индекс.*

Доказательство.

1) $\rightarrow$ 2). Предположим, что язык L принимается некоторым конечным автоматом $M' = (Q', \Sigma', \delta', q_0', F')$. Пусть R' — отношение эквивалентности, определяемое следующим образом: $xR'y$ тогда и только тогда, когда $\delta'(q_0', x) = \delta'(q_0', y)$. Отношение R' правоинвариантно, поскольку, если $\delta'(q_0', x) = \delta'(q_0', y)$, то для любого $z \in \Sigma^*$ имеем $\delta'(q_0', xz) = \delta'(\delta'(q_0', x), z) = \delta'(\delta'(q_0', y), z) = \delta'(q_0', yz)$.

Индекс отношения R' конечен, поскольку (самое большее) он равен числу состояний fa M'. Кроме того, язык L есть объединение тех классов эквивалентности, которые включают элемент x, такой, что $\delta'(q_0', x) = p$, где $p \in F'$.

2) $\rightarrow$ 3). Покажем сначала, что любое отношение эквивалентности R', для которого выполняется утверждение 2, является уточнением отношения R, т.е.

каждый класс эквивалентности R' целиком содержится в некотором классе эквивалентности R. Если это так, то индекс отношения R не может быть больше индекса отношения R'. Индекс отношения R', как было показано, конечен. Следовательно, индекс отношения R тоже конечен.

Пусть $x R' y$. Так как отношение R' правоинвариантно, то для любого $z \in \Sigma^*$ имеет место $xz R' yz$ и, таким образом, $xz \in L$ точно тогда, когда $yz \in L$, т.е. $x R y$. Следовательно, $R' \subseteq R$, и потому $[x]_{R'} \subseteq [x]_R$. Это и значит, что любой класс эквивалентности отношения эквивалентности R' содержится в некотором классе эквивалентности отношения эквивалентности R.

Из этого следует, что индекс отношения эквивалентности R не может быть больше индекса отношения эквивалентности R'. Индекс отношения эквивалентности R' по предположению конечен. Следовательно, индекс отношения эквивалентности R тоже конечен.

3) $\rightarrow$ 1). Пусть $x R y$. Тогда для любых $w, z \in \Sigma^*$ цепочка $xwz \in L$ в точности тогда, когда цепочка $ywz \in L$. Следовательно, $xw R yw$, и потому R — правоинвариантно.

Построим конечный автомат $M = (Q, \Sigma, \delta, q_0, F)$, где в качестве Q возьмем конечное множество классов эквивалентности R, т.е. $Q = \{[x]_R \mid x \in \Sigma^*\}$; положим $\delta([x]_R, a) = [xa]_R$, и это определение непротиворечиво, так как R — правоинвариантно; положим $q_0 = [\varepsilon]$ и $F = \{[x]_R \mid x \in L\}$.

Очевидно, что конечный автомат M принимает язык L, поскольку $\delta(q_0, x) = \delta([\varepsilon]_R, x) = [x]_R$, и, таким образом, $x \in T(M)$ тогда и только тогда, когда $[x]_R \in F$.

Теорема 3.3. *Конечный автомат с минимальным числом состояний, принимающий язык L, единствен с точностью до изоморфизма* (*т.е. переименования состояний*) *и есть fa M из теоремы* 3.2.

Доказательство. При доказательстве теоремы 3.2 мы установили, что любой конечный автомат $M' = (Q', \Sigma', \delta', q_0', F')$, принимающий язык L, индуцирует отношение эквивалентности R', индекс которого не меньше индекса отношения эквивалентности R, определенного при формулировке утверждения 3 предыдущей теоремы. Поэтому число состояний fa M' больше или равно числу состояний fa M, построенного в третьей части доказательства теоремы 3.2.

Если M' — fa с минимальным числом состояний, то число его состояний равно числу состояний fa M и между состояниями M' и M можно установить одно-однозначное соответствие.

Действительно, пусть $q' \in Q'$. Должна существовать некоторая цепочка $x \in \Sigma^*$, такая, что $\delta'(q_0', x) = q'$, ибо в противном случае состояние q' без какого-нибудь ущерба для языка, принимаемого этим автоматом, можно было бы исключить из множества состояний Q' как недостижимое. Отбросив такое недостижимое состояние, мы получили бы автомат с меньшим числом состояний, ко-

торый принимал бы все тот же язык. Но это противоречило бы предположению, что M' является конечным автоматом с минимальным числом состояний.

Пусть $q' \in Q'$ и $q' = \delta'(q_0', x)$. Сопоставим с состоянием $q' \in Q'$ состояние $q \in Q$, достижимое автоматом M по прочтении той же цепочки x: $q = \delta(q_0, x) = \delta([\varepsilon]_R, x) = [x]_R$. Это сопоставление является непротиворечивым. Действительно, если q', $p' \in Q'$ и $q' = p'$, причем $q' = \delta'(q_0', x)$, а $p' = \delta'(q_0', y)$, то их образы есть соответственно $q = \delta(q_0, x) = \delta([\varepsilon]_R, x) = [x]_R$ и $p = \delta(q_0, y) = \delta([\varepsilon]_R, y) = [y]_R$. Учитывая, что x и y принадлежат одному и тому же классу эквивалентности отношения R' и что $R' \subseteq R$, заключаем, что x и y также находятся в одном и том же классе эквивалентности отношения R, т.е. $[x]_R = [y]_R$, и потому $q = p$. Другими словами, если прообразы состояний равны, то равны и их образы.

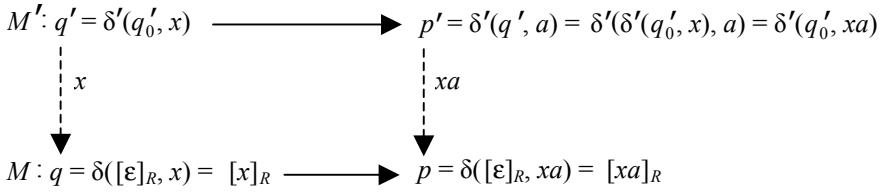

Рис. 3.4.

Кроме того, если fa M' совершает переход из состояния q' в состояние p', прочитав символ $a \in \Sigma$, то fa M переходит из состояния q, являющегося образом q', в состояние p, являющееся образом p', прочитав тот же самый символ $a \in \Sigma$, так как $\delta([x]_R, a) = [xa]_R$ (рис. 3.4).

§ 3.3. Недетерминированные конечные автоматы

Теперь мы введем понятие недетерминированного конечного автомата (ndfa — nondeterministic finite automaton). От своего детерминированного аналога он отличается только типом управляющего отображения (δ). Мы увидим, что любое множество, принимаемое недетерминированным конечным автоматом, может также приниматься детерминированным конечным автоматом. Но недетерминированный конечный автомат является полезным понятием при доказательстве теорем. Кроме того, с этого простейшего понятия легче начать знакомство с недетерминированными устройствами, которые не эквивалентны своим детерминированным аналогам.

***Определение* 3.6.** *Недетерминированным конечным автоматом* называется формальная система $M = (Q, \Sigma, \delta, q_0, F)$, где Q — конечное непустое множество *состояний*; Σ — *входной алфавит*; δ — *отображение* типа $Q \times \Sigma \to 2^Q$, $q_0 \in Q$ — *начальное состояние*; $F \subseteq Q$ — множество *конечных состояний*.

Существенная разница между детерминированной и недетерминированной моделями конечного автомата состоит в том, что значение $\delta(q, a)$ является (возможно пустым) множеством состояний, а не одним состоянием.

Запись $\delta(q, a) = \{p_1, p_2, \ldots, p_k\}$ означает, что недетерминированный конечный автомат M в состоянии q, сканируя символ a на входной ленте, продвигает входную головку вправо к следующей ячейке и выбирает любое из состояний $p_1, p_2, \ldots, p_k$ в качестве следующего.

Область определения δ может быть расширена на $Q \times \Sigma^*$ следующим образом:

$$\delta(q, \varepsilon) = \{q\},\ \delta(q, xa) = \bigcup_{p \in \delta(q, x)} \delta(p, a) \text{ для каждого } x \in \Sigma^* \text{ и } a \in \Sigma.$$

Область определения δ может быть расширена далее до $2^Q \times \Sigma^*$ следующим образом:

$$\delta(\{p_1, p_2, \ldots, p_k\}, x) = \bigcup_{i=1}^{k} \delta(p_i, x).$$

***Определение* 3.7.** Цепочка $x \in \Sigma^*$ *принимается недетерминированным конечным автоматом* M, если существует состояние p, такое, что $p \in F$ и $p \in \delta(q_0, x)$. Множество всех цепочек x, принимаемых ndfa M, обозначается $T(\mathrm{M})$.

Замечание 3.2. Напомним, что 2^Q, где Q — любое множество, обозначает степенное множество или множество всех подмножеств Q.

Пример 3.2. Рассмотрим недетерминированный конечный автомат, который распознает множество $\{0,1\}^*\{00,11\}\{0,1\}^*$:

$M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \delta, q_0, \{q_2, q_4\})$, где

$\delta(q_0, 0) = \{q_0, q_3\}$, $\delta(q_0, 1) = \{q_0, q_1\}$, $\delta(q_1, 0) = \varnothing$, $\delta(q_1, 1) = \{q_2\}$, $\delta(q_2, 0) = \{q_2\}$, $\delta(q_2, 1) = \{q_2\}$, $\delta(q_3, 0) = \{q_4\}$, $\delta(q_3, 1) = \varnothing$, $\delta(q_4, 0) = \{q_4\}$, $\delta(q_4, 1) = \{q_4\}$.

На рис. 3.5 приведена диаграмма состояний этого автомата. Фактически он принимает любые цепочки, составленные из нулей и единиц, в которых встречаются два подряд идущих нуля или единицы.

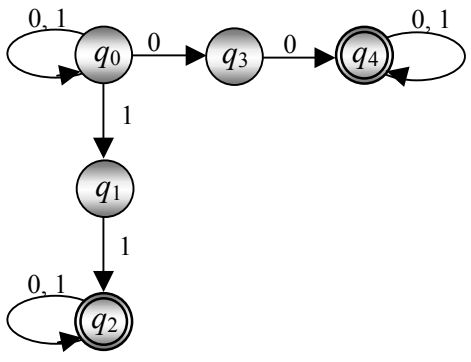

Рис. 3.5.

Теорема 3.4. *Пусть L — множество, принимаемое недетерминированным конечным автоматом. Тогда существует детерминированный конечный автомат, который принимает L.*

Доказательство. Пусть $M = (Q, \Sigma, \delta, q_0, F)$ — ndfa и $L = T(M)$. Определим dfa $M' = (Q', \Sigma, \delta', q_0', F')$ следующим образом. Положим $Q' = \{[s] \mid s \in 2^Q\}$. Состояние из множества Q' будем представлять в виде $[q_1, q_2, \ldots, q_i]$, где $q_1, q_2, \ldots, q_i$ — состояния из множества Q. Будем использовать обозначение φ для случая $i = 0$ или, что то же самое, $s = \varnothing$. Начальное состояние $q_0' = [q_0]$. Таким образом, dfa M' будет хранить след всех состояний, в которых ndfa M мог бы быть в любой данный момент. Пусть F' — множество всех состояний из Q', содержащих хотя бы одно состояние из множества конечных состояний F. Входной алфавит Σ — такой же, как в данном ndfa M.

Определим $\delta'([q_1, q_2, \ldots, q_i], a) = [p_1, p_2, \ldots, p_j]$ тогда и только тогда, когда $\delta(\{q_1, q_2, \ldots, q_i\}, a) = \{p_1, p_2, \ldots, p_j\}$.

Индукцией по длине l входной цепочки $x \in \Sigma^*$ легко показать, что $\delta'(q_0', x) = [q_1, q_2, \ldots, q_i]$ тогда и только тогда, когда $\delta(q_0, x) = \{q_1, q_2, \ldots, q_i\}$.

База. Пусть $l = 0$. Утверждение выполняется, ибо $\delta'(q_0', \varepsilon) = q_0' = [q_0]$ и $\delta(q_0, \varepsilon) = \{q_0\}$.

Индукционная гипотеза. Предположим, что утверждение выполняется для всех $l \le n$ $(n \ge 0)$.

Индукционный переход. Докажем, что тогда утверждение выполняется и для $l = n + 1$.

Пусть $x = za$, где $z \in \Sigma^*$, $|z| = n$, $a \in \Sigma$. Тогда $\delta'(q_0', x) = \delta'(q_0', za) = \delta'(\delta'(q_0', z), a)$. По индукционному предположению $\delta'(q_0', z) = [p_1, p_2, \ldots, p_j]$ тогда и только тогда, когда $\delta(q_0, z) = \{p_1, p_2, \ldots, p_j\}$. В то же время по построению $\delta'([p_1, p_2, \ldots, p_j], a) = [q_1, q_2, \ldots, q_i]$ тогда и только тогда, когда $\delta(\{p_1, p_2, \ldots, p_j\}, a) = \{q_1, q_2, \ldots, q_i\}$. Таким образом, $\delta'(q_0', x) = \delta'(q_0', za) = \delta'([p_1, p_2, \ldots, p_j], a) = [q_1, q_2, \ldots, q_i]$ тогда и только тогда, когда $\delta(q_0, x) = \delta(q_0, za) = \delta(\{p_1, p_2, \ldots, p_j\}, a) = \{q_1, q_2, \ldots, q_i\}$. Чтобы закончить доказательство, остается добавить, что $\delta'(q_0', x) \in F'$ точно тогда, когда $\delta(q_0, x)$ содержит состояние из множества конечных состояний F. Следовательно, $T(M) = T(M')$. Что и требовалось доказать.

Поскольку детерминированные (dfa) и недетерминированные (ndfa) конечные автоматы распознают одни и те же множества, мы будем называть их общим термином *конечные автоматы* (fa) в тех случаях, когда это различие не существенно.

Пример 3.3. Пусть $M = (\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_1\})$ — ndfa, где $\delta(q_0, 0) = \{q_0, q_1\}$, $\delta(q_0, 1) = \{q_1\}$, $\delta(q_1, 0) = \varnothing$, $\delta(q_1, 1) = \{q_0, q_1\}$.

Построим детерминированный конечный автомат, эквивалентный данному. Положим $M' = (Q', \{0, 1\}, \delta', q_0', F')$. Согласно теореме 3.4 в качестве состоя-

ний детерминированного автомата следует взять все подмножества множества $\{q_0, q_1\}$, включая пустое, т.е. $Q' = \{\varphi, [q_0], [q_1], [q_0, q_1]\}$, причем $q_0' = [q_0]$.

Конечные состояния автомата M' представлены теми подмножествами, которые содержат конечные состояния данного автомата (в нашем случае: q_1), т.е. $F' = \{[q_1], [q_0, q_1]\}$.

Наконец, $\delta'([q_0], 0) = [q_0, q_1]$, $\delta'([q_0], 1) = [q_1]$, $\delta'([q_1], 0) = \varphi$, $\delta'([q_1], 1) = [q_0, q_1]$, $\delta'([q_0, q_1], 0) = [q_0, q_1]$, $\delta'([q_0, q_1], 1) = [q_0, q_1]$, $\delta'(\varphi, 0) = \varphi$, $\delta'(\varphi, 1) = \varphi$.

Поясним, что $\delta'([q_0, q_1], 0) = [q_0, q_1]$, так как $\delta(q_0, 0) = \{q_0, q_1\}$, $\delta(q_1, 0) = \varnothing$, и $\{q_0, q_1\} \cup \varnothing = \{q_0, q_1\}$. Аналогично, $\delta'([q_0, q_1], 1) = [q_0, q_1]$, ибо $\delta(q_0, 1) = \{q_1\}$, $\delta(q_1, 1) = \{q_0, q_1\}$ и $\{q_1\} \cup \{q_0, q_1\} = \{q_0, q_1\}$.

§ 3.4. Конечные автоматы и языки типа 3

Теперь мы возвращаемся к связи языков, порождаемых грамматиками типа 3, с множествами, которые принимаются конечными автоматами. Для удобства рассуждений введем понятие *конфигурации конечного автомата*.

***Определение* 3.8.** Пусть $M = (Q, \Sigma, \delta, q_0, F)$ — конечный автомат. *Конфигурацией* конечного автомата M назовем состояние управления в паре с непрочитанной частью входной цепочки.

Пусть (q, ax) — конфигурация fa M, где $q \in Q$, $a \in \Sigma$, $x \in \Sigma^*$, и пусть $p = \delta(q, a)$ в случае, если M — dfa, или $p \in \delta(q, a)$ в случае, когда M — ndfa. Тогда fa M может перейти из конфигурации (q, ax) в конфигурацию (p, x), и этот факт мы будем записывать как $(q, ax) \vdash (p, x)$. Далее символом $\overset{*}{\vdash}$ обозначается рефлексивно-транзитивное замыкание этого отношения на множестве конфигураций. Запись $(q_0, x) \overset{*}{\vdash} (p, \varepsilon)$, где $p \in F$, равнозначна записи $x \in T(M)$.

Теорема 3.5. *Пусть* $G = (V_N, V_T, P, S)$ *— грамматика типа* 3. *Тогда существует конечный автомат* $M = (Q, \Sigma, \delta, q_0, F)$, *такой, что* $T(M) = L(G)$.

Доказательство. Построим ndfa M, о котором идет речь. В качестве состояний возьмем нетерминалы грамматики и еще одно дополнительное состояние $A \notin V_N$. Итак, $Q = V_N \cup \{A\}$. Начальное состояние автомата M есть S. Если множество P содержит правило $S \to \varepsilon$, то $F = \{S, A\}$. В противном случае $F = \{A\}$. Напомним, что начальный нетерминал S не будет появляться в правых частях правил, если $S \to \varepsilon \in P$.

Включим A в $\delta(B, a)$, если $B \to a \in P$. Кроме того, в $\delta(B, a)$ включим все $C \in V_N$, такие, что $B \to aC \in P$. Положим $\delta(A, a) = \varnothing$ для каждого $a \in V_T$.

Построенный автомат M, принимая цепочку x, моделирует ее вывод в грамматике G. Требуется показать, что $T(M) = L(G)$.

I. Пусть $x = a_1 a_2 \ldots a_n$ и $x \in L(G)$, $n \geq 1$. Тогда существует вывод вида

$$S \Rightarrow a_1 A_1 \Rightarrow a_1 a_2 A_2 \Rightarrow \ldots \Rightarrow a_1 a_2 \ldots a_{n-1} A_{n-1} \Rightarrow a_1 a_2 \ldots a_{n-1} a_n,$$

где $A_1, \ldots, A_{n-1} \in V_N$. Очевидно, что в нем используются следующие правила:

$$S \to a_1 A_1,\ A_1 \to a_2 A_2,\ \ldots,\ A_{n-2} \to a_{n-1} A_{n-1},\ A_{n-1} \to a_n \in P.$$

По построению δ

$$A_1 \in \delta(S, a_1),\ A_2 \in \delta(A_1, a_2),\ \ldots,\ A_{n-1} \in \delta(A_{n-2}, a_{n-1}),\ A \in \delta(A_{n-1}, a_n).$$

Следовательно, существует последовательность конфигураций

$$(S, a_1 a_2 \ldots a_n) \vdash (A_1, a_2 \ldots a_n) \vdash \ldots \vdash (A_{n-1}, a_n) \vdash (A, \varepsilon),$$

причем $A \in F$ и потому $x \in T(M)$. Если же $x = \varepsilon$, то $x \in L(G)$, и поскольку в этом случае $(S, \varepsilon) \vdash^* (S, \varepsilon)$, $S \in F$, то $x \in T(M)$.

II. Пусть теперь $x = a_1 a_2 \ldots a_n$ и $x \in T(M)$, $n \geq 1$. Тогда существует последовательность конфигураций вида

$$(S, a_1 a_2 \ldots a_n) \vdash (A_1, a_2 \ldots a_n) \vdash \ldots \vdash (A_{n-1}, a_n) \vdash (A, \varepsilon),$$

где $A \in F$. Очевидно, что

$$A_1 \in \delta(S, a_1),\ A_2 \in \delta(A_1, a_2),\ \ldots,\ A_{n-1} \in \delta(A_{n-2}, a_{n-1}),\ A \in \delta(A_{n-1}, a_n).$$

Но это возможно лишь при условии, что существуют правила

$$S \to a_1 A_1,\ A_1 \to a_2 A_2,\ \ldots,\ A_{n-2} \to a_{n-1} A_{n-1},\ A_{n-1} \to a_n \in P.$$

Используя их, легко построить вывод вида

$$S \Rightarrow a_1 A_1 \Rightarrow a_1 a_2 A_2 \Rightarrow \ldots \Rightarrow a_1 a_2 \ldots a_{n-1} A_{n-1} \Rightarrow a_1 a_2 \ldots a_{n-1} a_n = x,$$

т. е. $x \in L(G)$.

Если же $x = \varepsilon$ и $x \in T(M)$, то $(S, \varepsilon) \vdash^* (S, \varepsilon)$ и $S \in F$. Но это возможно, если только существует правило $S \to \varepsilon \in P$. А тогда $S \Rightarrow \varepsilon$ и $x \in L(G)$. Что и требовалось доказать.

Теорема 3.6. *Пусть $M = (Q, \Sigma, \delta, q_0, F)$ — конечный автомат. Существует грамматика G типа 3, такая, что $L(G) = T(M)$.*

Доказательство. Без потери общности можно считать, что M — dfa. Построим грамматику $G = (V_N, V_T, P, S)$, положив $V_N = Q$, $V_T = \Sigma$, $S = q_0$, $P = \{q \to ap \mid \delta(q, a) = p\} \cup \{q \to a \mid \delta(q, a) = p \text{ и } p \in F\}$. Очевидно, что G — грамматика типа 3.

I. Пусть $x \in T(M)$ и $|x| > 0$. Покажем, что $x \in L(G)$.

Предположим, что $x = a_1 a_2 \ldots a_n$, $n > 0$. Существует последовательность конфигураций автомата M: $(q_0, a_1 a_2 \ldots a_n) \vdash (q_1, a_2 \ldots a_n) \vdash \ldots \vdash (q_{n-1}, a_n) \vdash (q_n, \varepsilon)$, причем $q_n \in F$. Соответственно $\delta(q_0, a_1) = q_1$, $\delta(q_1, a_2) = q_2, \ldots, \delta(q_{n-1}, a_n) = q_n$. По построению в множестве правил P существуют правила вида $q_i \to a_{i+1} q_{i+1}$ ($i = 0, 1, \ldots, n-1$) и правило $q_{n-1} \to a_n$. С их помощью можно построить вывод

$$q_0 \Rightarrow a_1 q_1 \Rightarrow a_1 a_2 q_2 \Rightarrow \ldots \Rightarrow a_1 a_2 \ldots a_{n-1} q_{n-1} \Rightarrow a_1 a_2 \ldots a_n.$$

А это значит, что $x \in L(G)$.

II. Пусть $x \in L(G)$ и $|x| > 0$. Покажем, что $x \in T(M)$.

Предположим, что $x = a_1 a_2 \ldots a_n$, $n > 0$. Существует вывод вида

$$q_0 \Rightarrow a_1 q_1 \Rightarrow a_1 a_2 q_2 \Rightarrow \ldots \Rightarrow a_1 a_2 \ldots a_{n-1} q_{n-1} \Rightarrow a_1 a_2 \ldots a_n.$$

Соответственно существуют правила $q_i \to a_{i+1}q_{i+1}$ ($i = 0, 1, \ldots, n-1$) и правило $q_{n-1} \to a_n$. Очевидно, что они обязаны своим существованием тому, что $\delta(q_i, a_{i+1}) = q_{i+1}$ ($i = 0, 1, \ldots, n-1$) и $q_n \in F$. А тогда существует последовательность конфигураций fa M вида

$$(q_0, a_1a_2 \ldots a_n) \vdash (q_1, a_2 \ldots a_n) \vdash \ldots \vdash (q_{n-1}, a_n) \vdash (q_n, \varepsilon),$$

причем $q_n \in F$. Это значит, что $x \in T(M)$.

Если $q_0 \notin F$, то $\varepsilon \notin T(M)$ и $L(G) = T(M)$. Если $q_0 \in F$, то $\varepsilon \in T(M)$. В этом случае $L(G) = T(M) \setminus \{\varepsilon\}$. По теореме 2.1 мы можем получить из G новую грамматику G_1 типа 3, такую, что $L(G_1) = L(G) \cup \{\varepsilon\} = T(M)$. Что и требовалось доказать.

Пример 3.4. Рассмотрим грамматику типа 3 $G = (\{S, B\}, \{0, 1\}, P, S)$, где $P = \{S \to 0B,\ B \to 0B,\ B \to 1S,\ B \to 0\}$. Мы можем построить ndfa $M = (\{S, B, A\}, \{0, 1\}, \delta, S, \{A\})$, где δ определяется следующим образом:

1) $\delta(S, 0) = \{B\}$, 2) $\delta(S, 1) = \varnothing$,
3) $\delta(B, 0) = \{B, A\}$, 4) $\delta(B, 1) = \{S\}$,
5) $\delta(A, 0) = \varnothing$, 6) $\delta(A, 1) = \varnothing$.

По теореме 3.5 $T(M) = L(G)$, в чем легко убедиться непосредственно.

Теперь мы используем построения теоремы 3.4, чтобы найти dfa M_1, эквивалентный автомату M.

Положим $M_1 = (Q_1, \{0, 1\}, \delta_1, [S], F_1)$, где $Q_1 = \{\varphi, [S], [B], [A], [S, B], [S, A], [B, A], [S, B, A]\}$, $F_1 = \{[A], [S, A], [B, A], [S, B, A]\}$; δ_1 определяется следующим образом:

1) $\delta_1([S], 0) = [B]$, 2) $\delta_1([S], 1) = \varphi$,
3) $\delta_1([B], 0) = [B, A]$, 4) $\delta_1([B], 1) = [S]$,
5) $\delta_1([B, A], 0) = [B, A]$, 6) $\delta_1([B, A], 1) = [S]$,
7) $\delta_1(\varphi, 0) = \varphi$, 8) $\delta_1(\varphi, 1) = \varphi$.

Имеются и другие определения δ_1. Однако ни в какие другие состояния, кроме φ, $[S]$, $[B]$, $[B, A]$ автомат M_1 никогда не входит, и все другие состояния и правила, определяющие и использующие их, могут быть удалены из множеств состояний Q_1, F_1 и δ_1 как бесполезные.

Теперь согласно построениям теоремы 3.6 по автомату M_1 построим грамматику типа 3:

$G_1 = (V_N, V_T, P, S)$, где $V_N = \{\varphi, [S], [B], [B, A]\}$, $V_T = \{0,1\}$, $S = [S]$,

$P = \{(1)\ [S] \to 0[B],\ (2)\ [S] \to 1\varphi,\ (3)\ [B] \to 0[B, A],\ (4)\ [B] \to 1[S],\ (5)\ [B] \to 0,$
$(6)\ [B, A] \to 0[B, A],\ (7)\ [B, A] \to 1[S],\ (8)\ [B, A] \to 0,\ (9)\ \varphi \to 0\varphi,$
$(10)\ \varphi \to 1\varphi\}$.

Грамматика G_1 значительно сложнее грамматики G, но $L(G_1) = L(G)$. Действительно, грамматику G_1 можно упростить, если заметить, что правила для $[B, A]$ порождают в точности те же цепочки, что и правила для $[B]$. Поэтому нетерминал $[B, A]$ можно заменить всюду на $[B]$ и исключить появившиеся дубликаты правил для $[B]$. Кроме того, отметим, что нетерминал φ не порождает ни

одной терминальной цепочки. Поэтому он может быть исключен вместе с правилами, в которые он входит. В результате получим грамматику

$G_2 = (\{[S], [B]\}, \{0, 1\}, P_2, [S])$, где $P_2 = \{(1)\ [S] \to 0[B],\ (2)\ [B] \to 0[B],$
$(3)\ [B] \to 1[S],\ (4)\ [B] \to 0\}$.

Очевидно, что полученная грамматика G_2 отличается от исходной грамматики G лишь обозначениями нетерминалов. Другими словами, $L(G_2) = L(G)$.

§ 3.5. Свойства языков типа 3

Поскольку класс языков, порождаемых грамматиками типа 3, равен классу множеств, принимаемых конечными автоматами, мы будем использовать обе формулировки при описании свойств класса языков типа 3. Прежде всего покажем, что языки типа 3 образуют *булеву алгебру*.

***Определение* 3.9.** *Булева алгебра* множеств есть совокупность множеств, замкнутая относительно объединения, дополнения и пересечения.

***Определение* 3.10.** Пусть $L \subseteq \Sigma_1^*$ — некоторый язык и $\Sigma_1 \subseteq \Sigma_2$. Под *дополнением* $\overline{L}$ языка L подразумевается множество $\Sigma_2^* \setminus L$.

Лемма 3.1. Класс языков типа 3 замкнут относительно объединения.

Доказательство. Возможны два подхода: один использует недетерминированные конечные автоматы, другой основывается на грамматиках. Мы будем пользоваться вторым подходом.

Пусть L_1 и L_2 — языки типа 3, порождаемые соответственно грамматиками типа 3: $G_1 = (V_N^{(1)}, V_T^{(1)}, P_1, S_1)$ и $G_2 = (V_N^{(2)}, V_T^{(2)}, P_2, S_2)$. Можно предположить, что $V_N^{(1)} \cap V_N^{(2)} = \varnothing$, ибо в противном случае этого всегда можно достичь путем переименования нетерминалов данных грамматик. Предположим также, что $S \notin (V_N^{(1)} \cup V_N^{(2)})$.

Построим новую грамматику $G_3 = \left(\{S\} \cup V_N^{(1)} \cup V_N^{(2)},\ V_T^{(1)} \cup V_T^{(2)},\ P_3,\ S\right)$, где $P_3 = (P_1 \cup P_2) \setminus \{S_1 \to \varepsilon,\ S_2 \to \varepsilon\} \cup \{S \to \alpha \mid \exists\ S_1 \to \alpha \in P_1$ или $\exists\ S_2 \to \alpha \in P_2\}$. Очевидно, что $S \underset{G_3}{\Rightarrow} \alpha$ тогда и только тогда, когда $S_1 \underset{G_1}{\Rightarrow} \alpha$ или $S_2 \underset{G_2}{\Rightarrow} \alpha$. В первом случае из α могут выводиться только цепочки в алфавите $V_N^{(1)} \cup V_T^{(1)}$, во втором — только цепочки в алфавите $V_N^{(2)} \cup V_T^{(2)}$. Формально, если $S_1 \underset{G_1}{\Rightarrow} \alpha$, то $\alpha \overset{*}{\underset{G_3}{\Rightarrow}} x$ тогда и только тогда, когда $\alpha \overset{*}{\underset{G_1}{\Rightarrow}} x$. Аналогично, если $S_2 \underset{G_2}{\Rightarrow} \alpha$, то $\alpha \overset{*}{\underset{G_3}{\Rightarrow}} x$ тогда и только тогда, когда $\alpha \overset{*}{\underset{G_2}{\Rightarrow}} x$. Сопоставив все сказанное, заключаем, что $S \overset{*}{\underset{G_3}{\Rightarrow}} x$ тогда и только тогда, когда либо $S_1 \overset{*}{\underset{G_1}{\Rightarrow}} x$, либо $S_2 \overset{*}{\underset{G_2}{\Rightarrow}} x$. А это и значит, что $L(G_3) = L(G_1) \cup L(G_2)$. Что и требовалось доказать.

Лемма 3.2. Класс множеств, принимаемых конечными автоматами (порождаемых грамматиками типа 3), замкнут относительно дополнения.

Доказательство. Пусть $M_1 = (Q, \Sigma_1, \delta_1, q_0, F)$ — dfa и $T(M_1) = S_1$. Пусть Σ_2 — конечный алфавит, содержащий Σ_1, и пусть $d \notin Q$ — новое состояние. Мы построим fa M_2, который принимает $\Sigma_2^* \setminus S_1$.

Положим $M_2 = (Q \cup \{d\}, \Sigma_2, \delta_2, q_0, (Q \setminus F) \cup \{d\})$, где (1) $\delta_2(q, a) = \delta_1(q, a)$ для каждого $q \in Q$ и $a \in \Sigma_1$, если $\delta_1(q, a)$ определено; (2) $\delta_2(q, a) = d$ для тех $q \in Q$ и $a \in \Sigma_2$, для которых $\delta_1(q, a)$ не определено; (3) $\delta_2(d, a) = d$ для каждого $a \in \Sigma_2$.

Интуитивно fa M_2 получается расширением входного алфавита fa M_1 до алфавита Σ_2, добавлением состояния "ловушки" d и затем перестановкой конечных и неконечных состояний. Очевидно, что fa M_2 принимает $\Sigma_2^* \setminus S_1$.

Теорема 3.7. *Класс множеств, принимаемых конечными автоматами, образует булеву алгебру.*

Доказательство непосредственно следует из лемм 3.1 и 3.2 и того факта, что $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$.

Теорема 3.8. *Все конечные множества принимаются конечными автоматами.*

Доказательство. Рассмотрим множество, содержащее только одну непустую цепочку $x = a_1 a_2 \ldots a_n$. Мы можем построить конечный автомат M, принимающий только эту цепочку. Положим $M = (\{q_0, q_1, q_2, \ldots, q_n, p\}, \{a_1, a_2, \ldots, a_n\}, \delta, q_0, \{q_n\})$, где $\delta(q_i, a_{i+1}) = q_{i+1}$, $\delta(q_i, a) = p$, если $a \neq a_{i+1}$ $(i = 0, 1, \ldots, n-1)$, $\delta(q_n, a) = \delta(p, a) = p$ для всех a. Очевидно, что fa M принимает только цепочку x.

Множество, содержащее только пустую цепочку, принимается конечным автоматом $M = (\{q_0, p\}, \Sigma, \delta, q_0, \{q_0\})$, где $\delta(q_0, a) = \delta(p, a) = p$ для всех $a \in \Sigma$. Действительно, только пустая цепочка переведет автомат в состояние q_0, являющееся конечным.

Пустое множество принимается конечным автоматом $M = (\{q_0\}, \Sigma, \delta, q_0, \varnothing)$, где $\delta(q_0, a) = q_0$ для всех $a \in \Sigma$.

Утверждение теоремы немедленно следует из свойства замкнутости языков типа 3 относительно объединения. Что и требовалось доказать.

***Определение* 3.11.** *Произведением* или *конкатенацией* языков L_1 *и* L_2 называется множество $L_1L_2 = \{z \mid z = xy, x \in L_1, y \in L_2\}$. Другими словами, каждая цепочка в языке L_1L_2 есть конкатенация цепочки из L_1 с цепочкой из L_2.

Например, если $L_1 = \{01, 11\}$ и $L_2 = \{1, 0, 101\}$, то множество $L_1L_2 = \{011, 010, 01101, 111, 110, 11101\}$.

Теорема 3.9. *Класс множеств, принимаемых конечными автоматами, замкнут относительно произведения.*

Доказательство. Пусть $M_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1)$ и $M_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$ — детерминированные конечные автоматы, принимающие языки L_1 и L_2 соответственно.

Предположим, что $Q_1 \cap Q_2 = \varnothing$. Кроме того, без потери общности можно предположить, что $\Sigma_1 = \Sigma_2 = \Sigma$ (в противном случае мы могли бы добавить "мертвые" состояния в Q_1 и Q_2, как при доказательстве леммы 3.2).

Мы построим ndfa M_3, принимающий язык L_1L_2. Положим $M_3 = (Q_1 \cup Q_2, \Sigma, \delta_3, q_1, F)$, где

1) $\delta_3(q, a) = \{\delta_1(q, a)\}$ для любого $q \in Q_1 \setminus F_1$,

2) $\delta_3(q, a) = \{\delta_1(q, a), \delta_2(q_2, a)\}$ для любого $q \in F_1$,

3) $\delta_3(q, a) = \{\delta_2(q, a)\}$ для любого $q \in Q_2$.

Если $\varepsilon \notin L_2$, то $F = F_2$, иначе $F = F_1 \cup F_2$.

Правило 1 воспроизводит движения автомата M_1 до тех пор, пока он не достигает какого-нибудь из его конечных состояний, приняв некоторую (возможно пустую) начальную часть входной цепочки, принадлежащую языку L_1. Затем согласно правилу 2 он может продолжать повторять движения автомата M_1 или перейти в режим воспроизведения движений автомата M_2, начиная с его начального состояния. В последнем случае все дальнейшие движения благодаря правилу 3 повторяют движения автомата M_2. Если fa M_2 принимает (возможно пустое) окончание входной цепочки, принадлежащее языку L_2, то и автомат M_3 принимает всю входную цепочку. Другими словами, $T(M_3) = L_1L_2$.

***Определение* 3.12.** *Замыкание языка* L есть множество $L^* = \bigcup_{k=0}^{\infty} L^k$.

Предполагается, что $L^0 = \{\varepsilon\}$, $L^n = L^{n-1}L = LL^{n-1}$ при $n > 0$.

Пример 3.5. Если $L = \{01, 11\}$, то $L^* = \{\varepsilon, 01, 11, 0101, 0111, 1101, 1111, ...\}$.

Теорема 3.10. *Класс множеств, принимаемых конечными автоматами, замкнут относительно замыкания.*

Доказательство. Пусть $M = (Q, \Sigma, \delta, q_0, F)$ — dfa и $L = T(M)$. Построим ndfa M', который принимает язык L^*. Положим $M' = (Q \cup \{q_0'\}, \Sigma, \delta', q_0', F \cup \{q_0'\})$, где $q_0' \notin Q$ — новое состояние, и

$$\delta'(q_0', a) = \begin{cases} \{\delta(q_0, a), q_0\}, \text{ если } \delta(q_0, a) \in F, \\ \{\delta(q_0, a)\} \text{ в противном случае;} \end{cases}$$

$$\delta'(q, a) = \begin{cases} \{\delta(q, a), q_0\}, \text{ если } \delta(q, a) \in F, \\ \{\delta(q, a)\} \text{ в противном случае для всех } q \in Q. \end{cases}$$

Предназначение нового начального состояния q_0' — принимать пустую цепочку. Если $q_0 \notin F$, мы не можем просто сделать q_0 конечным состоянием, поскольку автомат M может снова прийти в состояние q_0, прочитав некоторую непустую цепочку, не принадлежащую языку L.

Докажем теперь, что $T(M') = L^*$.

I. Предположим, что $x \in L^*$. Тогда либо $x = \varepsilon$, либо $x = x_1x_2...x_n$, где $x_i \in L$ ($i = 1, 2, ..., n$). Очевидно, что автомат M' принимает ε. Ясно также, что из $x_i \in L$ следует, что $\delta(q_0, x_i) \in F$. Таким образом, множества $\delta'(q_0', x_i)$ и $\delta'(q_0, x_i)$ каждое со-

держит состояние q_0 и некоторое состояние p (возможно $p = q_0$) из множества F. Следовательно, множество $\delta'(q_0', x)$ содержит некоторое состояние из F и потому $x \in T(M')$.

II. Предположим теперь, что $x = a_1a_2 \ldots a_n \in T(M')$. Это значить, что

$$(q_0', a_1a_2 \ldots a_n) \vdash (q_1', a_2 \ldots a_n) \vdash \ldots \vdash (q_{n-1}', a_n) \vdash (q_n', \varepsilon),$$

причем $q_n' \in F \cup \{q_0'\}$. Ясно, что $q_n' = q_0'$ только в случае $n = 0$. В противном случае существует некоторая подпоследовательность состояний q_{i1}', $q_{i2}', \ldots, q_{im}'$ ($m \geq 1$) такая, что значение $q_{ik}' = q_0$ для всех $k = 1, 2, \ldots, m-1$, а $q_{im}' = q_n' \in F$. Это возможно только, если при некоторых j ($1 \leq j \leq n$) имеет место $q_j' \in \delta'(q_{j-1}', a_j)$ и $\delta(q_{j-1}', a_j) = q_j' \in F$. Поэтому $x = x_1x_2 \ldots x_m$, так что $\delta(q_0, x_k) \in F$ для $1 \leq k \leq m$. Это означает, что $x_k \in L$, а $x \in L^m \subset L^*$. Что и требовалось доказать.

Теорема 3.11. (С. Клини). *Класс множеств, принимаемых конечными автоматами, является наименьшим классом, содержащим все конечные множества, замкнутым относительно объединения, произведения и замыкания.*

Доказательство. Обозначим наименьший класс множеств, принимаемых конечными автоматами, содержащий все конечные множества и замкнутый относительно объединения, произведения и замыкания, через M. То, что класс множеств, принимаемых конечными автоматами, содержит класс M, является непосредственным следствием из леммы 3.1 и теорем 3.8–3.10. Остается показать, что класс M содержит класс множеств, принимаемых конечными автоматами.

Пусть L_1 — множество, принимаемое некоторым конечным автоматом $M = (\{q_1, q_2, \ldots, q_n\}, \Sigma, \delta, q_1, F)$. Пусть R_{ij}^k обозначает множество всех цепочек x, таких, что $\delta(q_i, x) = q_j$, причем, если y является непустым префиксом x, не совпадающим с x, то $\delta(q_i, y) = q_l$, где $l \leq k$. Другими словами, R_{ij}^k есть множество всех цепочек, которые переводят fa M из состояния q_i в состояние q_j, не проходя через какое-либо состояние q_l, где $l > k$. Заметим, что под “прохождением через состояние” мы подразумеваем вход и выход вместе. Но i и j могут быть больше k.

Мы можем определить R_{ij}^k рекурсивно:

$$R_{ij}^k = R_{ij}^{k-1} \cup R_{ik}^{k-1} (R_{kk}^{k-1})^* R_{kj}^{k-1},$$

$$R_{ij}^0 = \{a \mid a \in \Sigma, \delta(q_i, a) = q_j\}.$$

Приведенное определение R_{ij}^k неформально означает, что цепочки, которые переводят автомат M из состояния q_i в состояние q_j без перехода через состояния выше, чем q_k, (1) либо находятся во множестве R_{ij}^{k-1}, т.е. никогда не приводят автомат в состояние столь высокое, как q_k, (2) либо каждая такая цепочка состоит из цепочки во множестве R_{ik}^{k-1} (которая переводит автомат M в состояние q_k первый раз), за которой следует сколько-то цепочек из множества R_{kk}^{k-1}, переводящих автомат M из состояния q_k снова в состояние q_k без перехода через состояние q_k и высшие состояния, за которыми следует цепочка из множества

R_{kj}^{k-1}, переводящая автомат M из состояния q_k в состояние q_j, который при этом опять же не достигает состояния q_k и не проходит состояний с большими номерами.

Индукцией по параметру k мы можем показать, что множество R_{ij}^{k} для всех i и j находятся в пределах класса M.

База. Пусть $k = 0$. Утверждение очевидно, поскольку все множества R_{ij}^{0} являются конечными.

Индукционная гипотеза. Предположим, что утверждение выполняется для всех k, таких, что $0 \le k \le m$ ($0 \le m < n$).

Индукционный переход. Докажем, что утверждение верно и для $k = m + 1$. Это так, поскольку R_{ij}^{m+1} выражается через объединение, конкатенацию и замыкание различных множеств вида R_{pq}^{m}, каждое из которых по индукционному предположению находится в классе M.

Остается заметить, что $L_1 = \bigcup\limits_{q_j \in F} R_{1j}^{n}$.

Таким образом, множество L_1 находится в классе M — наименьшем классе множеств, содержащем все конечные множества, замкнутом относительно объединения, конкатенации и замыкания. Что и требовалось доказать.

***Следствие* 3.1** (из теоремы Клини). Из теоремы 3.11 следует, что любое выражение, построенное из конечных подмножеств множества Σ^*, где Σ — конечный алфавит, и конечного числа операций объединения ‘$\cup$‘, произведения ‘.’ и замыкания ‘*’ со скобками, которые определяют порядок действий, обозначает множество, принимаемое некоторым конечным автоматом. И наоборот, каждое множество, принимаемое некоторым конечным автоматом, может быть представлено в виде такого выражения. Это обеспечивает нас хорошим средством для описания регулярных множеств. Оно называется *регулярным выражением*.

Пример 3.5: числа языка Паскаль. Пусть $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Тогда любое число языка Паскаль можно представить в виде следующего регулярного выражения:

$$D^{+}(\{.\}D^{+} \cup \{\varepsilon\})\,(\{e\}(\{+, -\} \cup \{\varepsilon\})\,D^{+} \cup \{\varepsilon\}).$$

Здесь использован символ *плюс Клини* ($^+$), определяемый следующими равенствами:

$$A^{+} = A^{*}A = A\,A^{*} = \bigcup_{k=1}^{\infty} A^{k}.$$

Напомним, что *звездочка Клини* (*), обозначающая замыкание, определяется следующим равенством:

$$A^{*} = \bigcup_{k=0}^{\infty} A^{k}.$$

Члены $\{\varepsilon\}$ обеспечивают необязательность дробной части и порядка (минимальная цепочка, представляющая число на языке Паскаль, состоит из одной цифры).

§ 3.6. Алгоритмически разрешимые проблемы, касающиеся конечных автоматов

В этом параграфе мы покажем, что существуют алгоритмы, отвечающие на многие вопросы, касающиеся конечных автоматов и языков типа 3.

Теорема 3.12. *Множество цепочек, принимаемых конечным автоматом с n состояниями*,

1) *не пусто тогда и только тогда, когда он принимает цепочку длиной, меньше n*;

2) *бесконечно тогда и только тогда, когда он принимает цепочку длиной* l, $n \le l < 2n$.

Доказательство.

Необходимость условия 1 вытекает из следующих рассуждений от противного. Предположим, что множество $T(M) \neq \varnothing$, но ни одной цепочки длиной меньше n в этом множестве не существует. Пусть $x \in T(M)$, где $M = (Q, \Sigma, \delta, q_0, F)$ — конечный автомат с n состояниями, и $|x| \ge n$. Пусть x — одна из самых коротких таких цепочек. Очевидно, что существует такое состояние $q \in Q$, что $x = x_1x_2x_3$, где $x_2 \neq \varepsilon$, и $\delta(q_0, x_1) = q$, $\delta(q, x_2) = q$, $\delta(q, x_3) \in F$. Но тогда $x_1x_3 \in T(M)$, поскольку $\delta(q_0, x_1x_3) = \delta(q_0, x_1x_2x_3) \in F$. В то же время $|x_1x_3| < |x_1x_2x_3| = |x|$, но это противоречит предположению, что x — одна из самых коротких цепочек, принимаемых fa M.

Достаточность условия 1 очевидна. Действительно, если конечный автомат принимает цепочку с меньшей длиной, чем n, то множество $T(M)$ уже не пусто (какой бы длины цепочка ни была).

Докажем теперь утверждение 2.

Необходимость условия 2 доказывается способом от противного. Пусть fa M принимает бесконечное множество цепочек, и ни одна из них не имеет длину l, $n \le l < 2n$.

Если бы в множестве $T(M)$ существовали только цепочки длиной $l < n$, то по доказанному язык был бы конечен, но это не так. Поэтому существуют и цепочки длиной $l \ge 2n$. Пусть x — одна из самых коротких цепочек, таких, что $x \in T(M)$ и $|x| \ge 2n$. Очевидно, что существует такое состояние $q \in Q$, что $x = x_1x_2x_3$, где $1 \le |x_2| \le n$, и $\delta(q_0, x_1) = q$, $\delta(q, x_2) = q$, $\delta(q, x_3) \in F$. Но тогда $x_1x_3 \in T(M)$, поскольку $\delta(q_0, x_1x_3) = \delta(q_0, x_1x_2x_3) \in F$ при том, что $|x_1x_3| \ge n$ (ибо $|x| = |x_1x_2x_3| \ge 2n$ и $1 \le |x_2| \le n$). Поскольку по предположению в $T(M)$ цепочек длиной $n \le l < 2n$ не существует, то $|x_1x_3| \ge 2n$. Следовательно, вопреки предположению, что $x = x_1x_2x_3 \in T(M)$ — одна из самых коротких цепочек, длина которой больше или равна $2n$, нашлась более короткая цепочка $x_1x_3 \in T(M)$ и тоже с длиной, большей или равной $2n$. Это противоречие доказывает необходимость условия 2.

Достаточность условия 2 вытекает из следующих рассуждений. Пусть существует $x \in T(M)$, причем $n \le |x| < 2n$. Как и ранее, можем утверждать, что су-

ществует $q \in Q$, $x = x_1x_2x_3$, где $x_2 \neq \varepsilon$, и $\delta(q_0, x_1) = q$, $\delta(q, x_2) = q$, $\delta(q, x_3) \in F$. Но тогда цепочки вида $x_1x_2^ix_3 \in T(M)$ при любом i. Очевидно, что множество $T(M)$ бесконечно. Что и требовалось доказать.

***Следствие* 3.2.** Из доказанной теоремы следует существование алгоритмов, разрешающих вопрос о пустоте, конечности и бесконечности языка, принимаемого любым данным конечным автоматом.

Действительно, алгоритм, проверяющий непустоту языка, может систематически генерировать все цепочки с постепенно увеличивающейся длиной, но меньшей n. Каждая из этих цепочек пропускается через автомат. Либо автомат примет какую-нибудь из этих цепочек, и тогда алгоритм завершится с положительным ответом, либо ни одна из этих цепочек не будет принята, и тогда алгоритм завершится с отрицательным результатом. В любом случае процесс завершается за конечное время.

Алгоритм для проверки бесконечности языка можно построить аналогичным образом, только он должен генерировать и тестировать цепочки длиной от n до $2n - 1$ включительно.

Теорема 3.13. *Существует алгоритм для определения, являются ли два конечных автомата эквивалентными* (*т.е. принимают ли они один и тот же язык*).

Доказательство. Пусть M_1 и M_2 — конечные автоматы, принимающие языки L_1 и L_2 соответственно. По теореме 3.7 множество $(L_1 \cap \bar{L}_2) \cup (\bar{L}_1 \cap L_2)$ принимается некоторым конечным автоматом M_3. Легко видеть, что множество $T(M_3)$ не пусто тогда и только тогда, когда $L_1 \neq L_2$. Следовательно, согласно теореме 3.12 существует алгоритм для определения, имеет ли место $L_1 = L_2$. Что и требовалось доказать.