Глава 1
Основные определения и обозначения

1.1. Множества и операции над ними

Понятия множества и элемента множества являются первичными, т. е. не подлежат определениям в строго математическом смысле.

Обозначают $x \in X$, если элемент x содержится в множестве X (или, другими словами, x принадлежит X).

Обозначают $x \notin X$, если элемент x не содержится в множестве X (или, другими словами, x не принадлежит X).

Множества натуральных и целых чисел будем обозначать соответственно $\mathbb{N}$, $\mathbb{Z}$.

Обычно в приложениях рассматриваются только те множества, которые содержатся в некотором фиксированном множестве, называемом универсальным (*универсумом*) — U. Тогда любое множество X определяется следующим образом:

$$X = \{x \in U \mid P(x)\}, \tag{1.1}$$

где $P(x)$ — некоторый предикат, который выполняется только для элементов множества X. Выражение (1.1) читается так: X — множество, состоящее из всех таких элементов x из U, для которых выполняется $P(x)$.

Замечание 1. Во многих случаях, если это не приводит к двусмысленности, множество U в (1.1) опускают и пишут

$$X = \{x \mid P(x)\}.$$

Замечание 2. В логических выражениях операцию конъюнкции иногда будем заменять запятой, если это не приведет к двусмысленности.

Определение 1 (Подмножество). Множество A называют подмножеством множества B, и пишут $A \subseteq B$, если каждый элемент A содержится в B, т. е.

$$A \subseteq B \;\Leftrightarrow\; (\forall x, (x \in A) \to (x \in B)).$$

Определение 2 (Равенство множеств). Множество A называют равным множеству B, и пишут $A = B$, если каждый элемент A содержится в B, и каждый элемент из B содержится в A, т. е.

$$A = B \;\Leftrightarrow\; (\forall x, (x \in A) \Leftrightarrow (x \in B)).$$

Другими словами множества A и B равны, если $A \subseteq B$ и $B \subseteq A$.

Если равенство множеств A и B не выполняется, пишут $A \neq B$ и говорят, что A не равно B.

Определение 3 (Собственное подмножество). Множество A называют собственным подмножеством множества B, и пишут $A \subset B$, если A является подмножеством B и $A \neq B$, т. е.

$$A \subset B \;\Leftrightarrow\; ((A \subseteq B) \;\&\; (A \neq B)).$$

Определение 4 (Объединение множеств). Множество A называют объединением множеств B и C, и пишут $A = B \cup C$ или $A = B + C$, если оно содержит в себе все те и только те элементы, которые содержатся хотя бы в одном из этих множеств, т. е.

$$A = B \cup C = B + C \;\Leftrightarrow\; A = \{x \mid (x \in B) \vee (x \in C)\}.$$

Определение 5 (Пересечение множеств). Множество A называют пересечением множеств B и C, и пишут $A = B \cap C$, если оно содержит все те и только те элементы, которые содержатся как в B, так и в C, т. е.

$$A = B \cap C \;\Leftrightarrow\; A = \{x \mid (x \in B) \;\&\; (x \in C)\}.$$

Определение 6 (Разность множеств). Множество A называют разностью множеств B и C, и пишут $A = B \backslash C$ или $A = B - C$, если оно содержит все те и только те элементы B, которые не содержатся в C, т. е.

$$A = B \backslash C = B - C \;\Leftrightarrow\; A = \{x \mid (x \in B) \;\&\; (x \notin C)\}.$$

Определение 7 (Пустое множество). Множество, которому не принадлежит ни один элемент, называют пустым множеством и обозначают $\varnothing$.

Считают, что пустое множество является подмножеством любого множества.

Множество всех подмножеств множества A обозначают $\mathscr{P}(A)$:

$$\mathscr{P}(A) = \{B \mid B \subseteq A\}.$$

Определение 8 (Конечное множество). Множество A называют конечным, если оно состоит из конечного числа элементов $n \in \mathbb{N}$. Число n называют мощностью множества A и пишут $n = \#A$

Мощность $\mathscr{P}(A)$ равна $2^{\#A}$.

Определение 9 (Декартово произведение множеств). Декартовым произведением множеств A и B называют множество всех упорядоченных пар (x, y) таких, что $x \in A$ и $y \in B$, т. е.

$$A \times B = \{(x, y) \mid (x \in A) \,\&\, (y \in B)\}.$$

Пример 1. Пусть $A = \{0, 1, 2, \{3, 4\}\}$, $B = \{1, 2, 3, 4\}$. Тогда справедливы следующие соотношения.

$$\begin{gathered}
0 \in A, \quad \{3,4\} \in A, \qquad \{3,4\} \notin B, \\
\{3,4\} \subseteq B, \quad \{3,4\} \subset B, \quad \{1,2,3,4\} \subseteq B, \\
\varnothing \subseteq B, \quad \varnothing \subset B, \\
\#A = 4, \quad \#B = 4, \\
A + B = \{0,1,2,3,4,\{3,4\}\}, \quad A \cap B = \{1,2\}, \quad B - A = \{3,4\},
\end{gathered}$$

$$\begin{aligned}
\mathscr{P}(B) = \{&\varnothing, \{1\}, \{2\}, \{3\}, \{4\}, \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \\
&\{3,4\}, \{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}, \{1,2,3,4\}\}, \\
A \times B = \{&(0,1), (0,2), (0,3), (0,4), (1,1), (1,2), (1,3), (1,4), (2,1), \\
&(2,2), (2,3), (2,4), (\{3,4\},1), (\{3,4\},2), (\{3,4\},3), (\{3,4\},4)\}, \\
B \times B = \{&(1,1), (1,2), (1,3), (1,4), (2,1), (2,2), (2,3), (2,4), \\
&(3,1), (3,2), (3,3), (3,4), (4,1), (4,2), (4,3), (4,4)\}. \quad \square
\end{aligned}$$

1.2. Отношения на множествах

Определение 10 (Бинарное отношение двух множеств). Бинарным отношением двух множеств называется любое подмножество их декартова произведения.

Если R — бинарное отношение множеств A и B, то $R \subseteq A \times B$. В таком случае множество A называют областью определения отношения R, а множество B — областью значений отношения R.

Если $(x, y) \in R$, то говорят, что x и y связаны отношением R и пишут xRy или $y \in R(x)$.

Определение 11 (Бинарное отношение на множестве). Если $R \subseteq A \times A$, то R называют бинарным отношением на множестве A.

Пример 2. Рассмотрим множество $A = \{2, 3, 4, 5, 6, 7, 8, 9\}$. Рассмотрим отношение $\mid$ на множестве A, заданное следующим образом: элементы x и y из множества A находятся в отношении $\mid$, если x является делителем

y (т. е. $2|4$, $3|6$ и т. д.). Тогда

$$| = \{(2,2),(2,4),(2,6),(2,8),(3,3),(3,6),(3,9),\\ (4,4),(4,8),(5,5),(6,6),(7,7),(8,8),(9,9)\}. \quad \square$$

Пример 3. Пусть $A = \{2,3,4,5,6,7,8,9\}$. Рассмотрим отношение $\equiv_3$ на множестве A, заданное следующим образом: элементы x и y из множества A находятся в отношении $\equiv_3$, если остаток от деления $x \bmod 3$ равен остатку от деления $y \bmod 3$ (т. е. $2 \equiv_3 5$, $3 \equiv_3 6$ и т. д.). Тогда

$$\equiv_3 = \{(2,2),(2,5),(2,8),(3,3),(3,6),(3,9),\\ (4,4),(4,7),(5,2),(5,5),(5,8),(6,3),(6,6),(6,9),(7,4),(7,7),\\ (8,2),(8,5),(8,8),(9,3),(9,6),(9,9)\}. \quad \square$$

Определение 12 (Степень отношения). Степень отношения R на множестве A определяют следующим образом:
1) $R^0 = \{(x,x) \mid \forall x \in A\}$;
2) $R^1 = R$;
3) $R^k = \{(x,y) \mid \exists z \in A, xRz, zR^{k-1}y\}$.

Пример 4. Рассмотрим множество $A = \{2,3,4,5,6\}$. Рассмотрим на этом множестве отношение

$$R = \{(2,3),\ (3,4),\ (4,5),\ (5,6)\}.$$

Тогда

$$\begin{aligned} R^0 &= \{(2,2),(3,3),(4,4),(5,5),(6,6)\};\\ R^1 &= \{(2,3),(3,4),(4,5),(5,6)\};\\ R^2 &= \{(2,4),(3,5),(4,6)\};\\ R^3 &= \{(2,5),(3,6)\};\\ R^4 &= \{(2,6)\};\\ R^5 &= \varnothing. \end{aligned} \quad \square$$

Определение 13 (Рефлексивное отношение). Отношение R, заданное на множестве A, называют рефлексивным, если для каждого $x \in A$ имеет место xRx.

Определение 14 (Антирефлексивное отношение). Отношение R, заданное на множестве A, называют антирефлексивным, если для каждого $x \in A$ выполняется $(x,x) \notin R$.

Определение 15 (Транзитивное отношение). Отношение R, заданное на множестве A, называют транзитивным, если для любых $x, y, z \in A$ из выполнения соотношений xRy и yRz следует выполнение xRz.

Определение 16 (Симметричное отношение). Отношение R, заданное на множестве A, называют симметричным, если для любых $x, y \in A$ из выполнения соотношения xRy следует выполнение yRx.

Определение 17 (Антисимметричное отношение). Отношение R, заданное на множестве A, называют антисимметричным, если для любых $x, y \in A$ из выполнения соотношений xRy и yRx следует $x = y$.

Определение 18 (Отношение эквивалентности). Отношение R, заданное на множестве A, называют отношением эквивалентности, если оно рефлексивно, транзитивно и симметрично.

Пример 5. Отношение $\equiv_3$ из примера 3 является отношением эквивалентности, так как оно рефлексивно, транзитивно и симметрично. Отношение $|$ из примера 2 не является отношением эквивалентности: оно рефлексивно, транзитивно, но не симметрично. $\square$

Определение 19 (Класс эквивалентности). Пусть R — отношение эквивалентности на множестве A. Классом эквивалентности, порожденным элементом $a \in A$ называют множество $[a]_R \stackrel{\text{def}}{=} \{x \in A \mid xRa\}$.

Совокупность всех классов эквивалентности отношения R на множестве A обозначают $[A]_R$.

Пример 6. Для отношения $\equiv_3$ в примере 3 можно выделить три класса эквивалентности:

$$[1]_{\equiv_3} = \{4, 7\},$$
$$[2]_{\equiv_3} = \{2, 5, 8\},$$
$$[3]_{\equiv_3} = \{3, 6, 9\}. \qquad \square$$

Определение 20 (Замыкание отношения). Отношение R_C называют замыканием отношения R относительно заданного свойства если

1) R_C обладает заданным свойством;
2) $R \subseteq R_C$;
3) для любого отношения T, для которого выполняются свойства 1 и 2, выполняется соотношение $\#R_C \leqslant \#T$.

Часто рассматривают транзитивные и рефлексивные-транзитивные замыкания отношений, т. е. замыкания отношений относительно соответственно свойства транзитивности, и свойств рефлексивности и транзитивности.

Утверждение 1 (О транзитивном замыкании отношения)**.** *Транзитивным замыканием отношения R на множестве A является отношение*

$$R^+ = \bigcup_{k=1}^{\infty} R^k.$$

Утверждение 2 (О рефлексивном транзитивном замыкании отношения)**.** *Рефлексивным транзитивным замыканием отношения R на множестве A является отношение*

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

Пример 7. Рассмотрим отношение $R = \{(1,2),\ (2,3),\ (3,4),\ (4,5),\ (5,6)\}$ на множестве $A = \{2,3,4,5,6\}$. Результаты возведения в степень R^k для $k < 6$, были приведены в примере 4. Для $k \geqslant 6$ $R^k = \varnothing$. Следовательно,

$$\begin{aligned}
R^+ = \{&(1,2),(2,3),(3,4),(4,5),(5,6),(1,3),(2,4),(3,5),(4,6),\\
&\qquad (1,4),(2,5),(3,6),(1,5),(2,6),(1,6)\};\\
R^* = \{&(1,1),(2,2),(3,3),(4,4),(5,5),\\
&(6,6),(1,2),(2,3),(3,4),(4,5),(5,6),(1,3),(2,4),(3,5),(4,6),\\
&\qquad (1,4),(2,5),(3,6),(1,5),(2,6),(1,6)\}. \quad \square
\end{aligned}$$

Контрольные вопросы

1. Какими средствами можно описать бесконечное множество?
2. Какие операции над множествами вы знаете?
3. Сколько элементов содержится в множестве всех подмножеств конечного множества?
4. Сколько элементов будет содержать декартово произведение двух множеств?
5. Что называется отношением на множестве?
6. Какими свойствами может обладать отношение на множестве?
7. Как определяется замыкание отношения?
8. Что из себя представляет рефлексивное, транзитивное замыкание отношения?
9. Какое отношение называется отношением эквивалентности?

Глава 2
Формальные языки

2.1. Алфавит и цепочки

Понятие символа интуитивно ясно. Строгого математического определения давать ему не будем. Под символом будем понимать букву, спецсимвол или токен, в зависимости от обстоятельств.

Определение 21 (Алфавит). Алфавитом называют конечное непустое множество символов.

Определение 22 (Цепочка). *Словом* или *цепочкой* называют последовательность символов некоторого алфавита.

Пример 8. Цепочка 011101 — цепочка в бинарном алфавите $\{0, 1\}$.

Цепочка $abacdadba$ — цепочка в алфавите $\{a, b, c, d\}$. □

Определение 23 (Пустая цепочка). Пустая цепочка — цепочка, не содержащая ни одного символа. Эту цепочку можно рассматривать как цепочку в любом алфавите.

Пустую цепочку будем обозначать ε.

Определение 24 (Длина цепочки). Длиной цепочки называют число позиций в ней. Длину некоторой цепочки w обозначают $|w|$.

Пример 9. Длина цепочки 011101 обозначается $|011101|$ и равна 6.

Длина цепочки $abacdadba$ — $|abacdadba| = 9$.

Длина цепочки ε — $|\varepsilon| = 0$. □

Определение 25 (Конкатенация). Пусть α и β — цепочки. Тогда $\alpha\beta$ обозначает их конкатенацию (соединение), т. е. цепочку, в которой последовательно записаны сначала символы цепочки α, а затем символы цепочки β.

Пример 10. Пусть $\alpha = 01101$ и $\beta = 110$. Тогда

$$\alpha\beta = 01101110, \quad \beta\alpha = 11001101.$$ □

Для любой цепочки γ справедливы равенства

$$\varepsilon\gamma = \gamma\varepsilon = \gamma.$$

Определение 26 (Подцепочка). Пусть α — цепочка, представимая в виде $\alpha = \beta\gamma\varphi$, где β, γ, φ — некоторые цепочки. Тогда цепочки β, γ и φ называют подцепочками цепочки α. Подцепочку β называют префиксом цепочки α, а φ называют суффиксом цепочки α. Если $\beta \neq \alpha$, то β называют собственным префиксом. Если $\varphi \neq \alpha$, то φ называют собственным суффиксом.

Определение 27 (Степень алфавита). Пусть Σ — некоторый алфавит. Определим Σ^k как множество всевозможных цепочек длины k, состоящих из символов алфавита Σ.

Пример 11. Заметим, что $\Sigma^0 = \{\varepsilon\}$ независимо от алфавита Σ, т. к. ε — единственная цепочка нулевой длины.

Если $\Sigma = \{0, 1\}$, то

$$\begin{aligned}
\Sigma^1 =& \{0, 1\},\\
\Sigma^2 =& \{00, 01, 10, 11\},\\
\Sigma^3 =& \{000, 001, 010, 011, 100, 101, 110, 111\} \ldots
\end{aligned} \qquad \square$$

Множество всех слов над алфавитом Σ обозначают Σ^*.

Это множество можно определить как объединение всевозможных степеней алфавита Σ:

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots = \bigcup_{k=0}^{\infty} \Sigma^k.$$

Множество всех непустых цепочек над алфавитом Σ обозначают Σ^+. По аналогии с Σ^* множество всех непустых цепочек можно определить как

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots = \bigcup_{k=1}^{\infty} \Sigma^k.$$

Определение 28 (Зеркальное отражение цепочки). Цепочку, в которой символы цепочки α выписаны в обратном порядке называют *зеркальным отражением* (*инверсией*) цепочки α. Результат зеркального отражения цепочки α обозначают α^R.

Пример 12.

$$\begin{aligned}
1^R &= 1; & (011101)^R &= 101110;\\
\varepsilon^R &= \varepsilon; & (0123456789)^R &= 9876543210.
\end{aligned} \qquad \square$$

2.2. Языки и операции над ними

Определение 29 (Язык). Множество L цепочек над фиксированным алфавитом Σ называют языком над этим алфавитом.

Заметим, что язык над алфавитом Σ не обязан использовать в своих словах все символы алфавита Σ. То есть, в алфавите могут присутствовать символы, которые не встречаются в словах языка. Поэтому, если известно, что L — язык над Σ, то можно утверждать, что L — язык над любым алфавитом, содержащим все элементы Σ.

Пример 13. Рассмотрим несколько языков.

1. $L = \{\varepsilon,\ 01,\ 0011,\ 000111,\ 00001111,\ \ldots\}$ — язык над алфавитом $\{0, 1\}$, в каждом слове которого k единиц следуют за k нулями, для любого $k \geqslant 0$.
2. $L = \varnothing$ — пустой язык в любом алфавите.
3. $L = \{\varepsilon\}$ — язык в любом алфавите, состоящий лишь из одного пустого слова.
4. $L = \{\varepsilon, 01, 10, 0011, 1100, 1001, 0110, 0101, \ldots\}$ — язык над алфавитом $\{0, 1\}$, в словах которого равное количество единиц и нулей. □

Определение 30 (Конечный язык). Язык L называют конечным, если он содержит конечное число слов. В противном случае язык бесконечный.

Единственное ограничение для множеств, которые могут быть языками, состоит в том, что все алфавиты конечны. Таким образом, хотя языки и могут содержать бесконечное число цепочек, но эти цепочки должны состоять из символов некоторого алфавита.

Так как языки представляют собой множества, то все операции над множествами можно перенести и на языки. Здесь приведем только дополнительные специфические операции.

Определение 31 (Разность языков). *Разностью* двух языков L_1 и L_2 называют третий язык L_3, определенный следующим образом:

$$L_3 = \{\alpha \mid \alpha \in L_1, \alpha \notin L_2\}.$$

Пишут $L_3 = L_1 - L_2$.

Определение 32 (Дополнение языка). *Дополнением* языка L над алфавитом Σ называют язык L', определенный следующим образом:

$$L' = \Sigma^* - L.$$

Пишут $L' = {\sim}L$.

Определение 33 (Конкатенация языков). *Конкатенацией* языков L_1 и L_2 называют третий язык L_3, определенный следующим образом:

$$L_3 = \{\alpha\beta \mid \alpha \in L_1,\ \beta \in L_2\}.$$

Пишут $L_3 = L_1L_2$.

Пример 14. Пусть $L_1 = \{01, 001\}$, $L_2 = \{\varepsilon, 1, 00\}$. Тогда

$$L_1L_2 = \{01, 011, 0100, 001, 0011, 00100\};$$
$$L_2L_1 = \{01, 001, 101, 1001, 0001, 00001\}. \qquad \square$$

Определение 34 (Левое частное). *Левым частным* языков L_1 и L_2 называют третий язык L_3, определенный следующим образом:

$$L_3 = \{\beta \mid \alpha\beta \in L_1, \alpha \in L_2\}.$$

Пишут $L_3 = L_2 \backslash L_1$.

Стоит обратить внимание, что запись $G \backslash H$ обозначает операцию левого частного в случае, если G и H — языки. Если G и H произвольные множества, то такая запись по определению 6 обозначает разность этих множеств. Для операции разности языков G и H используется только обозначение, данное в определении 31, т. е. $G - H$.

Определение 35 (Правое частное). *Правым частным* языков L_1 и L_2 называют третий язык L_3, определенный следующим образом:

$$L_3 = \{\alpha \mid \alpha\beta \in L_1, \beta \in L_2\}.$$

Пишут $L_3 = L_1/L_2$.

Пример 15. Пусть $L_1 = \{01, 001, 1001, 100\}$, $L_2 = \{1, 00, 01\}$. Тогда

$$L_2 \backslash L_1 = \{\varepsilon, 1, 001, 00\};$$
$$L_1/L_2 = \{\varepsilon, 0, 00, 10, 100, 1\}. \qquad \square$$

Определение 36 (Степень языка). Понятие степени некоторого языка L определяют следующим образом:

1) $L^0 = \{\varepsilon\}$;
2) $L^1 = L$;
3) $L^{k+1} = L^kL$.

Пример 16. Если $L = \{01, 001\}$, то

$$\begin{aligned}
L^0 =&\{\varepsilon\};\\
L^1 =&\{01, 001\};\\
L^2 =&\{0101, 01001, 00101, 001001\};\\
L^3 =&\{010101, 0101001, 0100101, 01001001,\\
&0010101, 00101001, 00100101, 001001001\}. \qquad \square
\end{aligned}$$

Определение 37 (Итерация языка). Итерацией языка L называют множество L^* всевозможных слов, составленных из слов языка L. Другими словами

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

Определение 38 (Регулярные операции). Операции объединения, конкатенации и итерации называют *регулярными* операциями.

Определение 39 (Положительная итерация языка). Положительной итерацией языка L называют множество L^+ всех слов, составленных из одного или более слов языка L. Другими словами

$$L^+ = \bigcup_{k=1}^{\infty} L^k.$$

Положительную итерацию языка L можно определить формулой:

$$L^+ = LL^*.$$

Определение 40 (Зеркальное отражение языка). Зеркальным отражением (инверсией) языка L называют множество L^R, составленное из инверсий слов языка L:

$$L^R = \{a \mid a^R \in L\}.$$

Определение 41 (Подстановка). Пусть L — язык над алфавитом Σ. Пусть для каждого символа a из Σ определен $\sigma(a)$ — язык над некоторым алфавитом Σ_a. Доопределим

$$\sigma(\varepsilon) = \varepsilon, \quad \sigma(XY) = \sigma(X)\sigma(Y), \quad X, Y \in \Sigma^*.$$

Такое отображение σ из Σ^* в множество слов над алфавитом Σ', где

$$\Sigma' = \bigcup_{a \in \Sigma} \Sigma_a,$$

называют подстановкой. Для языка L результатом операции подстановки является язык L':

$$L' = \{w \mid w \in \sigma(v), v \in L\}.$$

Пишут $L' = \sigma(L)$.

Пример 17. Пусть над алфавитом $\{0, 1, 2\}$ задан язык $L = \{00, 010, 1210\}$. Пусть для языка L задана подстановка σ:

$$\sigma(0) = \{0, 1\}; \qquad \sigma(1) = \{2\}; \qquad \sigma(2) = \{3, \varepsilon\}.$$

Тогда

$$\sigma(00) = \sigma(0)\sigma(0) = \{0, 1\}\{0, 1\} = \{00, 01, 10, 11\};$$

$$\sigma(010) = \sigma(0)\sigma(1)\sigma(0) = \{0, 1\}\{2\}\{0, 1\} = \{020, 021, 120, 121\};$$

$$\sigma(1210) = \sigma(1)\sigma(2)\sigma(1)\sigma(0) = \{2\}\{3, \varepsilon\}\{2\}\{0, 1\} = \{2320, 2321, 220, 221\}.$$

Объединяя результаты подстановки в слова получим результат подстановки σ в язык L.

$$\sigma(L) = \{00, 01, 10, 11, 020, 021, 120, 121, 220, 221, 2320, 2321\}. \qquad \square$$

Пример 18. Пусть задан язык $L = \{00, 010, 1210, 210, 2220\}$ над алфавитом $\{0, 1, 2\}$. Пусть для языка L задана подстановка σ:

$$\sigma(0) = \{1\}; \qquad \sigma(1) = \{\varepsilon\}; \qquad \sigma(2) = \{0\}.$$

Тогда

$$\begin{gathered}
\sigma(00) = \sigma(0)\sigma(0) = \{1\}\{1\} = \{11\};\\
\sigma(010) = \sigma(0)\sigma(1)\sigma(0) = \{1\}\{\varepsilon\}\{1\} = \{11\};\\
\sigma(1210) = \sigma(1)\sigma(2)\sigma(1)\sigma(0) = \{\varepsilon\}\{0\}\{\varepsilon\}\{1\} = \{01\};\\
\sigma(210) = \sigma(2)\sigma(1)\sigma(0) = \{0\}\{\varepsilon\}\{1\} = \{01\};\\
\sigma(2220) = \sigma(2)\sigma(2)\sigma(2)\sigma(0) = \{0\}\{0\}\{0\}\{1\} = \{0001\}.
\end{gathered}$$

Объединяя результаты подстановки от всех слов исходного языка получим результат подстановки σ в язык L.

$$\sigma(L) = \{01, 11, 0001\}. \qquad \square$$

Определение 42 (Гомоморфизм). Подстановку, в которой для всех $a \in \Sigma$ $\#\sigma(a) = 1$, называют *гомоморфизмом*.

Очевидно, что подстановка в примере 18 является гомоморфизмом.

2.3. Способы задания языков

Любой конечный язык можно задать перечислением всех его слов. Такая процедура невозможна для бесконечных языков. Для таких языков необходима процедура, позволяющая определение языка конечными средствами.

Существуют несколько способов задания языков. Все их можно разделить на три группы: спецификаторы свойств языка, конструкторы и распознаватели.

Определение 43 (Спецификаторы свойств языка). Спецификаторы свойств языка — описание совокупности свойств, которой должны удовлетворять все и только все слова языка.

К спецификаторам свойств языка можно отнести регулярные выражения, описание множеств слов с использованием предикатов.

Определение 44 (Конструктор). Конструктор языка L — набор правил (абстрактное порождающее устройство), с помощью которого можно построить все и только все слова языка L.

Таким образом, с помощью конструктора языка L нельзя построить слово не принадлежащее языку L. К конструкторам языков относятся грамматики, синтаксические диаграммы, формы Бэкуса—Наура.

Определение 45 (Распознаватель). Распознаватель языка L — абстрактное устройство (набор правил проверки), которое при получении на входе любого слова определяет принадлежность этого слова языку L.

К распознавателям можно отнести конечные автоматы, линейно ограниченные автоматы, машины Тьюринга.

Пример 19. Пусть L — язык над алфавитом $\{0,\ 1\}$ определенный следующим образом:

1) $\varepsilon \in L$;
2) Если $\alpha \in L$ то $0\alpha 1 \in L$;
3) Никакие другие слова не принадлежат L.

Можно утверждать, что

$$L = \{0^i 1^i \mid i \geqslant 0\}. \tag{2.1}$$

Действительно, из пункта 1) следует, что $0^0 1^0 \in L$. Предполагая по индукции, что $0^i 1^i \in L$, из 2) получаем $0^{i+1} 1^{i+1} \in L$. Следовательно множество в правой части равенства (2.1) содержится в L. Предполагая, что $0^i 1^i$ единственное слово длины $2i$ в L, приходим к выводу, что $0^{i+1} 1^{i+1}$ единственное слово длины $2(i+1)$ в L. Т. к. L не содержит ни одного слова нечетной длины делаем заключение, что L содержится в правой части (2.1), т. е. равенство (2.1) имеет место.

Заметим, что пункты 1)–3) представляют собой описание порождающего устройства для языка L, а равенство (2.1) — спецификатор свойств языка. □

Пример 20. Пусть L_1 — язык над алфавитом $\{0,\ 1\}$ определенный следующим образом:
1) $\varepsilon \in L_1$;
2) если $\alpha \in L$ то $0\alpha 1 \in L$ и $1\alpha 0 \in L_1$;
3) если $\alpha \in L_1$ и $\beta \in L_1$ то $\alpha\beta \in L_1$;
4) никакие другие слова не принадлежат L_1.

Покажем, что

$$L_1 = \{w \in \{0,1\}^* \mid N_0(w) = N_1(w)\}. \tag{2.2}$$

Действительно, из пункта 1) следует, что ε — слово в котором столько же нулей сколько и единиц, содержится в L_1. Рассуждая по индукции предположим, что L_1 содержит все слова длины $2i$, в которых одинаковое количество нулей и единиц. Пусть w произвольное слово длины $2(i+1)$ с таким же свойством. Без потери общности предположим, что 1-й символ в w — символ 0. Если последний символ в w — 1, то тогда имеем $w = 0w'1$, где w' — слово длины $2i$ с одинаковым числом нулей и единиц. Из 2) и предположения индукции следует, что $w \in L_1$. Если же последний символ в w — 0, то $w = w'w''$, где w' и w'' — цепочки, длина которых не превышает $2i$ и в каждой из которых столько же нулей, сколько и единиц. Из 3) и предположения индукции следует, что $w \in L_1$. Отсюда следует, что (2.2) верно. □

Пример 21. Пусть L_2 — язык над алфавитом $\{0,\ 1\}$ состоящий из всевозможных слов, которые сворачиваются в ε последовательностью замен подцепочек 01 на ε.

Ясно, что L_2 является подмножеством языка L_1 определенного в примере 20. Также очевидно, что L_2 — собственное подмножество языка L_1, т. к. ни одно из слов 10, 0110 или 011001 не принадлежит L_2. Если в L_2 заменить 0 и 1 соответственно на левую и правую скобки, то можно определить L_2 как множество цепочек составленных из правильно сбалансированных скобок.

Определение языка L_2 может быть воспринято как описание распознавателя. □

Контрольные вопросы

1. Какое множество может являться алфавитом?
2. Сколько языков можно определить над алфавитом $\{0,1\}$?
3. Поясните различие между устройствами для порождения и распознавания языков.
4. Приведите пример определения языка с помощью спецификаторов свойств.

Глава 3
Системы текстовых замен

Определение 46 (Система текстовых замен). *Системой текстовых замен* называется упорядоченная пара $RW = (\Omega, P)$ где
1) Ω — алфавит;
2) P — конечное множество правил

$$P \subset \Omega^* \times \Omega^*,$$

т. е. каждый элемент множества P — пара цепочек (α, β), где α и β — произвольные цепочки над алфавитом Ω.

Пару $(\alpha, \beta) \in P$ называют правилом замены или продукцией и обычно записывают в виде $\alpha \to \beta$.

Определение 47 (Отношение непосредственной выводимости). Пусть $\alpha \to \beta \in P$ — правило в системе текстовых замен $RW = (\Omega, P)$, а $\gamma, \varphi \in \Omega^*$. Тогда говорят, что цепочка $\gamma\beta\varphi$ *непосредственно выводится* из $\gamma\alpha\varphi$ (или цепочка $\gamma\alpha\varphi$ непосредственно порождает цепочку $\gamma\beta\varphi$) по системе текстовых замен RW, и пишут $\gamma\alpha\varphi \Rightarrow_{RW} \gamma\beta\varphi$. Отношение $\Rightarrow_{RW}$ на множестве Ω^* называют отношением *непосредственной выводимости* в системе текстовых замен RW.

Определение 48 (Отношение нетривиальной выводимости). Транзитивное замыкание отношения $\Rightarrow_{RW}$ обозначают $\Rightarrow^+_{RW}$ и называют отношением *нетривиальной выводимости* в системе текстовых замен RW. Если $\alpha, \beta \in \Omega^*$ и $\alpha \Rightarrow^+_{RW} \beta$, то говорят, что цепочка β выводится из α нетривиальным образом.

Определение 49 (Отношение выводимости). Рефлексивное транзитивное замыкание отношения $\Rightarrow_{RW}$ обозначают $\Rightarrow^*_{RW}$ и называют отношением *выводимости* в системе текстовых замен RW. Если $\alpha, \beta \in \Omega^*$ и $\alpha \Rightarrow^*_{RW} \beta$, то говорят, что цепочка β выводится из α.

Определение 50 (Вывод цепочки). Если β выводится из α по системе текстовых замен RW, то последовательную запись

$$\alpha \Rightarrow_{RW} \gamma_1 \Rightarrow_{RW} \gamma_2 \Rightarrow_{RW} \ldots \Rightarrow_{RW} \gamma_k \Rightarrow_{RW} \beta$$

будем называть выводом цепочки β из цепочки α.

Часто, когда известно о какой системе текстовых замен идет речь, имя системы текстовых замен (в нашем случае RW) в вышеперечисленных обозначениях отношений будем опускать.

Пример 22. Рассмотрим систему текстовых замен $RW = (\Omega, P)$, в которой

$$\Omega = \{0, 1, 2, 3\},$$
$$P = \{01 \to 210,\ 03 \to 33,\ 123 \to 021,\ \varepsilon \to 0,\ 221 \to 3\}.$$

В такой системе текстовых замен возможны выводы:

$$123 \Rightarrow 021 \Rightarrow 0201 \Rightarrow 02210 \Rightarrow 030 \Rightarrow 330;$$
$$3 \Rightarrow 03 \Rightarrow 33 \Rightarrow 033 \Rightarrow 333 \Rightarrow 0333 \Rightarrow 3333;$$
$$1232323 \Rightarrow 0212323 \Rightarrow 0202123 \Rightarrow 0202021 \Rightarrow$$
$$\Rightarrow 02020201 \Rightarrow 020202210 \Rightarrow 0202030.$$

Имея в виду вышеприведенные выводы цепочек, можем написать

$$123 \Rightarrow^* 330;$$
$$3 \Rightarrow^* 3333;$$
$$1232323 \Rightarrow^* 0202030. \qquad \square$$

Система текстовых замен может рассматриваться как конструктор языка и как распознаватель. Следующие определения демонстрируют это.

Определение 51 (Язык, порожденный системой текстовых замен). Пусть задана система текстовых замен $RW = (\Omega, P)$ и определено некоторое множество $AX \subset \Omega^*$. Множество слов

$$L_g(RW, AX) = \{\beta \mid\ \alpha \Rightarrow^* \beta,\ \alpha \in AX\}$$

называют языком, порожденным системой текстовых замен RW с помощью множества аксиом AX.

Определение 52 (Язык, допускаемый системой текстовых замен). Пусть задана система текстовых замен $RW = (\Omega, P)$ и определено некоторое множество $AX \subset \Omega^*$. Множество слов

$$L_a(RW, AX) = \{\alpha \mid\ \alpha \Rightarrow^* \beta,\ \beta \in AX\}$$

называют языком, допускаемым системой текстовых замен RW с помощью множества аксиом AX.

Пример 23. Рассмотрим язык из примера 21. Этот язык может распознаваться системой текстовых замен $RW = (\{0,1\}, \{01 \to \varepsilon\})$ и может быть представлен в виде $L_a(RW, \{\varepsilon\})$. $\square$

Пример 24. Рассмотрим язык из примера 19. Этот язык может быть определен с помощью системы текстовых замен

$$RW = (\{0,1,S\}, \{S \to \varepsilon, S \to 0S1\})$$

и может быть представлен в виде

$$L_g(RW, \{S\}) \cap \{0,1\}^*. \tag{3.1}$$

В данном случае система тестовых замен рассматривается как порождающее устройство. Пересечение в (3.1) означает, что из всех порождаемых системой текстовых замен цепочек рассматриваются только те, что составлены из символов алфавита $\{0,1\}$ (отсекаются слова содержащие дополнительный символ S). $\square$

Пример 25. Рассмотрим язык из примера 20. Этот язык может быть определен с помощью системы текстовых замен

$$RW = (\{0,1,S\}, \{S \to \varepsilon, S \to 0S1, S \to 1S0, S \to SS\})$$

и может быть представлен в виде

$$L_g(RW, \{S\}) \cap \{0,1\}^*. \tag{3.2}$$

В данном случае система тестовых замен рассматривается как порождающее устройство. Так же как и в предыдущем примере пересечение в (3.2) означает, что из всех порождаемых системой текстовых замен цепочек рассматриваются только те, в которых отсутствует символ S. $\square$

Контрольные вопросы

1. На каком множестве задается отношение выводимости в системе текстовых замен?
2. Какими свойствами обладает отношение выводимости в системе текстовых замен?
3. Объясните как система текстовых замен может порождать язык.
4. Дайте определение формального языка допускаемого системой текстовых замен?

Глава 4
Грамматики

4.1. Формальное определение грамматики

Определение 53 (Порождающая грамматика). *Порождающей грамматикой* называют четверку $G = (N, \Sigma, P, S)$, где

1) N — алфавит нетерминальных символов;
2) Σ — алфавит терминальных символов;
3) P — конечное множество правил

$$P \subset (N + \Sigma)^* N (N + \Sigma)^* \times (N + \Sigma)^*,$$

т. е. каждый элемент множества P — пара цепочек (α, β), где α — произвольная цепочка из терминальных и нетерминальных символов, содержащая в себе хотя бы один нетерминальный символ, а β — произвольная цепочка из терминальных и нетерминальных символов.

Пару (α, β) обычно принято записывать в виде $\alpha \to \beta$.

4) S — нетерминальный символ из N, начальный символ грамматики.

В последнем определении терминалы (элементы множества терминальных символов) представляют собой символы, из которых строятся цепочки языка. В свою очередь нетерминалы (элементы множества нетерминальных символов) играют роль переменных в грамматике. Начальный символ грамматики задает язык, определяемый грамматикой. Остальные нетерминалы играют роль вспомогательных переменных.

Порождающая грамматика соответствует системе текстовых замен $RW = (N \cup \Sigma, P)$. Согласно этому соответствию можно определить отношения непосредственной выводимости и выводимости по грамматике как соответствующие отношения по системе текстовых замен. Тогда язык, определенный порождающей грамматикой G задается как

$$L(G) = L_g(RW, \{S\}) \cap \Sigma^*.$$

Но можно провести независимое определение языка порожденного грамматикой.

Определение 54 (Язык, порожденный грамматикой). Языком $L(G)$, порожденным (определенным) грамматикой $G = (N, \Sigma, P, S)$, называют множество всех терминальных цепочек (цепочек, состоящих только из терминальных символов), выводимых из начального символа грамматики:

$$L(G) = \{\gamma \in \Sigma^* \mid S \Rightarrow_G^* \gamma\}.$$

Определение 55 (Эквивалентные грамматики). Две грамматики называются *эквивалентными*, если они определяют один и тот же язык.

Пример 26. Рассмотрим грамматику $G = (N, \Sigma, P, S)$, где $N = \{S\}$, $\Sigma = \{0, 1\}$, $P = \{S \to 0S1,\ S \to 01\}$.

В этой грамматике всего лишь два правила и один нетерминальный символ.

Цепочка 01 выводится из S, а следовательно эта цепочка принадлежит языку $L(G)$.

Имеет место $S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000111$. Следовательно $S \Rightarrow^* 000111$, а значит 000111 также принадлежит языку $L(G)$.

Можно показать что язык $L(G)$ состоит из всех непустых цепочек, где последовательность единиц стоит за последовательностью из такого же количества нулей.

$$L(G) = \{0^k 1^k \mid k > 0\}$$

□

Пример 27. Рассмотрим грамматику $G = (N, \Sigma, P, A)$, где $N = \{A, B, C\}$, $\Sigma = \{a, b, c\}$, множество P состоит из правил:

$$\begin{array}{lll} A \to aABC, & A \to abC, & CB \to BC, \\ bB \to bb, & bC \to bc, & cC \to cc. \end{array}$$

Язык $L(G)$ содержит цепочки вида $a^n b^n c^n$ для каждого $n \geqslant 1$, т. к. мы можем использовать первое правило $n - 1$ раз, чтобы получить $A \Rightarrow^* a^{n-1}A(BC)^{n-1}$. Затем мы используем второе правило, чтобы получить $A \Rightarrow^* a^n bC(BC)^{n-1}$. Третье правило дает нам возможность расположить B и C так, чтобы все B предшествовали всем C. Таким образом $A \Rightarrow^* a^n bB^{n-1}C^n$. Затем, используя четвертое правило $n - 1$ раз, получаем $A \Rightarrow^* a^n b^n C^n$. Применим один раз пятое правило, получая $A \Rightarrow^* a^n b^n cC^{n-1}$. Наконец, применив $n - 1$ раз шестое правило, окончательно получаем $A \Rightarrow^* a^n b^n c^n$. Легко показать, что в общем случае возможен только такой порядок применения правил, т. е.

$$L(G) = \{a^n b^n c^n \mid n \geqslant 1\}.$$

□

Пример 28. Рассмотрим грамматику $G = (N, \Sigma, P, S)$, где $N = \{S, A, B\}$,

$\Sigma = \{a, b\}$, а P задается правилами

$$\begin{array}{lll} S \to aB, & S \to bA, & \\ A \to a, & A \to aS, & A \to bAA, \\ B \to b, & B \to bS, & B \to aBB. \end{array}$$

Можно показать, что эта грамматика определяет язык состоящий из цепочек, в которых одинаковое количество символов a и b. □

Замечание 3. Часто, для краткости, правила грамматики, имеющие одинаковую левую часть, будем объединять при записи в «одно» правило, у которого левая часть — левая часть исходных правил, а в правой части все правые части исходных правил, разделенные символом «|». Таким образом грамматика из примера 28 запишется следующим образом:

$$\begin{array}{l} S \to aB \mid bA, \\ A \to a \mid aS \mid bAA, \\ B \to b \mid bS \mid aBB. \end{array}$$

Пример 29. Рассмотрим грамматику $G = (N, \Sigma, P, S)$, где $N = \{S, A, B, C, D\}$, $\Sigma = \{a, b\}$, а P задается правилами

$$\begin{array}{llll} S \to CD, & C \to aCA, & C \to bCB, & C \to \varepsilon, \\ D \to \varepsilon, & AD \to aD, & Aa \to aA, & Ab \to bA, \\ BD \to bD, & Ba \to aB, & Bb \to bB. & \end{array}$$

Можно показать, что эта грамматика определяет язык

$$L(G) = \{ww \mid w \in \{a, b\}^*\}$$ □

Пример 30. Рассмотрим грамматику $G = (N, \Sigma, P, S)$, где $N = \{S\}$, $\Sigma = \{a, b\}$, $P = \{S \to aS, S \to bS, S \to a, S \to b\}$. Эта грамматика определяет язык, состоящий из всех непустых цепочек над алфавитом $\{a, b\}$, т. е. $L(G) = \{a, b\}^+$. □

Определение 56 (Аналитическая грамматика). *Аналитической* (распознающей) грамматикой называют четверку $G = (N, \Sigma, P, S)$, где

1) N — алфавит нетерминальных символов;
2) Σ — алфавит терминальных символов;
3) P — конечное множество продукций

$$P \subset (N + \Sigma)^* \times (N + \Sigma)^* N (N + \Sigma)^*;$$

4) S — нетерминальный символ из N, заключительный (целевой, финаль-

ный, завершающий) символ грамматики.

Аналитическая грамматика G соответствует системе текстовых замен $RW = (N \cup \Sigma, P)$ с языком

$$L(G) = L_a(RW, \{S\}) \cap \Sigma^*.$$

Можно определить язык, допускаемый аналитической грамматикой следующим образом.

Определение 57 (Язык, допускаемый грамматикой). Языком $L(G)$, определенным (допускаемым) аналитической грамматикой $G = (N, \Sigma, P, S)$, называют множество всех терминальных цепочек (цепочек, состоящих только из терминальных символов), из которых можно вывести заключительный символ грамматики:

$$L(G) = \{w \in \Sigma^* \mid w \Rightarrow^* S\}.$$

Определение 58 (Двойственная грамматика). Пусть $G = (N, \Sigma, P, S)$ — порождающая (аналитическая) грамматика. Пусть P' состоит из всех правил вида $\alpha \to \beta$, таких что $\beta \to \alpha \in P$. Тогда аналитическая (порождающая) грамматика $G_1 = (N, \Sigma, P', S)$ называется двойственной для грамматики G

Теорема 1 (О двойственной грамматике)**.** *Пусть для порождающей (аналитической) грамматики* $G = (N, \Sigma, P, S)$ *построена двойственная аналитическая (порождающая) грамматика* $G_1 = (N, \Sigma, P', S)$. *Тогда верно, что* $L(G) = L(G_1)$.

Доказательство. Пусть G — порождающая грамматика.

Утверждаем, что для любого слова $w \in (N + \Sigma)^*$

$$S \Rightarrow_G^* w \text{ тогда итолько тогда, когда } w \Rightarrow_{G_1}^* S. \tag{4.1}$$

Действительно, можно показать, что

$$S \Rightarrow_G w_1 \Rightarrow_G w_2 \Rightarrow_G \cdots \Rightarrow_G w_k \tag{4.2}$$

тогда и только тогда, когда

$$w_k \Rightarrow_{G_1} \cdots \Rightarrow_{G_1} w_2 \Rightarrow_{G_1} w_1 \Rightarrow_{G_1} S. \tag{4.3}$$

По определению P' это имеет место для $k = 1$. Рассуждая по индукции предположим, что утверждение истинно для некоторого $k \geqslant 1$, т. е. (4.2) имеет место тогда и только тогда, когда выполнено (4.3). Но $w_k \Rightarrow_G w_{k+1}$

тогда и только тогда, когда $w_{k+1} \Rightarrow_{G_1} w_k$. Собирая последнее воедино, получим, что (4.1) верно, а следовательно верно и утверждение теоремы. $\square$

Таким образом, сформулированные для порождающих грамматик утверждения справедливы для аналитических грамматик и наоборот.

В дальнейшем, если особо не оговаривается, под термином «грамматика» будем понимать порождающие грамматики.

4.2. Классификация грамматик по Хомскому

Существует классификация грамматик по четырем типам. Введем классификацию для порождающих грамматик. Двойственные им аналитические грамматики будут иметь тип исходных порождающих грамматик.

Определение 53 описывает *грамматику общего вида*, или, другими словами, *грамматику типа* 0.

Определение 59 (Грамматика типа 1). Грамматику $G = (N, \Sigma, P, S)$ называют *контекстно-зависимой грамматикой* (КЗ-грамматикой) или *грамматикой типа* 1, если любое правило в ней имеет вид

$$\gamma_1 A \gamma_2 \to \gamma_1 \alpha \gamma_2,$$

где $\gamma_1, \gamma_2 \in (N + \Sigma)^*$, $A \in N$, $\alpha \in (N + \Sigma)^+$. Также возможно присутствие правила

$$S \to \varepsilon$$

при условии, что S не встречается в правых частях остальных правил.

Определение 60 (Грамматика типа 2). Грамматику $G = (N, \Sigma, P, S)$ называют *контекстно-свободной грамматикой* (КС-грамматикой) или *грамматикой типа* 2, если любое ее правило имеет вид $A \to \alpha$, $A \in N$, $\alpha \in (N + \Sigma)^*$.

То есть грамматика является грамматикой типа 2, если в любом ее правиле в левой части точно один символ — нетерминальный.

КС-грамматики приведены в примерах 26, 28 и 30.

Определение 61 (Грамматика типа 3). Грамматику $G = (N, \Sigma, P, S)$ называют *леволинейной грамматикой* или *грамматикой типа* 3, если любое ее правило имеет вид

$$A \to B\alpha,$$

или

$$A \to \alpha,$$

где $A, B \in N$, $\alpha \in \Sigma^*$.

Наряду с приведенной классификацией существует несколько определений дополнительных классов грамматик. С некоторыми из них мы познакомимся в рамках данного курса. Один из таких классов — неукорачивающие грамматики.

Определение 62 (Неукорачивающие грамматики). Порождающую грамматику $G = (N, \Sigma, P, S)$ называют *неукорачивающей грамматикой*, если для любого правила $\alpha \to \beta \in P$ выполняется $|\alpha| \leqslant |\beta|$. Кроме того, в неукорачивающей грамматике возможно присутствие правила

$$S \to \varepsilon$$

при условии, что S не встречается в правых частях остальных правил.

Грамматика из примера 27 является неукорачивающей.

Язык, определенный грамматикой, называют языком того же типа, что и грамматика. Например, язык, определенный грамматикой типа 3, называют языком типа 3, или леволинейным языком, а язык, определенный грамматикой типа 2, называют языком типа 2, или контекстно-свободным языком (КС-языком).

Будем обозначать $\mathscr{L}_i$ — класс языков типа i.

4.3. Свойства языков и грамматик

Теорема 2 (О существовании эквивалентной грамматики с ограничением на наличие терминальных символов в правилах)**.** *Для любой грамматики $G = (N, \Sigma, P, S)$ можно построить эквивалентную грамматику $G' = (N', \Sigma, P', S)$ такую, что каждое правило в P', содержащее символы из Σ имеет вид $A \to a$, где $A \in N'$, $a \in \Sigma$. Более того, если G — грамматика типа 0, 1, 2 или неукорачивающая, то G' также — грамматика типа 0, 1, 2 или неукорачивающая соответственно.*

Доказательство. Для каждого $a \in \Sigma$ создадим новый нетерминал A_a и сформируем новое множество нетерминалов

$$N' = N \cup \{A_a \mid a \in \Sigma\}.$$

Множество P' получается из P заменой во всех правилах терминалов a на соответствующие A_a, а также добавлением всевозможных правил

$$A_a \to a, \quad a \in \Sigma. \tag{4.4}$$

Покажем, что для полученной грамматики $G' = (N', \Sigma, P', S)$

$$L(G) \subseteq L(G'). \tag{4.5}$$

Пусть непустое слово $w = w_1 w_2 \ldots w_n \in L(G)$. Тогда можем вывести по грамматике G' сначала слово $A_{w_1} A_{w_2} \ldots A_{w_n}$, а затем, используя (4.4) слово w. Если $\varepsilon \in L(G)$, то вывод этого слова можно произвести в G'. Значит (4.5) верно.

Покажем, что

$$L(G') \subseteq L(G). \tag{4.6}$$

Определим гомоморфизм h из $(N' + \Sigma)^*$ в $(N + \Sigma)^*$

$$h(A_a) = a, \quad a \in \Sigma,$$
$$h(X) = X, \quad X \in N \cup \Sigma.$$

Пусть $\alpha \Rightarrow_{G'} \beta$. Если такой вывод был произведен с помощью правила $A_a \to a$, то $h(\alpha) = h(\beta)$. Иначе, $h(\alpha) \Rightarrow_G h(\beta)$. В любом случае

$$h(\alpha) \Rightarrow_G^* h(\beta).$$

Отсюда, если $w \in L(G')$, то

$$S = h(S) \Rightarrow_G^* h(w) = w,$$

откуда $w \in L(G)$, что доказывает (4.6).

Преобразования, проведенные для получения грамматики G' не выводят грамматику из того класса, к которому относилась грамматика G. То есть G' — грамматика типа 0, 1, 2 или неукорачивающая, если грамматика G является таковой.

Таким образом, утверждение теоремы доказано. $\square$

Пример 31. Рассмотрим грамматику $G = (N, \Sigma, P, A)$, где $N = \{A, B, C\}$, $\Sigma = \{a, b, c\}$, множество P состоит из правил:

$$A \to aABC, \quad A \to abc, \quad CB \to BC,$$
$$bB \to bb, \quad bC \to bc, \quad cC \to cc.$$

Построим новую грамматику так, как это определено в доказательстве теоремы 2.

Для каждого терминального символа введем соответствующий ему новый нетерминальный символ с дополнительным новым правилом.

$$D \to a, \quad E \to b, \quad F \to c. \tag{4.7}$$

Подставим новые символы в исходные правила вместо соответствующих терминальных. Получим

$$\begin{gathered} A \to DABC, \quad A \to DEF, \quad CB \to BC, \\ EB \to EE, \quad EC \to EF, \quad FC \to FF. \end{gathered} \tag{4.8}$$

Получим новую грамматику $G' = (N', \Sigma, P', A)$, где

$$N' = \{A, B, C, D, E, F\},$$

P' состоит из всех правил перечисленных в (4.7) и (4.8). □

Теорема 3 (О неукорачиваемости контекстно-зависимых языков)**.** *Классы языков $\mathscr{L}_1$ и неукорачивающих эквивалентны.*

Доказательство. Так как по определению любая грамматика типа 1 является неукорачивающей, для доказательства теоремы достаточно показать, что для любой неукорачивающей грамматики найдется эквивалентная грамматика типа 1.

Пусть дана произвольная неукорачивающая грамматика $G = (N, \Sigma, P, S)$. По теореме 2 можем предположить, что терминальные символы не встречаются в левых частях правил грамматики. Пусть $\#P = n$

Построим новую грамматику G' с множеством правил P', определенным следующим образом.

1. Если $S \to \varepsilon \in P$, то $S \to \varepsilon \in P'$.
2. Если $\gamma_1 A \gamma_2 \to \gamma_1 \alpha \gamma_2 \in P$, где $\gamma_1, \gamma_2 \in (N + \Sigma)^*$, $A \in N$, $\alpha \in (N + \Sigma)^+$, то $\gamma_1 A \gamma_2 \to \gamma_1 \alpha \gamma_2 \in P'$.
3. Если i-е правило в P не подходит под определение правила в пункте 2 и имеет вид

$$X_1 X_2 \dots X_{k-1} X_k \to Y_1 Y_2 \dots Y_{l-1} Y_l,$$

то в P' включаем набор правил

$$\begin{gathered} X_1 X_2 X_3 \dots X_{k-1} X_k \to Z_{i1} X_2 X_3 \dots X_{k-1} X_k, \\ Z_{i1} X_2 X_3 \dots X_{k-1} X_k \to Z_{i1} Z_{i2} X_3 \dots X_{k-1} X_k, \\ Z_{i1} Z_{i2} X_3 \dots X_{k-1} X_k \to Z_{i1} Z_{i2} Z_{i3} \dots X_{k-1} X_k, \\ \dots \\ Z_{i1} Z_{i2} Z_{i3} \dots Z_{i\,k-1} X_k \to Z_{i1} Z_{i2} Z_{i3} \dots Z_{i\,k-1} Z_{ik}, \end{gathered}$$

$$
\begin{gathered}
Z_{i1}Z_{i2}Z_{i3}\dots Z_{i\,k-1}Z_{ik} \to Y_1Z_{i2}Z_{i3}\dots Z_{i\,k-1}Z_{ik},\\
Y_1Z_{i2}Z_{i3}\dots Z_{i\,k-1}Z_{ik} \to Y_1Y_2Z_{i3}\dots Z_{i\,k-1}Z_{ik},\\
Y_1Y_2Z_{i3}\dots Z_{i\,k-1}Z_{ik} \to Y_1Y_2Y_3\dots Z_{i\,k-1}Z_{ik},\\
\dots\\
Y_1Y_2Y_3\dots Y_{k-1}Z_{ik} \to Y_1Y_2Y_3\dots Y_{k-1}Y_k\dots Y_l,
\end{gathered}
$$

где Z_{ij} — новые нетерминалы.

Пусть N_Z — множество, составленное из всех символов Z_{ij}, введенных на шаге 3.

Тогда грамматика $G' = (N', \Sigma, P', S)$, где $N' = N' \cup N_Z$, будет определять тот же язык, что и исходная неукорачивающая грамматика. Более того, по определению, получившаяся грамматика является контекстно-зависимой. □

Пример 32. Рассмотрим грамматику, полученную в предыдущем примере. Грамматика $G' = (N', \Sigma, P', A)$, где $N' = \{A, B, C, D, E, F\}$, $\Sigma = \{a, b, c\}$, P' состоит из правил:

$$
\begin{array}{lll}
A \to DABC, & A \to DEF, & CB \to BC,\\
EB \to EE, & EC \to EF, & FC \to FF,\\
D \to a, & E \to b, & F \to c.
\end{array}
$$

Данная грамматика является неукорачивающей, но не является грамматикой типа 1 из-за наличия в ней правила

$$CB \to BC.$$

В соответствии с алгоритмом преобразования, предложенным в доказательстве последней теоремы это правило в новой грамматике будет заменено на правила

$$
\begin{aligned}
CB &\to Z_{31}B,\\
Z_{31}B &\to Z_{31}Z_{32},\\
Z_{31}Z_{32} &\to BZ_{32},\\
BZ_{32} &\to BC.
\end{aligned}
$$

Так как все остальные правила удовлетворяют условиям, накладываемым на контекстно-зависимые грамматики, то их можно оставить без изменения. Новая грамматика $G'' = (N' \cup \{Z_{31}, Z_{32}\}, \Sigma, P'', A)$, где P''

состоит из правил

$$\begin{array}{cccc} A \to DABC, & A \to DEF, & CB \to Z_{31}B, & Z_{31}B \to Z_{31}Z_{32}, \\ Z_{31}Z_{32} \to BZ_{32}, & BZ_{32} \to BC, & EB \to EE, & EC \to EF, \\ FC \to FF, & D \to a, & E \to b, & F \to c, \end{array}$$

является контекстно-зависимой грамматикой и определяет тот же язык, что и грамматика G'. □

Теорема 4 (Классификация конечных языков). *Любой из классов языков $\mathscr{L}_i$, $0 \leqslant i \leqslant 3$, содержит все конечные языки.*

Доказательство. Пусть $w_1, w_2, \ldots, w_n$ набор слов над алфавитом Σ. Тогда грамматика

$$G_1 = (\{S\}, \Sigma, \{S \to w_1, S \to w_2, \ldots, S \to w_n\}, S)$$

порождает язык $L(G_1) = \{w_1, w_2, \ldots, w_n\}$.

Грамматика

$$G_2 = (\{S\}, \Sigma, \{S \to S\}, S)$$

порождает пустой язык $L(G_2) = \varnothing$.

Как грамматика G_1, так и грамматика G_2 подходит под определение грамматики любого типа. Следовательно теорема доказана. □

Пример 33. Пусть дан конечный язык

$$L = \{\varepsilon, a, ab, abaa\}.$$

Пользуясь методом, представленным в доказательстве теоремы 4, для него можно определить грамматику

$$G = (N, \Sigma, P, S),$$

где $N = \{S\}$, $\Sigma = \{a, b\}$ и множество P представлено правилами

$$S \to \varepsilon \mid a \mid ab \mid abaa.$$

Полученная грамматика удовлетворяет ограничениям, накладываемым на любой из типов грамматик, а значит данный язык можно отнести классу языков любого типа. □

Теорема 5 (О замкнутости класса $\mathscr{L}_3$ относительно регулярных операций). *Класс леволинейных языков замкнут относительно регулярных операций.*

Доказательство. Пусть L и L' — леволинейные языки, определенные с помощью грамматик

$$G = (N, \Sigma, P, S), \quad G' = (N', \Sigma', P', S'),$$

соответственно.

Пусть $N \cap N' = \varnothing$ (в противном случае можем переименовать нетерминалы одного из множеств).

I. Язык $L \cup L'$ определяется грамматикой

$$G_1 = (N + N' + \{S_0\}, \Sigma + \Sigma', P \cup P' \cup \{S_0 \to S, S_0 \to S'\}, S_0).$$

Действительно, любое слово языка L выводится из S. Следовательно его можно вывести и из S_0. Аналогично, любое слово языка L' можно вывести из S_0. То есть любое слово объединения $L \cup L'$ можно вывести из S_0. С другой стороны, так как в грамматиках G и G' нет одинаковых нетерминалов, при выводе слова из S по грамматике G_1 невозможно получить цепочку, содержащую нетерминальные символы из N', т. е. при таком выводе нельзя применить ни одного правила из P'. Аналогично, при выводе цепочки из нетерминала S' по грамматике G_1 невозможно применить ни одного правила из P. Следовательно из нетерминала S_0 можно вывести только слова из $L(G_1) \cup L(G_2)$.

II. Заменим в P' каждое правило вида

$$A \to \alpha, \quad A \in N', \alpha \in \Sigma'^* \tag{4.9}$$

на правило $A \to S\alpha$. Обозначим полученное множество правил через P_1. Тогда грамматика типа 3

$$G_2 = (N + N', \Sigma + \Sigma', P \cup P_1, S')$$

порождает язык LL'.

В самом деле, если $\beta \in L'$, то существует вывод $S' \Rightarrow_{G'} \beta$. Вследствие произведенной замены правил вида (4.9), имеет место вывод по грамматике G_2 $S' \Rightarrow_{G_2} Sw$. Значит, для любого слова $\alpha \in L$ выполняется $S' \Rightarrow^*_{G_2} S\beta \Rightarrow^*_{G_2} \alpha\beta$. Так как слово β взято произвольным, все слова конкатенации LL' можно вывести по грамматике G_2 из S'. Более того, легко убедиться, что никакие другие слова не выводятся по грамматике G_2 из S'.

III. Заменим в P' каждое правило вида (4.9) на правило вида

$$A \to S'\alpha$$

и обозначим полученное множество правил через P_2. Тогда грамматика типа 3

$$G_3 = (N' + \{S_0\}, \Sigma', P' \cup P_2 \cup \{S_0 \to \varepsilon, S_0 \to S'\}, S_0)$$

порождает язык L'^*.

По грамматике G_3 возможны выводы

$$\begin{aligned}
&S_0 \Rightarrow_{G_3} \varepsilon;\\
&S_0 \Rightarrow_{G_3} S';\\
&S' \Rightarrow^*_{G_3} \alpha, \quad \forall\alpha \in L';\\
&S' \Rightarrow^*_{G_3} S'\alpha, \quad \forall\alpha \in L'.
\end{aligned}$$

Собирая воедино, получаем, что из S_0 можно вывести любое слово, составленное из слов языка L', т. е. $L(G_3) = L'^*$. $\square$

Пример 34. Рассмотрим две грамматики.

$G_1 = (\{S, A, B\}, \{0, 1, 2\}, P_1, S)$, с множеством продукций P_1:

$$\begin{aligned}
&S \to A11 \mid B0,\\
&A \to S01 \mid 1,\\
&B \to A0 \mid B2.
\end{aligned}$$

$G_2 = (\{E, C, D\}, \{0, 1, 2\}, P_2, E)$, с множеством продукций P_2:

$$\begin{aligned}
&E \to C0 \mid \varepsilon,\\
&C \to C1 \mid D2,\\
&D \to E0 \mid 2.
\end{aligned}$$

В соответствии с доказательством последней теоремы объединение языков $L(G_1) \cup L(G_2)$ будет определять грамматика
$G_3 = (\{S_0, S, A, B, C, D, E\}, \{0, 1, 2\}, P_3, S_0)$, в которой P_3 представлено продукциями:

$$\begin{aligned}
&S_0 \to S \mid E,\\
&S \to A11 \mid B0, \qquad && E \to C0 \mid \varepsilon,\\
&A \to S01 \mid 1, && C \to C1 \mid D2,\\
&B \to A0 \mid B2, && D \to E0 \mid 2.
\end{aligned}$$

Для конкатенации языков $L(G_1)L(G_2)$ преобразуем правила грамматики G_2, не содержащие нетерминальных символов в правой части

$$E \to \varepsilon, \qquad D \to 2 \tag{4.10}$$

в правила
$$E \to S, \qquad D \to S2 \tag{4.11}$$
Объединив правила грамматики G_1 с модифицированным множеством правил грамматики G_2, получим грамматику $G_4 = (\{S, A, B, C, D, E\}, \{0, 1, 2\}, P_4, E)$, где P_4 — множество правил:
$$\begin{array}{ll} S \to A11 \mid B0, & E \to C0 \mid S, \\ A \to S01 \mid 1, & C \to C1 \mid D2, \\ B \to A0 \mid B2, & D \to E0 \mid S2. \end{array}$$

Преобразуя правила (4.10) так как сказано в последней части доказательства теоремы 5 получим новые правила
$$E \to E, \qquad D \to E2.$$

Внесём новый начальный нетерминальный символ и добавим два правила для него, после чего объединяем новые правила с правилами грамматики G_2. Получим множество правил P_5:
$$\begin{array}{l} S_0 \to \varepsilon \mid E, \\ E \to C0 \mid E \mid \varepsilon, \\ C \to C1 \mid D2, \\ D \to E0 \mid E2 \mid 2. \end{array}$$

Грамматика $G_5 = (\{S_0, E, C, D\}, \{0, 1, 2\}, P_5, S_0)$ определяет итерацию $L(G_2)^*$

Аналогичным образом можно найти грамматику для итерации $L(G_1)^*$: $G_6 = (\{S_0, S, A, B\}, \{0, 1, 2\}, P_6, S_0)$, с множеством продукций P_6:
$$\begin{array}{l} S_0 \to \varepsilon \mid S, \\ S \to A11 \mid B0, \\ A \to S01 \mid S1 \mid 1, \\ B \to A0 \mid B2. \end{array}$$
□

Теорема 6 (О замкнутости класса $\mathscr{L}_2$ относительно регулярных операций)**.** *Класс контекстно-свободных языков замкнут относительно регулярных операций.*

Доказательство. Пусть L и L' определены с помощью контекстно-свободных грамматик
$$G = (N, \Sigma, P, S), \quad G' = (N', \Sigma', P', S'),$$

соответственно.

Пусть $N \cap N' = \varnothing$ (в противном случае можем переименовать нетерминалы одного из множеств).

Тогда язык $L \cup L'$ определяется грамматикой

$$(N + N' + \{S_0\}, \Sigma + \Sigma', P \cup P' \cup \{S_0 \to S, S_0 \to S'\}, S_0).$$

Язык LL' определяется грамматикой типа 2

$$(N + N' + \{S_0\}, \Sigma + \Sigma', P \cup P' \cup \{S_0 \to SS'\}, S_0).$$

Язык L^* определяется грамматикой типа 2

$$(N + \{S_0\}, \Sigma, P \cup \{S_0 \to \varepsilon, S_0 \to S_0S\}, S_0).$$

$\square$

Методы построения грамматик для конкатенации и итерации языков типа 2 несколько проще, чем подобные методы для языков типа 3. Такое упрощение связано с тем, что грамматики типа 3 — это грамматики типа 2 с дополнительными ограничениями, и такие ограничения не выполняются при построении грамматик в доказательстве последней теоремы. Действительно, рассмотрим грамматики G_1 и G_2 из примера 34 и построим сначала грамматику для конкатенации $L(G_1)L(G_2)$ методом, предложенным в доказательстве теоремы 6. Для этого объединяем все правила исходных грамматик и добавляем правило для нового начального нетерминала. Получим грамматику со следующим набором правил:

$$\begin{array}{ll}
S_0 \to SE, & \\
S \to A11 \mid B0, & E \to C0 \mid \varepsilon, \\
A \to S01 \mid 1, & C \to C1 \mid D2, \\
B \to A0 \mid B2, & D \to E0 \mid 2,
\end{array}$$

где S_0 — новый начальный нетерминал. Так как исходные грамматики можно было отнести ко 2-му типу, полученная грамматика определяет конкатенацию определенных ими языков. Но из-за первого правила построенную грамматику нельзя отнести к типу 3. Поэтому такое построение не доказывает факта, что конкатенация языков типа 3 есть язык типа 3.

Аналогичная ситуация имеет место при применении метода построения грамматики для итерации языка, предложенного в доказательстве теоремы 6 к грамматике типа 3. Построим этим методом грамматику для

$(L(G_1))^*$. Получим грамматику с набором правил:

$$\begin{aligned}
&S_0 \to \varepsilon \mid S_0S,\\
&S \to A11 \mid B0,\\
&A \to S01 \mid 1,\\
&B \to A0 \mid B2,
\end{aligned}$$

где S_0 — новый начальный нетерминал. Как и в предыдущем случае из за правила $S_0 \to S_0S$ данную грамматику нельзя отнести к типу 3.

Теорема 7 (О замкнутости класса $\mathscr{L}_1$ относительно регулярных операций)**.** *Класс контекстно-зависимых языков замкнут относительно регулярных операций.*

Доказательство. Пусть L и L' определены с помощью контекстно-зависимых грамматик

$$G = (N, \Sigma, P, S), \quad G' = (N', \Sigma', P', S'),$$

соответственно.

По теореме 2 можем предположить, что терминальные символы не встречаются в левых частях правил грамматик. Кроме того, пусть $N \cap N' = \varnothing$ (в противном случае можем переименовать нетерминалы одного из множеств).

I. Если $\varepsilon \notin L \cup L'$, то язык $L \cup L'$ определяется грамматикой

$$(N + N' + \{S_0\}, \Sigma + \Sigma', P \cup P' \cup \{S_0 \to S, S_0 \to S'\}, S_0). \qquad (4.12)$$

Если $\varepsilon \in L \cup L'$, рассмотрим языки $L - \{\varepsilon\}$ и $L' - \{\varepsilon\}$. Для них определим грамматику по (4.12), после чего добавляем дополнительное правило $S_0 \to \varepsilon$. Это не выведет получившуюся грамматику из класса языков $\mathscr{L}_1$, так как S_0 — новый нетерминал, который не встречается в правых частях правил. Полученная грамматика определяет язык $L \cup L'$.

II. Предположим, что $\varepsilon \notin L \cup L'$. Тогда грамматика типа 1

$$(N + N' + \{S_0\}, \Sigma + \Sigma', P \cup P' \cup \{S_0 \to SS'\}, S_0) \qquad (4.13)$$

порождает язык LL'.

Действительно, очевидно, что каждое слово в LL' может быть выведено по (4.13). С другой стороны, если

$$S_0 \Rightarrow \gamma_1 \Rightarrow \gamma_2 \cdots \Rightarrow \gamma_u$$

вывод по (4.13), тогда любая из цепочек γ_j может быть представлена в форме $\gamma_j = \vartheta_j \vartheta'_j$, где $S \Rightarrow^*_G \vartheta_j$ и $S' \Rightarrow^*_{G'} \vartheta'_j$. Это можно доказать по индукции. Ясно, что $\gamma_1 = SS'$ находится в такой форме. Если же γ_k в данной форме, то γ_{k+1} также в данной форме, т. к. левые части правил грамматик не содержат терминальных символов, а множества нетерминальных символов не пересекаются. Значит, только слова из LL' порождаются грамматикой (4.13).

Пусть $\varepsilon \in L \cup L'$. Обозначим $L_1 = L - \{\varepsilon\}$, $L'_1 = L' - \{\varepsilon\}$.

Очевидно, что LL' это один из следующих языков

$$L_1 L'_1 \cup L_1, \quad L_1 L'_1 \cup L'_1, \quad L_1 L'_1 \cup L_1 \cup L'_1 \cup \{\varepsilon\}.$$

Так как класс $\mathscr{L}_1$ замкнут относительно объединения, а в конкатенации участвуют языки без пустого слова, то весь класс $\mathscr{L}_1$ замкнут относительно конкатенации.

III. В связи с тем, что $(L \cup \{\varepsilon\})^* = L^*$, можем не теряя общности предположить, что L не содержит ε.

Действительно, если $\varepsilon \in L(G)$, мы всегда можем построить грамматику G_1 такую, что $L(G_1) = L(G) - \{\varepsilon\}$ (просто удаляем правило $S \to \varepsilon$).

В таком случае, грамматика

$$\begin{aligned}(N + \{S_0, S_1\}, \Sigma, P \cup &\{S_0 \to \varepsilon, S_0 \to S, S_0 \to S_1 S\} \\ &\cup \{S_1 a \to S_1 S a \mid a \in \Sigma\} \\ &\cup \{S_1 a \to S a \mid a \in \Sigma\}, S_0)\end{aligned} \tag{4.14}$$

порождает язык L^*. Действительно, легко видеть, что любое слово в L^* порождается (4.14). С другой стороны, рассмотрим произвольный вывод

$$S_0 \Rightarrow \gamma_1 \Rightarrow \gamma_2 \cdots \Rightarrow \gamma_u, \quad u \geqslant 1,$$

согласно (4.14). Если $\gamma_1 = \varepsilon$ или $\gamma_1 = S$, то в результате вывода можно получить только слова из L и, следовательно, $\gamma_u \in L^*$. В противном случае, $\gamma_1 = S_1 S$ и каждая цепочка γ_j находится в одной из следующих форм

1. $S_1 \varphi_1 \ldots \varphi_v$, $v \geqslant 1$, где первый символ в цепочках $\varphi_2, \ldots, \varphi_v$ терминальный и $S \Rightarrow^*_G \varphi_\nu$, $\nu = 1, \ldots, v$;
2. $\varphi_1 \ldots \varphi_v$, $v \geqslant 1$, где первый символ в цепочках $\varphi_2, \ldots, \varphi_v$ терминальный и $S \Rightarrow^*_G \varphi_\nu$, $\nu = 0, \ldots, v$.

Это можно показать по индукции. Пусть γ_j в форме 1 или 2. Из вида продукций в (4.14) и из того, что терминалы не встречаются в правых частях правил следует, что γ_{j+1} также в форме 1 или 2. Отсюда следует, что только цепочки из L^* могут быть выведены по (4.14). $\square$

Пример 35. В доказательстве последней теоремы для построения грамматики для итерации языка типа 1 предлагается более сложный метод, чем был предложен в доказательстве теоремы 6 (для итерации языка типа 2). Связано это с тем, что при применении упомянутого метода к произвольной грамматике типа 1 может быть получена грамматика, определяющая не только слова из итерации языка. Например, рассмотрим грамматику

$$G = (N, \Sigma, P, S),$$

где $N = \{S, A, B\}$, $\Sigma = \{a, b, c\}$ и множество P представлено правилами

$$\begin{array}{lll} S \to ABcB, & & \\ AB \to Aa, & Aa \to aa, & AB \to CB, \\ CB \to CD, & CD \to BD, & BD \to BA, \\ BA \to Bb, & Bb \to bb, & B \to c. \end{array}$$

Данная грамматика — грамматика типа 1. Легко проверить, что она определяет конечный язык

$$L(G) = \{aacc, bbcc\}. \tag{4.15}$$

Если построить грамматику для итерации этого языка методом из доказательства теоремы 6, получим грамматику с правилами

$$\begin{array}{lll} S_0 \to \varepsilon \mid S_0S, & S \to ABcB, & \\ AB \to Aa, & Aa \to aa, & AB \to CB, \\ CB \to CD, & CD \to BD, & BD \to BA, \\ BA \to Bb, & Bb \to bb, & B \to c. \end{array}$$

Любое слово из $(L(G))^*$ определяется этой грамматикой. Действительно, С помощью правил для S_0 можно получить цепочку из любого количества символов S, каждый из которых можно заменить по правилам исходной грамматики на слово языка $L(G)$. Но для построенной грамматики также возможен вывод:

$$\begin{aligned} S_0 \Rightarrow S_0S \Rightarrow S0SS \Rightarrow SS \Rightarrow ABcBS \Rightarrow \\ ABcBABcB \Rightarrow ABcBbBcB \Rightarrow ABcbbBcB \Rightarrow \\ ABcbbccB \Rightarrow ABcbbccc \Rightarrow Aacbbccc \Rightarrow aacbbccc. \end{aligned} \tag{4.16}$$

Если предположить, что построенная грамматика определяет итерацию языка $L(G)$, то последний вывод показывает, что цепочка $aacbbccc$ принадлежит итерации и следовательно должна представлять из себя конкатенацию слов языка $L(G)$, но из (4.15) следует, что это не так.

Применяя подобный метод, предложенный в теореме 7 получим грамматику с правилами:

$$\begin{array}{llll} S_0 \to \varepsilon \mid S \mid S_1S, & S \to ABcB, & & \\ S_1a \to S_1Sa \mid Sa, & AB \to Aa, & Aa \to aa, & AB \to CB, \\ S_1b \to S_1Sb \mid Sb, & CB \to CD, & CD \to BD, & BD \to BA, \\ S_1c \to S_1Sc \mid Sc, & BA \to Bb, & Bb \to bb, & B \to c. \end{array}$$

Легко убедиться, что с помощью правил этой грамматики можно построить только слова из итерации языка $L(G)$, и вывод, подобный 4.16, невозможен. □

Теорема 8 (О замкнутости класса $\mathscr{L}_0$ относительно регулярных операций)**.** *Класс языков общего вида замкнут относительно регулярных операций.*

Доказательство. Пусть L и L' определены с помощью грамматик

$$G = (N, \Sigma, P, S), \quad G' = (N', \Sigma', P', S'),$$

соответственно.

По теореме 2 можем предположить, что терминальные символы не встречаются в левых частях правил грамматик. Кроме того, пусть $N \cap N' = \varnothing$ (в противном случае можем переименовать нетерминалы одного из множеств).

I. Язык $L \cup L'$ определяется грамматикой

$$(N + N' + \{S_0\}, \Sigma + \Sigma', P \cup P' \cup \{S_0 \to S, S_0 \to S'\}, S_0).$$

II. Грамматика типа 0

$$(N + N' + \{S_0\}, \Sigma + \Sigma', P \cup P' \cup \{S_0 \to SS'\}, S_0) \tag{4.17}$$

порождает язык LL'.

Действительно, очевидно, что каждое слово в LL' может быть выведено по (4.17). С другой стороны, если

$$S_0 \Rightarrow \gamma_1 \Rightarrow \gamma_2 \Rightarrow \dots \Rightarrow \gamma_u$$

вывод по (4.17), тогда любая из γ_j может быть представлена в форме $\gamma_j = \vartheta_j\vartheta'_j$, где $S \Rightarrow^*_G \vartheta_j$ и $S' \Rightarrow^*_{G'} \vartheta'_j$. Это можно доказать по индукции. Ясно, что $\gamma_1 = SS'$ находится в такой форме. Если же γ_k в данной форме, то γ_{k+1} также в данной форме, т. к. левые части правил грамматик не содержат терминальных символов. А значит, только слова из LL' порождаются грамматикой (4.17).

III. Так как $(L \cup \{\varepsilon\})^* = L^*$, можем, не теряя общности, предположить, что L не содержит ε.

Действительно, если $\varepsilon \in L(G)$, мы всегда можем построить грамматику G_1 такую, что $L(G_1) = L(G) - \{\varepsilon\}$ (заменим каждое правило $\alpha \to \varepsilon$ на правила $x\alpha \to x$ и $\alpha x \to x$, для всех $x \in (N + \Sigma)$).

В таком случае, грамматика

$$\begin{aligned}(N + \{S_0, S_1\}, \Sigma, P \cup \{&S_0 \to \varepsilon, S_0 \to S, S_0 \to S_1 S\} \\ &\cup \{S_1 a \to S_1 S a \mid a \in \Sigma\} \\ &\cup \{S_1 a \to S a \mid a \in \Sigma\}, S_0)\end{aligned} \tag{4.18}$$

порождает язык L^*. Действительно, легко видеть, что любое слово в L^* порождается (4.18). С другой стороны, рассмотрим произвольный вывод

$$S_0 \Rightarrow \gamma_1 \Rightarrow \gamma_2 \cdots \Rightarrow \gamma_u, \quad u \geqslant 1$$

согласно (4.18). Если $\gamma_1 = \varepsilon$ или $\gamma_1 = S$, то в результате вывода можно получить только слова из L и, следовательно, $\gamma_u \in L^*$. В противном случае, $\gamma_1 = S_1 S$ и каждая цепочка γ_j находится в одной из следующих форм

1) $S_1 \varphi_1 \ldots \varphi_v$, $v \geqslant 1$, где первый символ в цепочках $\varphi_1, \ldots, \varphi_v$ терминальный и $S \Rightarrow_G^* \varphi_\nu$, $\nu = 1, \ldots, v$;
2) $\varphi_1 \ldots \varphi_v$, $v \geqslant 1$, где первый символ в цепочках $\varphi_1, \ldots, \varphi_v$ терминальный и $S \Rightarrow_G^* \varphi_\nu$, $\nu = 0, \ldots, v$.

Это можно показать по индукции. Пусть γ_j в форме 1 или 2. Из вида продукций в (4.18) и из того, что терминалы не встречаются в правых частях правил следует, что γ_{j+1} также в форме 1 или 2. Отсюда следует, что только цепочки из L^* могут быть выведены по (4.18). □

Теорема 9 (О замкнутости классов языков относительно положительной итерации)**.** *Любой из классов языков $\mathscr{L}_i$, $0 \leqslant i \leqslant 3$, замкнут относительно операции положительной итерации.*

Утверждение теоремы напрямую следует из замкнутости классов языков относительно регулярных операций, т. к. $L^+ = LL^*$.

Теорема 10 (О замкнутости классов языков относительно зеркального отражения)**.** *Любой из классов языков $\mathscr{L}_i$, $0 \leqslant i \leqslant 2$, замкнут относительно операции зеркального отражения.*

Доказательство следует из того факта, что если в грамматике типа i $(0 \leqslant i \leqslant 2)$ поменять каждое правило

$$\alpha \to \beta$$

на правило

$$\alpha^R \to \beta^R,$$

то получим грамматику, относящуюся к тому же классу, что и первоначальная, и определяющую зеркальное отражение исходного языка.

Теорема 11 (О замкнутости $\mathscr{L}_0$ и $\mathscr{L}_2$ относительно подстановки)**.** *Классы языков $\mathscr{L}_0$ и $\mathscr{L}_2$ замкнуты относительно подстановки.*

Доказательство. I. Рассмотрим класс $\mathscr{L}_2$.

Пусть $L = L(G)$, где

$$G = (N, \Sigma, P, S), \quad \Sigma = \{a_1, \dots, a_r\}.$$

Пусть σ — подстановка такая, что $\sigma(a_i)$, $1 \leqslant i \leqslant r$, порождается контекстно-свободной грамматикой

$$G_i = (N_i, \Sigma_i, P_i, S_i).$$

Предположим, что множества нетерминалов всех введенных грамматик попарно не пересекаются.

Обозначим через P_0 множество правил, полученное из P заменой a_i на S_i. Тогда грамматика

$$(N \cup N_1 \cup \dots \cup N_r, \Sigma_1 \cup \dots \cup \Sigma_r, P_0 \cup \dots \cup P_r, S)$$

есть грамматика типа 2, порождающая язык $\sigma(L)$.

II. Рассмотрим класс $\mathscr{L}_0$.

Пусть $L = L(G)$, где

$$G = (N, \Sigma, P, S), \quad \Sigma = \{a_1, \dots, a_r\}.$$

Пусть σ — подстановка такая, что $\sigma(a_i)$, $1 \leqslant i \leqslant r$ порождается грамматикой общего вида

$$G_i = (N_i, \Sigma_i, P_i, S_i).$$

Предположим, что множества нетерминалов всех введенных грамматик попарно не пересекаются.

Для каждого терминального символа $a_i \in \Sigma$ введем новый нетерминал A_i. Кроме этого введем два новых нетерминала X_0 и X.

Обозначим через P_0 множество правил, полученное из P заменой всех терминальных символов a_i на новый нетерминал A_i.

Возьмем новый нетерминал X и обозначим

$$P' = \{XA_i \to XS_i \mid 1 \leqslant i \leqslant r\},$$
$$P'' = \{Xa \to aX \mid a \in \Sigma_1 \cup \dots \cup \Sigma_r\}.$$

Тогда грамматика

$$\begin{aligned}&(N \cup N_1 \cup \dots \cup N_r \cup \{X, X_0\},\\ &\Sigma_1 \cup \dots \cup \Sigma_r,\\ &P_0 \cup \dots \cup P_r \cup P' \cup P'' \cup \{X_0 \to XS,\ X \to \varepsilon\}, X_0)\end{aligned}$$

есть грамматика типа 0, порождающая язык $\sigma(L)$. □

Следствие 11.1. *Классы языков $\mathscr{L}_0$ и $\mathscr{L}_2$ замкнуты относительно гомоморфизма.*

Теорема 12 (О замкнутости класса $\mathscr{L}_0$ относительно пересечения)**.** *Класс языков общего вида замкнут относительно пересечения.*

Доказательство. Пусть $L = L(G)$ и $L' = L(G')$, где

$$G = (N, \Sigma, P, S), \quad G' = (N', \Sigma', P', S').$$

грамматики типа 0.

Рассмотрим грамматику

$$G_1 = (N \cup N' \cup N_1, \Sigma \cup \Sigma', P \cup P' \cup P_1, X_0),$$

где

$$N_1 = \{A_a \mid a \in \Sigma \cup \Sigma'\} \ \cup \ \{X_0, X_1, X_2\}$$

— новые нетерминалы, а P_1 состоит из правил

$$X_0 \to X_1 S X_2 S' X_1 \tag{4.19}$$
$$X_2 a \to A_a X_2 \qquad a \in \Sigma \cup \Sigma' \tag{4.20}$$
$$bA_a \to A_a b \qquad a, b \in \Sigma \cup \Sigma' \tag{4.21}$$
$$X_1 A_a a \to a X_1 \qquad a \in \Sigma \cup \Sigma' \tag{4.22}$$
$$X_1 X_2 X_1 \to \varepsilon \tag{4.23}$$

Используя (4.19) и продукции из $P \cup P'$, все цепочки формы

$$X_1 w X_2 w' X_1, \quad w \in L,\ w' \in L' \tag{4.24}$$

могут быть выведены из X_0. По (4.20)-(4.22), цепочка

$$wX_1X_2X_1 \tag{4.25}$$

выводится из (4.24) тогда и только тогда, когда $w = w'$. По (4.23) слово w выводится из (4.25). Так как только таким образом можно получить из X_0 терминальную цепочку, делаем заключение, что

$$L(G_1) = L \cap L'.$$

□

Контрольные вопросы

1. Дайте определение порождающей грамматики.
2. Поясните понятие вывода строки в заданной грамматике.
3. Какие грамматики могут выступать как средства распознавания языка?
4. Поясните понятие распознавания слова с помощью заданной грамматики.
5. Какую роль в грамматике играют терминальные и нетерминальные символы?
6. Охарактеризуйте классы формальных языков в иерархии Хомского.
7. Охарактеризуйте правила контекстно-свободной грамматики.
8. Как классифицируются аналитические грамматики?
9. Что означает утверждение, что две грамматики эквивалентны?
10. Дайте определение двойственных грамматик.
11. Дайте определение неукорачивающей грамматики.
12. Какой класс языков порождается неукорачивающими грамматиками?

Глава 5
Иерархия автоматов

5.1. Конечные автоматы

Определение 63 (Конечный детерминированный автомат). Систему текстовых замен $M = (\Omega, P)$ называют *конечным детерминированным автоматом* (КДА), если

1) в алфавите Ω можно выделить два алфавита Q и Σ, называемые соответственно множеством символов состояний (множеством состояний) и множеством входных сигналов (входным алфавитом), причем $Q \cap \Sigma = \varnothing$, $Q \cup \Sigma = \Omega$;
2) в множестве Q выделяется элемент q_0, называемый начальным (стартовым) состоянием, и подмножество F — множество заключительных состояний;
3) любая продукция в P имеет вид

$$qa \to r, \quad q, r \in Q, a \in \Sigma, \tag{5.1}$$

и более того, для любой пары (q, a), где $q \in Q$ и $a \in \Sigma$ в P присутствует точно одна продукция вида (5.1).

Таким образом, для определения КДА достаточно определить пять компонент (Q, Σ, P, q_0, F). По другому КДА можно определить как пятерку $M = (Q, \Sigma, \delta, q_0, F)$, где Q, Σ, q_0, F обозначают те же объекты, что и в последнем определении, а элемент δ, называемый функцией переходов, можно представить как отображение вида $\delta : Q \times \Sigma \to Q$, что соответствует набору продукций из P.

Отношение выводимости $\Rightarrow^*$ для КДА имеет общий вид отношения выводимости в системах текстовых замен.

Определение 64 (Язык допускаемый КДА). Язык, определенный (допускаемый, распознаваемый) КДА $M = (Q, \Sigma, P, q_0, F)$, есть множество слов

$$L(M) = \{w \in \Sigma^* \mid q_0 w \Rightarrow^* q,\ q \in F\}.$$

Определение 65 (Конечный недетерминированный автомат). Система текстовых замен $M = (\Omega, P)$ называется *конечным недетерминированным автоматом* (НКА), если

1) в алфавите Ω можно выделить два алфавита Q и Σ, называемые соответственно множеством символов состояний (множеством состояний) и множеством входных сигналов (входным алфавитом), причем $Q \cap \Sigma = \varnothing$, $Q \cup \Sigma = \Omega$;
2) в множестве Q выделяется подмножество S, называемое множеством начальных (стартовых) состояний, и подмножество F — множество заключительных состояний;
3) любая продукция в P имеет вид

$$qa \to r, \quad q, r \in Q, a \in \Sigma.$$

По аналогии с КДА можно определить НКА как пятерку $M = (Q, \Sigma, P, S, F)$ или пятерку $M = (Q, \Sigma, \delta, S, F)$, в которой Q, Σ, S, F обозначают те же объекты, что и в определении НКА, а функцию переходов δ можно представить как отображение вида $\delta : Q \times \Sigma \to \mathscr{P}(Q)$, что соответствует набору продукций из P.

Отношение выводимости $\Rightarrow^*$ для НКА имеет общий вид отношения выводимости в системах текстовых замен.

Определение 66 (Язык допускаемый НКА). Язык, определенный (допускаемый, распознаваемый) НКА $M = (Q, \Sigma, P, S, F)$, есть множество слов

$$L(M) = \{w \in \Sigma^* \mid q_0 w \Rightarrow^* q,\ q_0 \in S, q \in F\}.$$

Очевидно, что конечные детерминированные автоматы являются частным случаем недетерминированных конечных автоматов, в которых накладывается ограничения на множество начальных состояний S (в детерминированном автомате должно быть точно одно начальное состояние) и множество продукций P (в множестве P должна присутствовать в точности одна продукция для каждой пары «состояние — входной сигнал» в левой части).

Функцию переходов конечного автомата обычно представляют в виде таблицы, называемой таблицей переходов.

Определение 67 (Таблица переходов). Для конечного автомата

$$M = (Q, \Sigma, P, S, F)$$

таблица переходов имеет $\#Q$ строк, соответствующих всевозможным состояниям автомата, и $\#\Sigma$ столбцов, соответствующих всевозможным входным сигналам автомата. В клетке таблицы в строке, соответствующей состоянию q, и в столбце, соответствующем входному сигналу a перечисляются такие состояния r, для которых P присутствует продукция $qa \to r$.

Если в такой таблице каким-то образом выделить начальные и заключительные состояния, то она будет однозначно определять весь автомат. Условимся здесь каждое начальное состояние автомата обозначать стрелкой, а каждое заключительное состояние обозначать звездочкой перед меткой соответствующей строки таблицы.

Пример 36. Определим конечный автомат M_1 для языка $L = \{\alpha 00\beta \mid \alpha, \beta \in \{0,1\}^*\}$ — множества всевозможных цепочек из нулей и единиц, в которых есть два подряд идущих нуля.

Пусть $M_1 = (\{q_0, q_1, q_f\}, \{0,1\}, P, q_0, \{q_f\})$, где P состоит из следующих продукций.

$$q_0 0 \to q_1, \quad q_0 1 \to q_0, \quad q_1 0 \to q_f,$$
$$q_1 1 \to q_0, \quad q_f 0 \to q_f, \quad q_f 1 \to q_f.$$

Множество P можно представить в виде таблицы:

	0	1
$\to q_0$	q_1	q_0
q_1	q_f	q_0
$*q_f$	q_f	q_f

В этом автомате три состояния. Автомат находится в состоянии q_0 до тех пор, пока на входе не будет получен символ 0. Если этот символ получен, то автомат считает его первым нулем из пары подряд идущих нулей и переходит в состояние q_1, где ожидает следующего символа, и если это будет 0, то переходит в заключительное состояние q_f из которого уже не выйдет, но если это будет 1 — то пара нулей не состоялась, и автомат возвращается в состояние q_0.

Например, цепочка 10110001 допускается этим автоматом, т. к. получив ее автомат поведет себя следующим образом

$$q_0 10110001 \Rightarrow q_0 0110001 \Rightarrow q_1 110001 \Rightarrow q_0 10001 \Rightarrow$$
$$\Rightarrow q_0 0001 \Rightarrow q_1 001 \Rightarrow q_f 01 \Rightarrow q_f 1 \Rightarrow q_f$$

Так как q_f — заключительное состояние, то последняя конфигурация в приведенной последовательности является заключительной.

Цепочка 01110 не допускается автоматом, т. к. получив ее автомат сможет применить продукции:

$$q_0 01110 \Rightarrow q_0 01110 \Rightarrow q_1 1110 \Rightarrow q_0 110 \Rightarrow q_0 10 \Rightarrow q_0 0 \Rightarrow q_1.$$

Прочитав всю цепочку, автомат не оказался в заключительном состоянии, а следовательно, цепочка не принадлежит языку, определяемому данным автоматом. □

Пример 37. Рассмотрим автомат

$$M_2 = (\{q_0, q_1, q_2, q_3, q_f\}, \{1, 2, 3\}, P, \{q_0\}, \{q_f\}),$$

в котором множество P определено следующим образом:

$$\begin{array}{lll} q_0 1 \to q_1, & q_0 2 \to q_2, & q_0 3 \to q_3, \\ q_1 1 \to q_1, & q_1 1 \to q_f, & q_1 2 \to q_1, \\ q_1 2 \to q_2, & q_1 3 \to q_1, & q_1 3 \to q_3, \\ q_2 2 \to q_2, & q_2 2 \to q_f, & q_2 1 \to q_1, \\ q_2 1 \to q_2, & q_2 3 \to q_2, & q_2 3 \to q_3, \\ q_3 1 \to q_3, & q_3 1 \to q_f, & q_3 2 \to q_3, \\ q_3 2 \to q_2, & q_3 3 \to q_1, & q_3 3 \to q_3. \end{array}$$

Множество продукций автомата можно переписать в виде таблицы переходов:

	1	2	3
$\to q_0$	q_1	q_2	q_3
q_1	q_1, q_f	q_1, q_2	q_1, q_3
q_2	q_2, q_1	q_2, q_f	q_2, q_3
q_3	q_3, q_1	q_3, q_2	q_3, q_f
$*q_f$	—	—	—

Этот автомат допускает множество всевозможных цепочек, в которых последний символ присутствует в цепочке как минимум еще раз.

Так, например, цепочка 1232 допускается автоматом. Здесь, среди различных вариантов вывода, можно выделить последовательность применения продукций автомата

$$q_0 1232 \Rightarrow q_1 232 \Rightarrow q_2 32 \Rightarrow q_2 2 \Rightarrow q_f,$$

допускающую входную цепочку. □

Определение 68 (Диаграмма переходов). *Диаграмма переходов* для конечного автомата $M = (Q, \Sigma, P, S, F)$ есть ориентированный граф, определяемый следующим образом.

1. Всякому состоянию из Q соответствует некоторая вершина.
2. Пусть $qa \to p$ — продукция автомата для некоторых состояний $q, p \in Q$ и входного символа $a \in \Sigma$. Тогда диаграмма переходов должна содер-

жать дугу из вершины q в вершину p, помеченную a. Если существует несколько входных символов, переводящих автомат из состояния q в состояние p, то диаграмма переходов может содержать одну дугу, отмеченную списком этих символов.
3. Диаграмма содержит стрелку, направленную в каждое начальное состояние, отмеченную как «Начало».
4. Вершины, не принадлежащие F, изображаются одинарным кружком. Вершины, соответствующие заключительным состояниям, помечаются двойным кружком.

Пример диаграммы переходов представлен на рис. 5.1.

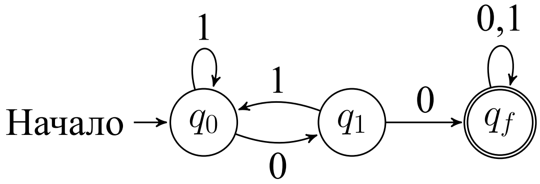


Рис. 5.1. Диаграмма переходов автомата из примера 36.

Определение 69 (Полностью определенный конечный автомат). Конечный автомат $M = (Q, \Sigma, P, S, F)$ называют полностью определенным, если для каждого $q \in Q$ и для каждого $a \in \Sigma$ найдется $r \in Q$ такое, что $qa \to r \in P$.

Любой конечный детерминированный автомат по определению является полностью определенным.

Автомат M_2 из примера 37 не является полностью определенным (нет реакции в заключительном состоянии).

Теорема 13 (Признаки непустоты и бесконечности автоматного языка)**.** *Язык, допускаемый конечным автоматом M с n состояниями*
1) *не пуст тогда и только тогда, когда M допускает цепочку длины, меньшей чем n;*
2) *бесконечный тогда и только тогда, когда он принимает цепочку длины l, $n \leqslant l < 2n$.*

Доказательство. Пусть $M = (Q, \Sigma, P, q_0, F)$ — конечный автомат, $\#Q = n$.

Достаточность условия 1 очевидна. Действительно, если M допускает цепочку длины, меньшей чем n, то язык $L(M)$ уже не пуст.

Необходимость условия 1 следует из следующих рассуждений от противного. Предположим, что $L(M)$ не пуст, но ни одной цепочки длины меньше n в этом языке не существует.

Рассмотрим цепочку $w \in L(M)$. Пусть w — одна из самых коротких таких цепочек. Пусть $w = a_1a_2a_3 \ldots a_m$, где $m \geqslant n$ и каждый a_i — входной символ.

Пусть $p_0 = q_0, p_1, p_2, \ldots, p_m$ — последовательность состояний, через которую переходит автомат при допуске входной цепочки w. То есть

$$p_0 w \Rightarrow p_1 a_2 a_3 \ldots a_m \Rightarrow p_2 a_3 \ldots a_m \Rightarrow^* p_m.$$

Причем $p_m \in F$.

Рассмотрим первые $n+1$ состояний из последовательности p_i для $i = 0, 1, \ldots, n$. Поскольку автомат M имеет n различных состояний, то найдутся два целых числа i и j $(0 \leqslant i < j \leqslant n)$, при которых $p_i = p_j$. Разобьем цепочку w на xyz следующим образом

1) $x = a_1 a_2 \ldots a_i$;
2) $y = a_{i+1} a_{i+2} \ldots a_j$;
3) $z = a_{j+1} a_{j+2} \ldots a_m$.

Цепочка x приводит автомат в состояние p_i, y — из p_i обратно в p_i (так как $p_i = p_j$), а z — остаток цепочки w. То есть

$$p_0 xyz \Rightarrow^* p_i yz \Rightarrow^+ p_j z \Rightarrow^* p_m.$$

Причем цепочка x может быть пустой при $i = 0$, а z — при $j = n = m$. Однако цепочка y не может быть пустой, т. к. i строго меньше j.

Если на вход автомата подать цепочку xz, автомат переходит из начального состояния q_0 в p_i, прочитав x. Поскольку $p_i = p_j$, то z переводит M из состояния p_i в допускающее состояние. Значит $xz \in L(M)$.

В то же время $|xz| < |xyz| = |w|$, но это противоречит предположению, что w — одна из самых коротких цепочек, допускаемых M.

Докажем теперь утверждение 2.

Достаточность условия 2 следует из нижеследующих рассуждений. Пусть существует $w \in L(M)$, причем $n \leqslant |w| < 2n$. Как и в предыдущих рассуждениях можем утверждать, что существуют $q \in Q$ и $q_f \in F$, что $w = xyz$, где $y \neq \varepsilon$, и $q_0 xyz \Rightarrow^* qyz \Rightarrow^+ qz \Rightarrow^* q_f$.

Но тогда цепочки вида $xy^i z \in L(M)$ при любом i.

Действительно, получая x автомат переходит из q_0 в q, затем, читая y^i циклически проходит через q, и, наконец, по z переходит в заключительное состояние. Таким образом, для любого i цепочка $xy^i z$ также распознается автоматом M, а значит, принадлежит языку $L(M)$.

Очевидно, множество $L(M)$ бесконечно.

Необходимость условия 2 доказывается методом от противного. Пусть M принимает бесконечное множество цепочек, причем ни одна из них не имеет длину l, $n \leqslant l < 2n$.

Если бы существовали только цепочки длиной $l < n$, то язык $L(M)$ был бы конечен, но это не так.

Рассмотрим цепочки языка длины $l \geqslant 2n$. Пусть w — одна из самых коротких таких цепочек. Очевидно, существуют $q \in Q$ и $q_f \in F$, что мы можем написать $w = xyz$, где $0 < |y| \leqslant n$, и $q_0xyz \Rightarrow^* qyz \Rightarrow^+ qz \Rightarrow^* q_f$. Но тогда $xz \in L(M)$, поскольку $q_0xz \Rightarrow^* qz \Rightarrow^* q_f$. Из того, что $|w| \geqslant 2n$ следует, что $|xz| \geqslant n$. Т.к. в языке $L(M)$ нет слов длины $n \leqslant l < 2n$, $|xz| \geqslant 2n$. В то же время $|xz| < |xyz| = |w|$, но это противоречит предположению, что w — одна из самых коротких таких цепочек, допускаемых M.

Что и требовалось доказать. □

Теорема 14 (Об автоматности языков типа 3). *Для языка L следующие три условия эквивалентны:*

1) L допускаем конечным детерминированным автоматом;

2) L допускаем конечным недетерминированным автоматом;

3) L — язык типа 3.

Доказательство. Любой КДА (Q, Σ, P, q_0, F) можно представить в виде НКА $(Q, \Sigma, P, \{q_0\}, F)$. Таким образом, следствие 1→2 очевидно.

Рассмотрим предположение 2→3, т. е. покажем, что если язык допускается конечным автоматом, то это леволинейный язык.

Пусть $M = (Q, \Sigma, P, S, F)$ — НКА. Тогда язык определенный этим автоматом можно представить как объединение языков

$$L(M) = \bigcup_{q_0 \in S, q \in F} \{w \in \Sigma^* \mid q_0w \Rightarrow^* q\}. \tag{5.2}$$

Каждый язык в объединении в (5.2) распознается аналитической грамматикой типа 3, чье множество правил состоит из продукций исходного НКА и дополнительной продукции $\varepsilon \to q_0$ и чей заключительный символ q. Так как по теореме 5 объединение языков типа 3 есть язык типа 3, делаем заключение, что язык $L(M)$ — язык типа 3. То есть следствие 2→3 доказано.

Рассмотрим предположение 3→2, т. е. покажем, что любой леволинейный язык может быть определен конечным автоматом.

Пусть задан язык $L = L(G)$, где

$$G = (N, \Sigma, P, S)$$

— аналитическая грамматика типа 3.

Сначала покажем, что для грамматики G можно построить эквивалентную грамматику $G' = (N', \Sigma, P', S)$, каждое правило в которой представлено в одной из следующих форм

$$Ba \to A, \qquad a \in \Sigma,\ A, B \in N, \tag{5.3}$$

$$\varepsilon \to A, \qquad A \in N. \tag{5.4}$$

Кроме правил в форме (5.3) и (5.4) в G могут содержаться также правила вида

$$Ba_1 \dots a_k \to A, \qquad k \geqslant 2,\ a_i \in \Sigma,\ A, B \in N, \tag{5.5}$$

$$a_1 \dots a_k \to A, \qquad k \geqslant 1,\ a_i \in \Sigma,\ A, B \in N, \tag{5.6}$$

$$B \to A, \qquad A, B \in N. \tag{5.7}$$

Таким образом нам необходимо избавиться в G от правил (5.5)-(5.7).

Для каждого правила вида (5.5) введем новые нетерминалы

$$X_1, X_2, \dots, X_{k-1}$$

и заменим (5.5) на набор продукций

$$Ba_1 \to X_1,\ X_1a_2 \to X_2,\ \dots\ X_{k-1}a_k \to A.$$

Для каждого правила вида (5.6) введем новые нетерминалы

$$X_0, X_1, X_2, \dots, X_{k-1}$$

и заменим (5.6) на набор продукций

$$\varepsilon \to X_0,\ X_0a_1 \to X_1,\ X_1a_2 \to X_2,\ \dots,\ X_{k-1}a_k \to A.$$

Легко видеть, что язык, сгенерированный исходной грамматикой, не претерпит изменений в результате таких модификаций множества правил. Эту процедуру следует выполнять, пока в грамматике остаются правила вида (5.5) или (5.6). Пусть в результате получена грамматика G'', содержащая только правила вида (5.3), (5.4) и (5.7) и для которой выполняется $L(G'') = L(G)$.

Для того, чтобы избавиться от правил вида (5.7) сделаем следующее. Для каждого нетерминала $C \in N$ определим множество $U(C) = \{X \in N \mid C \Rightarrow^* X\}$, т. е. множество нетерминалов, которые можно получить из C применением любого количества правил вида (5.7). После этого удалим все продукции вида (5.7) из грамматики. Добавим в множество правил для каждого правила вида $\alpha \to A$ (правила в форме (5.3) или (5.4)) правила $\alpha \to X$ для всех $X \in U(A)$. Полученная грамматика $G' = (N', \Sigma, P', S)$ будет определять тот же язык, что и исходная грамматика G.

Заменим построенную грамматику недетерминированным конечным автоматом $M = (N', \Sigma, P'', N'', \{S\})$, где N'' множество нетерминалов из N', для которых в P' есть правило вида (5.4), P'' — все продукции из P' вида (5.3).

Покажем, что

$$L(M) = L(G'). \tag{5.8}$$

Действительно, пусть $w = a_1 a_2 \ldots a_r \in L(G')$, где $a_i \in \Sigma$, тогда существует вывод

$$a_1 a_2 \ldots a_r \Rightarrow \; X_0 a_1 a_2 \ldots a_r \Rightarrow \; X_1 a_2 \ldots a_r \Rightarrow^* X_{r-1} a_r \Rightarrow S \tag{5.9}$$

по правилам грамматики G', а так же по определению автомата M

$$X_0 a_1 a_2 \ldots a_r \Rightarrow \; X_1 a_2 \ldots a_r \Rightarrow^* X_{r-1} a_r \Rightarrow S. \tag{5.10}$$

С другой стороны, вывод (5.9) может быть получен из (5.10). Следовательно, (5.8) имеет место и верно следствие 3→2.

Рассмотрим предположение 2→1.

Пусть имеется НКА $M = (Q, \Sigma, P, S, F)$. Построим КДА $M' = (Q', \Sigma, P', q_0, F')$, допускающий тот же язык.

Возьмем в качестве множества Q' символов состояний автомата M' — множество всех подмножеств множества Q, т. е. $Q' = \mathscr{P}(Q)$.

Определим q_0 — начальное состояние автомата M' как множество всех стартовых состояний исходного автомата, т. е. $q_0 = S$.

Множество F' заключительных состояний нового автомата определим состоящим из подмножеств множества Q, содержащих хотя бы одно заключительное состояние, т. е.

$$F' = \{q \in Q' \mid q \cap F \neq \varnothing\}.$$

Для каждой пары (q, a), $q \in Q'$, $a \in \Sigma$, добавим в множество продукций P' правило

$$qa \to r, \quad \text{где} \;\; r = \{p \in Q \mid sa \to p \in P, s \in q\}. \tag{5.11}$$

Теперь можно показать, что $L(M) = L(M')$. Пусть $w = a_1 a_2 \ldots a_r \in L(M)$, где $a_i \in \Sigma$. Тогда существует вывод

$$s_0 a_1 a_2 \ldots a_r \Rightarrow \; s_1 a_2 \ldots a_r \Rightarrow^* s_{r-1} a_r \Rightarrow s_f \tag{5.12}$$

согласно продукциям автомата M. По определению автомата M',

$$q_0 a_1 a_2 \ldots a_r \Rightarrow \; q_1 a_2 \ldots a_r \Rightarrow^* q_{r-1} a_r \Rightarrow q_f, \tag{5.13}$$

где $s_i \in q_i$. q_f содержит s_f, а значит, является заключительным состоянием. С другой стороны, вывод (5.12) может быть получен из (5.13). Следовательно $L(M) = L(M')$, а значит верно следствие 2→1.

Теорема доказана. □

Следствие 14.1. *Класс языков $\mathscr{L}_3$ замкнут относительно дополнения и пересечения.*

Доказательство. Пусть L — язык типа 3. Тогда существует конечный детерминированный автомат $M = (Q, \Sigma, P, q_0, F)$ такой, что $L = L(M)$.

Построим автомат

$$M' = (Q, \Sigma, P, q_0, F'),$$

в котором

$$F' = Q - F.$$

Очевидно, что автомат M' будет допускать только те цепочки, которые не допускаются автоматом M и наоборот, т. е.

$$L(M') = \sim L.$$

Так как пересечение можно расписать через операции дополнения и объединения

$$L_1 \cap L_2 = \sim((\sim L_1) \cup (\sim L_2)),$$

по теореме 5, полученный в результате пересечения язык — язык типа 3. □

Следствие 14.2. *Любой язык типа 3 может быть порожден грамматикой*

$$G = (N, \Sigma, P, S),$$

где каждое правило из P имеет одну из двух следующих форм

$$\begin{array}{lll} A & \to Ba, & a \in \Sigma,\ A, B \in N, \\ A & \to \varepsilon, & A \in N. \end{array} \tag{5.14}$$

Доказательство. Пусть L — язык типа 3. Тогда найдется G — аналитическая грамматика типа 3, его допускающая. По алгоритму, приведенному в доказательстве теоремы 14, по грамматике G найдем грамматику G' состоящую только из правил вида (5.3)-(5.4). В грамматике, двойственной к G' правила будут иметь вид (5.14). □

Пример 38. Рассмотрим недетерминированный конечный автомат, заданный следующей таблицей переходов (автомат из примера 37).

	1	2	3
$\to q_0$	q_1	q_2	q_3
q_1	q_1, q_f	q_1, q_2	q_1, q_3
q_2	q_2, q_1	q_2, q_f	q_2, q_3
q_3	q_3, q_1	q_3, q_2	q_3, q_f
$*q_f$	—	—	—

Найдем КДА определяющий тот же язык, что и данный автомат.

В качестве состояний нового автомата возьмем множество всех подмножеств недетерминированного автомата. Для простоты, состояния нового автомата переименуем буквами (или парами букв) латинского алфавита, как показано в таблице переходов на рис. 5.2.

		1	2	3
A	$\varnothing$	A	A	A
→B	$\{q_0\}$	C	D	E
C	$\{q_1\}$	M	K	L
D	$\{q_2\}$	K	O	N
E	$\{q_3\}$	L	N	P
∗F	$\{q_f\}$	A	A	A
G	$\{q_0, q_1\}$	M	K	L
H	$\{q_0, q_2\}$	K	O	N
I	$\{q_0, q_3\}$	L	N	P
∗J	$\{q_0, q_f\}$	C	D	E
K	$\{q_1, q_2\}$	X	X	W
L	$\{q_1, q_3\}$	Y	W	Y
∗M	$\{q_1, q_f\}$	M	K	L
N	$\{q_2, q_3\}$	W	Z	Z
∗O	$\{q_2, q_f\}$	K	O	N
∗P	$\{q_3, q_f\}$	L	N	P
Q	$\{q_0, q_1, q_2\}$	X	X	W
R	$\{q_0, q_1, q_3\}$	Y	W	Y
∗S	$\{q_0, q_1, q_f\}$	M	K	L
T	$\{q_0, q_2, q_3\}$	W	Z	Z
∗U	$\{q_0, q_2, q_f\}$	K	O	N
∗V	$\{q_0, q_3, q_f\}$	L	N	P
W	$\{q_1, q_2, q_3\}$	EE	EE	EE
∗X	$\{q_1, q_2, q_f\}$	X	X	W
∗Y	$\{q_1, q_3, q_f\}$	Y	W	Y
∗Z	$\{q_2, q_3, q_f\}$	W	Z	Z
AA	$\{q_0, q_1, q_2, q_3\}$	EE	EE	EE
∗BB	$\{q_0, q_1, q_2, q_f\}$	X	X	W
∗CC	$\{q_0, q_1, q_3, q_f\}$	Y	W	Y
∗DD	$\{q_0, q_2, q_3, q_f\}$	W	Z	Z
∗EE	$\{q_1, q_2, q_3, q_f\}$	EE	EE	EE
∗FF	$\{q_0, q_1, q_2, q_3, q_f\}$	EE	EE	EE

Рис. 5.2. Таблица переходов КДА, построенного для НКА из примера 37

Определим новое множество продукций так, как это предложено в (5.11). Например, для определения правой части продукции $\text{Н}1 \to x$ вычисляем

$$x = \{r \mid q1 \to r \in P, q \in \text{Н}\}.$$

Так как $\text{Н} = \{q_0, q_2\}$, то множество x составят состояния из правых частей всех продукций, имеющих в левой части $q_0 1$ или $q_2 1$. Таких продукций три:

$$q_0 1 \to q_1, \qquad q_2 1 \to q_1, \qquad q_2 1 \to q_2.$$

Объединяя правые части правил получим $x = \{q_1, q_2\}$, или, в новых обозначениях, $x = \text{К}$. Следовательно, искомая продукция

$$\text{Н}1 \to \text{К}.$$

Сделав подобные построения для каждой пары «состояние, входной сигнал» получим автомат с таблицей переходов, показанной на рис. 5.2.

Полученный автомат является полностью определенным детерминированным конечным автоматом.

В построенном детерминированном автомате есть лишние («бесполезные») состояния. Например из начального состояния В автомат никогда не сможет перейти в состояние АА.

Определение 70 (Недостижимое состояние). Состояние q конечного автомата $M = (Q, \Sigma, P, S, F)$ называется недостижимым, если не найдется начального состояния $q_0 \in S$ и цепочки $\alpha \in \Sigma^*$, приводящей автомат из начального состояния в состояние q. Другими словами $\nexists \alpha \in \Sigma^* : q_0\alpha \Rightarrow^* q$.

Логично предположить, что недостижимые состояния — бесполезные состояния. Удаление из автомата недостижимых состояний не приведет к изменению языка, допускаемого автоматом.

Теорема 15 (О КДА без недостижимых состояний)**.** *Для любого конечного детерминированного автомата M найдется эквивалентный конечный детерминированный автомат M' без недостижимых состояний.*

Доказательство. Пусть $M = (Q, \Sigma, P, q_0, F)$ — заданный КДА.

Возьмем $Q' = \{q_0\}$.

Добавим в Q' всевозможные состояния $r \in Q$ для которых найдется продукция

$$q_0 a \to r \in P$$

для некоторого входного сигнала $a \in \Sigma$.

Если в P найдется продукция $qa \to r$, такая, что $q \in Q'$ и $r \notin Q'$, то добавим r в Q'.

Будем повторять последний шаг, пока он приводит к пополнению множества Q'.

Очевидно, что в множество Q' попадут все достижимые состояния автомата M. Тогда автомат

$$M' = (Q', \Sigma, P', q_0, F'),$$

где $P' = P \cap (Q'\Sigma \times Q')$, $F' = F \cap Q'$, будет допускать тот же язык, что и исходный автомат. $\square$

Пример 39. Рассмотрим автомат, полученный в примере 38. Выполняя алгоритм, изложенный в доказательстве последней теоремы на начальном шаге получим:

$$Q' = \{\mathrm{B}\}.$$

Добавим в Q' все состояния, достижимые из B:

$$Q' = \{\mathrm{B,\ C,\ D,\ E}\}.$$

Последовательно пополняя множество Q' получим:

$$\begin{aligned}
&Q' = \{\mathrm{B,C,D,E},\\
\textit{достижимые из состояния } \mathrm{C}\textit{:}\quad &\mathrm{M,K,L},\\
\textit{достижимые из состояния } \mathrm{D}\textit{:}\quad &\mathrm{O,N},\\
\textit{достижимые из состояния } \mathrm{E}\textit{:}\quad &\mathrm{P},\\
\textit{достижимые из состояния } \mathrm{K}\textit{:}\quad &\mathrm{X,W},\\
\textit{достижимые из состояния } \mathrm{L}\textit{:}\quad &\mathrm{Y},\\
\textit{достижимые из состояния } \mathrm{N}\textit{:}\quad &\mathrm{Z},\\
\textit{достижимые из состояния } \mathrm{W}\textit{:}\quad &\mathrm{EE}\}.
\end{aligned}$$

Остальные 17 состояний, неперечисленные в Q', оказались недостижимыми. Удалив их, получим автомат с таблицей переходов, представленной на рис. 5.3. $\square$

Пример 40. Рассмотрим грамматику $G = (\{S, A, B\}, \{0, 1, 2\}, P, S)$, в которой множество P представлено продукциями

$$\begin{aligned}
S &\to A0 \mid B1 \mid \varepsilon,\\
A &\to S0 \mid B1 \mid 2,\\
B &\to S1 \mid A1.
\end{aligned}$$

		1	2	3
→B	$\{q_0\}$	C	D	E
C	$\{q_1\}$	M	K	L
D	$\{q_2\}$	K	O	N
E	$\{q_3\}$	L	N	P
K	$\{q_1, q_2\}$	X	X	W
L	$\{q_1, q_3\}$	Y	W	Y
∗M	$\{q_1, q_f\}$	M	K	L
N	$\{q_2, q_3\}$	W	Z	Z
∗O	$\{q_2, q_f\}$	K	O	N
∗P	$\{q_3, q_f\}$	L	N	P
W	$\{q_1, q_2, q_3\}$	EE	EE	EE
∗X	$\{q_1, q_2, q_f\}$	X	X	W
∗Y	$\{q_1, q_3, q_f\}$	Y	W	Y
∗Z	$\{q_2, q_3, q_f\}$	W	Z	Z
∗EE	$\{q_1, q_2, q_3, q_f\}$	EE	EE	EE

Рис. 5.3. Таблица переходов КДА из примера 38 без «бесполезных» продукций

Сначала получим двойственную грамматику для данной:

$$\begin{aligned} &A0 \to S, \quad B1 \to S, \quad \varepsilon \to S, \\ &S0 \to A, \quad B1 \to A, \quad 2 \to A, \\ &S1 \to B, \quad A1 \to B. \end{aligned}$$

Избавимся от правила

$$2 \to A,$$

заменив его на

$$\varepsilon \to C, \qquad C2 \to A.$$

По полученному множеству продукций построим конечный недетерминированный автомат $M = (\{S, A, B, C\}, \{0, 1, 2\}, P', \{C, S\}, \{S\})$, в котором P' — множество продукций

$$\begin{aligned} &A0 \to S, \quad B1 \to S, \\ &S0 \to A, \quad B1 \to A, \quad C2 \to A, \\ &S1 \to B, \quad A1 \to B. \end{aligned}$$

Если представить построенный автомат в виде таблицы, получим

	0	1	2
$\to *S$	A	B	—
A	S	B	—
B	—	S, A	—
$\to C$	—	—	A

Из последнего автомата можно получить конечный детерминированный автомат. Если оставить в таком автомате только достижимые состояния, то получим автомат, определенный следующей таблицей переходов (во втором столбце таблицы переходов приведена расшифровка новых обозначений состояний).

		0	1	2
$\to *q_0$	$\{S, C\}$	q_1	q_2	q_1
q_1	$\{A\}$	q_3	q_2	q_4
q_2	$\{B\}$	q_4	q_5	q_4
$*q_3$	$\{S\}$	q_1	q_2	q_4
q_4	$\varnothing$	q_4	q_4	q_4
$*q_5$	$\{S, A\}$	q_5	q_2	q_4

□

Определение 71 (Различение состояний автомата). Пусть задан конечный детерминированный автомат $M = (Q, \Sigma, P, q_0, F)$. q_1 и q_2 — два состояния этого автомата. Говорят, что цепочка α различает состояния q_1 и q_2 если

$$q_1\alpha \Rightarrow^* r_1, \quad q_2\alpha \Rightarrow^* r_2,$$

где $r_1, r_2 \in Q$, причем одно из состояний r_1 или r_2 является заключительным, а второе нет.

Определение 72 (k-неразличимые состояния). Два состояния q_1 и q_2 конечного детерминированного автомата $M = (Q, \Sigma, P, q_0, F)$ называют k-неразличимыми и пишут $q_1 \equiv^k q_2$, когда не найдется цепочки $\alpha \in \Sigma^*$, $|\alpha| \leqslant k$, различающей эти состояния.

Определение 73 (Эквивалентные состояния). Два состояния q_1 и q_2 конечного детерминированного автомата $M = (Q, \Sigma, P, q_0, F)$ называют неразличимыми (эквивалентными) и пишут $q_1 \equiv q_2$, если они k-неразличимы для любого $k \geqslant 0$.

Теорема 16 (Критерий неразличимости состояний автомата)**.** *Для того, чтобы два состояния q_1 и q_2 конечного детерминированного автомата $M = (Q, \Sigma, P, q_0, F)$ были неразличимы необходимо и достаточно, чтобы два этих состояния были $(n-2)$-неразличимы, где n — число состояний автомата M.*

Доказательство.

Необходимость. Тривиально. Если два состояния неразличимы, то они k-неразличимы для любого $k \geqslant 0$ (по определению неразличимости), в том числе и для $k = n - 2$.

Достаточность. Отношение k-неразличимости является отношением эквивалентности на множестве состояний автомата M. Два состояния

$r_1, r_2 \in Q$, принадлежащие разным классам эквивалентности при отношении k-неразличимости, не могут принадлежать одному классу эквивалентности при отношении l-неразличимости, $l > k$, по определению k-неразличимости.

Кроме того, из определения k-неразличимости имеем следующие соотношения.

1. $r_1 \equiv^0 r_2$ тогда и только тогда, когда $r_1 \in F$ и $r_2 \in F$ или $r_1 \notin F$ и $r_2 \notin F$;
2. $r_1 \equiv^{k+1} r_2$ тогда и только тогда, когда $r_1 \equiv^k r_2$ и для любого $a \in \Sigma$, при наличии в P продукций $r_1 a \to s_1$ и $r_2 a \to s_2$ выполняется $s_1 \equiv^k s_2$.

Из этих соотношений следует, что если найдется такое k, что $\equiv^k = \equiv^{k+1}$, то $\equiv^k = \equiv^l$, $\forall l > k$ и $\equiv^k = \equiv$. Действительно, пусть $\equiv^k = \equiv^{k+1}$. Тогда из соотношения 2 следует $r_1 \equiv^{k+1} r_2$ тогда и только тогда, когда $r_1 \equiv^{k+2} r_2$. Так как r_1 и r_2 — любые состояния, принадлежащие одному классу эквивалентности при k-неразличимости, то $\equiv^k = \equiv^{k+1} = \equiv^{k+2}$. Применяя свойство 2 многократно, получим $\equiv^k = \equiv$.

Предположим, что в автомате есть различимые состояния, а значит $F \neq \varnothing$, $F \neq Q$. Тогда $\#F \leqslant n-1$ и $\#(Q \backslash F) \leqslant n-1$. То есть в каждом из классов эквивалентности при отношении 0–неразличимости менее n состояний. Можно производить деление этих классов на новые не более $n-2$ раз. Достаточность доказана. $\square$

На основе последней теоремы существует метод нахождения всех пар эквивалентных состояний КДА, который называется методом заполнения таблицы.

Пусть $M = (Q, \Sigma, P, q_0, F)$ — КДА, в котором $n = \#Q$.

Создадим $n \times n$-таблицу T в которой строки и столбцы соответствуют состояниям данного автомата. Изначально все ячейки таблицы пусты. В дальнейшем будем помещать в клетку таблицы $T(q, r)$ знак $\times$, если состояния q и r различимы. Таблица будет симметрична относительно главной диагонали, а следовательно, когда будем заполнять ячейку таблицы $T(q, r)$, то будем заполнять и ячейку $T(r, q)$. В дальнейшем при заполнении одной ячейки таблицы не будем дополнительно упоминать о заполнении другой.

Если q — заключительное состояние, а r — нет, то q и r — различимые состояния. Помещаем знак $\times$ в ячейку $T(q, r)$.

Пусть q и r — состояния, для которых существует входной сигнал $a \in \Sigma$, такой что $qa \to p \in P$, $ra \to s \in P$ и $T(p, s) = \times$. Тогда (q, r) — пара различимых состояний. Помещаем знак $\times$ в ячейку $T(q, r)$.

Повторяем последний шаг пока он приводит к добавлению в таблицу знака $\times$.

Из теоремы 16 следует, что достаточно $n-2$ раза просмотреть по порядку все пустые клетки таблицы (выше или ниже главной диагонали) чтобы отметить все пары различимых состояний.

Пример 41. Рассмотрим автомат, представленный на рис. 5.4.

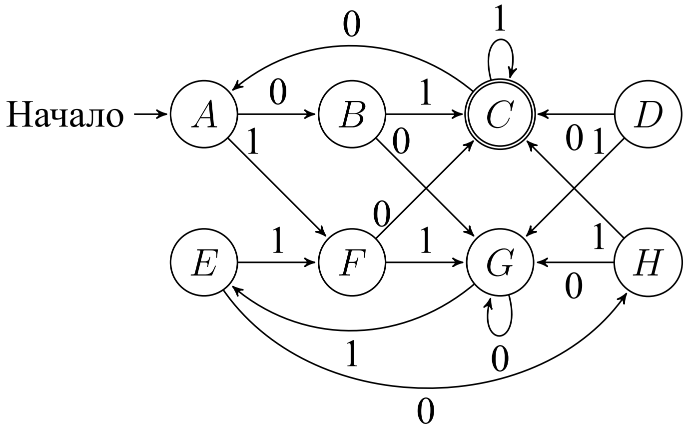


Рис. 5.4. Конечный автомат имеющий неразличимые состояния.

Найдем в нем все пары неразличимых состояний, используя метод заполнения таблицы. В качестве T возюмем 8×8-таблицу, строки и столбцы которой помечены состояниями данного автомата. Так как в представленном автомате только лишь одно состояние C заключительное, то помечаем в таблице T все ячейки в строке, соответствующей состоянию C, и в столбце, соответствующем состоянию C, за исключением ячейки $T(C,C)$. Получим таблицу:

	A	B	C	D	E	F	G	H
A			$\times$					
B			$\times$					
C	$\times$	$\times$		$\times$	$\times$	$\times$	$\times$	$\times$
D			$\times$					
E			$\times$					
F			$\times$					
G			$\times$					
H			$\times$					

Так как $T(C,H)=\times$, а автомат из состояний E и F по сигналу 0 переходит в состояния H и C, то значит пара $\{E,F\}$ —различимые состояния, помечаем $T(E,F)=T(F,E)=\times$. Аналогично, по сигналу 1 автомат переходит из состояния A в F, и из G в E. Но $T(E,F)=\times$, а следовательно помечаем $T(A,G)=T(G,A)=\times$. Продолжаем такие рассуждения, пока это возможно (т. е. пока возможно обнаружить новые пары различимых состояний). Окончательный вариант таблицы T:

	A	B	C	D	E	F	G	H
A		×	×	×		×	×	×
B	×		×	×	×	×	×	
C	×	×		×	×	×	×	×
D	×	×	×		×		×	×
E		×	×	×		×	×	×
F	×	×	×		×		×	×
G	×	×	×	×	×	×		×
H	×		×	×	×	×	×	

Итак, согласно полученной таблице только следующие пары состояний неразличимы: $\{A, E\}$, $\{B, H\}$, $\{D, F\}$. $\square$

Понятие неразличимости (эквивалентности) можно распространить на состояния различных автоматов следующим образом. Пусть $M_1 = (Q_1, \Sigma, P_1, q_1, F_1)$ и $M_2 = (Q_2, \Sigma, P_2, q_2, F_2)$ два КДА. Тогда неразличимость состояний $r_1 \in Q_1$ и $r_2 \in Q_2$ определяется соотношениями:

1. $r_1 \equiv^0 r_2$ тогда и только тогда, когда $r_1 \in F_1$ и $r_2 \in F_2$ или $r_1 \notin F_1$ и $r_2 \notin F_2$;
2. $r_1 \equiv^{k+1} r_2$ тогда и только тогда, когда $r_1 \equiv^k r_2$ и для любого $a \in \Sigma$, при наличии в P_1 продукции $r_1 a \to s_1$ и в P_2 продукции $r_2 a \to s_2$ выполняется $s_1 \equiv^k s_2$.
3. $(r_1 \equiv r_2) \Leftrightarrow (\forall k \geqslant 0,\ r_1 \equiv^k r_2)$.

Определение 74 (Эквивалентные автоматы). Два конечных детерминированных автомата $M_1 = (Q_1, \Sigma, P_1, q_1, F_1)$ и $M_2 = (Q_2, \Sigma, P_2, q_2, F_2)$ называют эквивалентными, если состояние q_1 эквивалентно q_2. Если автоматы M_1 и M_2 не являются эквивалентными, то они различимы.

По конечному автомату можно построить эквивалентный ему конечный детерминированный автомат с минимальным возможным числом состояний. Соответствующий процесс называется минимизацией конечного автомата. Минимизация заключается в удалении всех недостижимых состояний и исключении парных неразличимых состояний.

Теорема 17 (О минимальном автомате)**.** *Для любого конечного детерминированного автомата $M = (Q, \Sigma, P, q_0, F)$ найдется эквивалентный автомат $M' = (Q', \Sigma, P', q_0', F')$ без неразличимых состояний, допускающий тот же язык. Причем, если M — автомат без недостижимых состояний, то не существует КДА, допускающего $L(M)$ и имеющего меньше состояний, чем $\#Q'$.*

Доказательство. Разобьем множество состояний Q на классы эквивалентности, соответствующие отношению неразличимости $\equiv$ (используя выкладки из доказательства теоремы 16 или метод заполнения таблицы) и пусть $[Q]_\equiv$ — совокупность полученных классов эквивалентности.

Определим автомат $M' = (Q', \Sigma, P', q_0', F')$, где

1. $Q' = [Q]_\equiv$;
2. $q_0' = [q_0]_\equiv$;
3. $F' = \big\{[q]_\equiv \in [Q]_\equiv \mid q \in F\big\}$;
4. $P' = \{[q]_\equiv a \to [r]_\equiv \mid$ где $qa \to r \in P\}$.

Легко видеть, что построенный автомат допускает тот же язык, что и исходный автомат.

Пусть в M отсутствуют недостижимые состояния. Покажем, что построенный автомат имеет минимально-возможное число состояний.

Предположим, что существует КДА $M'' = (Q'', \Sigma, P'', q_0'', F'')$, такой, что $L(M'') = L(M)$ и в котором $\#Q'' < \#Q'$. Тогда существуют цепочки γ_1 и γ_2, такие, что эти цепочки переводят автомат M' в разные состояния $q_0'\gamma_1 \Rightarrow_{M'} q_1'$, $q_0'\gamma_2 \Rightarrow_{M'} q_2'$, но при этом переводят автомат M'' в одно и то же состояние $q_0''\gamma_1 \Rightarrow_{M''} q_1''$, $q_0''\gamma_2 \Rightarrow_{M''} q_1''$. Так как в автомате M' нет эквивалентных и недостижимых состояний, найдется цепочка ϑ, что $q_1'\vartheta \Rightarrow_{M'} q_3'$, $q_2'\vartheta \Rightarrow_{M'} q_4'$, где точно одно из состояний q_3', q_4' является заключительным. Очевидно, что точно одна из цепочек $\gamma_1\vartheta$, $\gamma_2\vartheta$ допускается автоматом M', и, следовательно, принадлежит языку $L(M)$. Но в автомате M'', цепочки $\gamma_1\vartheta$, $\gamma_2\vartheta$ переводят автомат в одно и то же состояние. Следовательно, эти цепочки одновременно принадлежат или не принадлежат языку $L(M)$. Получили противоречие.

Теорема доказана. □

Пример 42. Снова рассмотрим автомат, представленный на рис. 5.4. В этом автомате нет недостижимых состояний. Всевозможные пары неразличимых состояний этого автомата найдены в примере 41, а следовательно можно определить совокупность классов эквивалентности отношения неразличимости на множестве состояний автомата

$$[Q_1]_\equiv = \big\{\{A, E\}, \{B, H\}, \{C\}, \{D, F\}\big\}.$$

Построим новый автомат, состояниями которого будут классы эквивалентности из $[Q_1]_\equiv$. В новом автомате объявим начальным состояние, помеченное множеством $\{A, E\}$, т. к. A — начальное состояние исходного автомата. Заключительным состоянием будет $\{C\}$. Из состояния $\{A, E\}$ по сигналу 0 будет переход в $\{B, H\}$, т. к. в исходном автомате по сигналу 0 был переход из A в B. Из состояния $\{A, E\}$ по сигналу 1 автомат перейдет в $\{D, F\}$, т. к. в исходном автомате по сигналу 1 был переход из A в F. И т. д. На рис. 5.5 приведен окончательный вариант минимального автомата. □

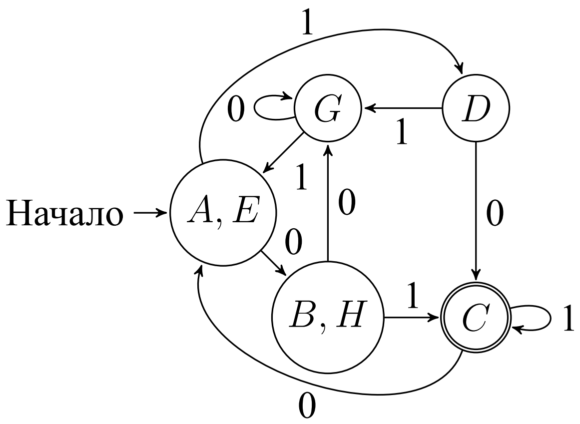


Рис. 5.5. Автомат полученный в результате минимизации автомата, представленного на рис. 5.4.

5.2. Автоматы с магазинной памятью

Определение 75 (Автомат с магазинной памятью). Система текстовых замен $M = (\Omega, P)$ называется *автоматом с магазинной памятью* (МП-автоматом) если

1) в алфавите Ω можно выделить три алфавита Q, Σ и Γ, называемые соответственно множеством символов состояний (множеством состояний), множеством входных сигналов (входным алфавитом) и множеством символов магазинной памяти, причем $Q \cup \Sigma \cup \Gamma = \Omega$;
2) в множестве Q выделяется элемент q_0 называемый начальным (стартовым) состоянием и подмножество F — множество заключительных состояний;
3) в множестве Γ выделяется элемент Γ_0 называемый начальным (стартовым) символом магазинной памяти;
4) любая продукция в P имеет одну из следующих форм:

$$Zq \to \gamma r, \qquad q, r \in Q, \ \gamma \in \Gamma^*, \ Z \in \Gamma, \tag{5.15}$$

$$Zqa \to \gamma r, \qquad q, r \in Q, \ \gamma \in \Gamma^*, \ Z \in \Gamma, \ a \in \Sigma. \tag{5.16}$$

Таким образом, для определения МП-автомата необходимо определить семь компонент

$$(Q, \Sigma, \Gamma, P, q_0, \Gamma_0, F).$$

Отношение выводимости $\Rightarrow^*$ для МП-автоматов имеет общий вид отношения выводимости в системах текстовых замен.

Определение 76 (Язык, допускаемый МП-автоматом). Язык определенный (допускаемый, распознаваемый) МП-автоматом $M = (Q, \Sigma, \Gamma, P, q_0, \Gamma_0, F)$

есть множество слов

$$L(M) = \{w \in \Sigma^* \mid \Gamma_0 q_0 w \Rightarrow^* \gamma q,\ q \in F, \gamma \in \Gamma^*\}.$$

Пример 43. Рассмотрим МП-автомат для языка, $L = \{ww^R | \ w \in (a+b)^*\}$.

$$M = (\{q_0, q_1, g_2\},\ \{a, b\},\ \{a, b, \Gamma_0\},\ P,\ q_0,\ \Gamma_0,\ \{q_0\}),$$

где множество продукций P определяется следующим образом:

$\Gamma_0 q_0 a \to \Gamma_0 a q_1,$	(5.17)	$a q_1 a \to q_2,$	(5.23)
$\Gamma_0 q_0 b \to \Gamma_0 b q_1,$	(5.18)	$b q_1 b \to q_2,$	(5.24)
$a q_1 a \to a a q_1,$	(5.19)	$a q_2 a \to q_2,$	(5.25)
$b q_1 a \to b a q_1,$	(5.20)	$b q_2 b \to q_2,$	(5.26)
$a q_1 b \to a b q_1,$	(5.21)	$\Gamma_0 q_2 \to q_0.$	(5.27)
$b q_1 b \to b b q_1,$	(5.22)		

Данный автомат переносит в магазин очередной символ (продукции (5.17)–(5.22)) до тех пор, пока не решит, что поместил туда всю первую половину слова. После этого на каждом шаге верхний символ магазина автомат сравнивает с очередным входным символом и, если сравнение прошло успешно, удаляет его (продукции (5.23)-(5.26)). Перебрав все такие пары автомат должен опустошить магазин вплоть до символа, отмечающего его дно, после чего переходит в заключительное состояние (продукция (5.27)).

Например этот автомат может допустить цепочку $abbaabba$ с помощью следующего вывода:

$$\Gamma_0 q_0 abbaabba \Rightarrow \Gamma_0 a q_1 bbaabba \Rightarrow \Gamma_0 ab q_1 baabba \Rightarrow \Gamma_0 abb q_1 aabba \Rightarrow$$
$$\Rightarrow \Gamma_0 abba q_1 abba \Rightarrow \Gamma_0 abb q_2 bba \Rightarrow \Gamma_0 ab q_2 ba \Rightarrow \Gamma_0 a q_2 a \Rightarrow \Gamma_0 q_2 \Rightarrow q_0 \quad \square$$

Утверждение 3. *Язык допускается автоматом с магазинной памятью тогда и только тогда, когда он контекстно-свободный.*

Определение 77 (Детерминированный МП-автомат). Автомат с магазинной памятью $M = (Q, \Sigma, \Gamma, P, q_0, \Gamma_0, F)$ называется детерминированным если:

1) для каждых $q \in Q$ и $Z \in \Gamma$ в множестве P содержится точно одна продукция вида (5.15) и нет продукций вида (5.16), или нет продукций вида (5.15) и точно одна продукция вида (5.16) для каждого $a \in \Sigma$;
2) если в (5.15) или (5.16) $Z = \Gamma_0$, то Γ_0 является первым символом в γ.

В последнем определении пункт 1 является необходимым для детерминированного поведения автомата, тогда как пункт 2 следит за тем, чтобы магазин автомата никогда не оказался пустым.

Определение 78 (Детерминированный язык типа 2). Язык типа 2 называют детерминированным, если он допускается детерминированным МП-автоматом.

5.3. Линейно ограниченные автоматы

Определение 79 (Линейно ограниченный автомат). Система текстовых замен $M = (\Omega, P)$ называется *линейно ограниченным автоматом* (ЛО-автоматом) если

1) в алфавите Ω можно выделить два алфавита Q, и Σ, называемые соответственно множеством символов состояний (множеством состояний) и множеством символов ленты (входным алфавитом), причем $Q \cup \Sigma = \Omega$, $Q \cap \Sigma = \varnothing$;
2) в множестве Q выделяется элемент q_0 называемый начальным (стартовым) состоянием, и подмножество F — множество заключительных состояний;
3) любая продукция в P имеет одну из следующих форм

$$qa \to rb, \qquad q, r \in Q,\ a, b \in \Sigma, \tag{5.28}$$

$$qa \to ar, \qquad q, r \in Q,\ a \in \Sigma, \tag{5.29}$$

$$cqa \to rca, \qquad q, r \in Q,\ a, c \in \Sigma. \tag{5.30}$$

Более того, для каждых q, a и r множество P не содержит ни одного правила вида (5.30) или содержит (5.30) для всех $c \in \Sigma$.

Таким образом ЛО-автомат можно представить как набор

$$M = (Q, \Sigma, P, q_0, F).$$

Определение 80 (Язык, допускаемый ЛО-автоматом). Язык определенный (допускаемый, распознаваемый) ЛО-автоматом $M = (Q, \Sigma, P, q_0, F)$ есть множество слов

$$L(M) = \{w \in \Sigma^* \mid q_0 w \Rightarrow^* \alpha q,\ q \in F, \alpha \in \Sigma^*\}.$$

Утверждение 4. *Язык допускается ЛО-автоматом тогда и только тогда, когда он контекстно-зависимый.*

Определение 81 (Детерминированный ЛО-автомат). ЛО-автомат называется детерминированным если для каждых q и a выполнено только одно из следующих условий.

1. В P есть точно одна продукция вида (5.28).
2. В P есть точно одна продукция вида (5.29).
3. В P есть точно один набор продукций вида (5.30), в котором c пробегает все значения из Σ.

5.4. Машины Тьюринга

Определение 82 (Машина Тьюринга). Система текстовых замен $M = (\Omega, P)$ называется *машиной Тьюринга* если

1) в алфавите Ω можно выделить два алфавита Q, и Γ, называемые соответственно множеством символов состояний (множеством состояний) и множеством символов ленты, а также дополнительный символ $\#$, называемый маркером границы строки, не принадлежащий этим двум алфавитам, причем $Q \cup \Gamma \cup \{\#\} = \Omega$;
2) в множестве Q выделяется элемент q_0, называемый начальным (стартовым) состоянием, и подмножество F — множество заключительных состояний;
3) в множестве Γ выделяется подмножество Σ называемое множеством входных сигналов, а также вспомогательный символ o;
4) любая продукция в P имеет одну из следующих форм

$$qa \to rb \qquad \text{(перезапись)}, \tag{5.31}$$
$$qac \to arc \qquad \text{(сдвиг вправо)}, \tag{5.32}$$
$$qa\# \to aro\# \qquad \text{(сдвиг вправо с расширением)}, \tag{5.33}$$
$$cqa \to rca \qquad \text{(сдвиг влево)}, \tag{5.34}$$
$$\#qa \to \#roa \qquad \text{(сдвиг влево с расширением)}, \tag{5.35}$$

где $q, r \in Q$, $a, b, c \in \Gamma$. Более того, для каждых q, a и r множество P не содержит ни одного правила вида (5.32) и (5.33) (соотв. (5.34) и (5.35)) или содержит и (5.32) и (5.33) (соотв. (5.34) и (5.35)) для всех $c \in \Gamma$. Ни для каких q и a цепочка qa не является подцепочкой левой части двух разных продукций из семейства продукций вида (5.31), (5.33) и (5.35).

Таким образом машину Тьюринга можно представить как набор

$$M = (Q, \Sigma, \Gamma, P, \#, o, q_0, F).$$

Определение 83 (Финализирующая цепочка). Для заданной машины Тьюринга $M = (Q, \Sigma, \Gamma, P, \#, o, q_0, F)$ цепочка $qa\alpha$, $q \in Q$, $a \in \Gamma \cup \{\#\}$, $\alpha \in \Gamma^* \cup \{\#\}$, называется *финализирующей*, если qa не является подцепочкой левой части ни одной продукции в P.

Определение 84 (Язык, допускаемый машиной Тьюринга). Язык определенный (допускаемый, распознаваемый) машиной Тьюринга

$$M = (Q, \Sigma, \Gamma, P, \#, o, q_0, F)$$

есть множество слов

$$L(M) = \{w \in \Sigma^* \mid \#q_0w\# \Rightarrow^* \#\alpha q\beta\#, \\ q \in F,\ \alpha, \beta \in \Gamma^*, q\beta\# - \text{финализирующая}\}.$$

Теорема 18 (О классе языков, допускаемых машиной Тьюринга)**.** *Если язык допускается машиной Тьюринга, то это язык типа 0.*

Доказательство. Пусть L — язык, допускаемый машиной Тьюринга

$$M = (Q, \Sigma, \Gamma, P, \#, o, q_0, F).$$

Определим $G = (N, \Sigma, P \cup P', X_0)$ — аналитическую грамматику типа 0, распознающую язык L. В качестве множества нетерминалов возьмем множество

$$N = Q \cup (\Gamma - \Sigma) \cup \{\#\} \cup \{X_0, X_1, X_2\}.$$

Множество продукций P' составим из правил:

$$\begin{array}{llll} \varepsilon \to \#q_0, & \varepsilon \to \#, & qa \to X_1, & X_1b \to X_1, \\ X_1\# \to X_2, & q\# \to X_2, & bX_2 \to X_2, & \#X_2 \to X_0, \end{array}$$

где b — пробегает все символы из Γ, q и a — такие значения, что $q \in F$ и цепочка qa является финализирующей.

Покажем, что $L(G) = L(M)$.

Действительно, если $w \in L(M)$, то по правилам построенной грамматики возможен вывод

$$w \Rightarrow \#q_0w \Rightarrow \#q_0w\# \Rightarrow^* \#\alpha qa\beta\# \Rightarrow^* \#\alpha X_1\beta\# \Rightarrow^* \#\alpha X_2 \Rightarrow^* X_0,$$

или, в случае $w = \varepsilon$

$$w \Rightarrow \#q_0 \Rightarrow \#q_0\# \Rightarrow \#X_2 \Rightarrow X_0.$$

Следовательно $w \in L(G)$.

В обратную сторону, пусть $w \in L(G)$. Если $w = \varepsilon$, то существует вывод по грамматике G из $\#q_0\#$ в X_0. Это подразумевает, что $q_0 \in F$, а следовательно $\varepsilon \in L(M)$. Если же $w \neq \varepsilon$, то по грамматике G существует вывод

$$w \Rightarrow^* \#\alpha qa\beta\# \Rightarrow^* X_0,$$

где $q \in F$, $a \in \Gamma$, $\alpha, \beta \in \Gamma^*$, qa — финализирующая цепочка. Это означает, что $w \in L(M)$. $\square$

Утверждение 5. *Любой язык типа 0 допускается машиной Тьюринга.*

Контрольные вопросы

1. Дайте определение конечного автомата.
2. В чем отличия работы детерминированного автомата и недетерминированного автомата?
3. Какими средствами можно определить конечный автомат?
4. Что представляет из себя диаграмма автомата?
5. Как определяется таблица переходов для конечного автомата?
6. Каким образом автомат может использоваться для определения языков.
7. Дайте определение языка, допускаемого конечным автоматом.
8. Какой класс языков определяется конечными автоматами?
9. Как определить является ли язык типа 3 пустым?
10. Как формируется множество продукций грамматики типа 3 для языка, заданного конечным автоматом?
11. Как определяется множество заключительных состояний при построении конечного автомата по грамматике типа 3?
12. При заданном языке типа 3 как построить конечный автомат для дополнения этого языка?
13. Как определить, что язык типа 3 является бесконечным?
14. В чем отличия конечного автомата и автомата с магазинной памятью?
15. В чем заключается порядок работы автомата с магазинной памятью?
16. Как автомат с магазинной памятью допускает язык?
17. Какой класс языков допускается автоматами с магазинной памятью.
18. Дайте определение языка, допускаемого автоматом в терминах системы текстовых замен.
19. Как определить является ли детерминированным автомат с магазинной памятью?
20. Дайте определение линейного ограниченного автомата.
21. Дайте определение языка, допускаемого линейным ограническнным автоматом.
22. Какой класс языков определяется линейными ограниченными автоматами?
23. Какую роль играет понятие финализирующей цепочки в терминах машин Тьюринга?
24. Как построить грамматику для языка, допускаемого машиной Тьюринга?

Глава 6
Регулярные и контекстно-свободные языки

6.1. Регулярные множества и регулярные выражения

Определение 85 (Регулярное множество). Пусть задан некоторый алфавит Σ. Тогда *регулярное множество* можно определить следующим образом.

1. $\varnothing$ — регулярное множество.
2. $\{\varepsilon\}$ — регулярное множество.
3. Если $a \in \Sigma$, то $\{a\}$ — регулярное множество.
4. Если L_1 и L_2 — регулярные множества, то $L_1 \cup L_2$, $L_1 L_2$ и L_1^* — регулярные множества.

Пример 44. Рассмотрим множество $L_1 = \{ab, ba\}$. Его можно представить в виде $L_1 = (AB) \cup (BA)$, где $A = \{a\}$, а $B = \{b\}$. A и B — регулярные множества, следовательно регулярны и множества AB и BA, а значит регулярно и L. $\square$

Пример 45. Рассмотрим множество $L_2 = \{01^k \mid k \geqslant 0\}$. Это множество можно представить в виде $L_2 = AB$, где $A = \{0\}$, а $B = \bigcup\limits_{k=0}^{\infty} \{1\}^k = \{1\}^*$. Так как A — регулярное множество, и B — итерация регулярного множества, — регулярное множество, то L_2 — регулярное множество. $\square$

Определение 86 (Регулярный язык). Язык, являющийся регулярным множеством, называют регулярным.

Для записи регулярных множеств используют так называемые регулярные выражения.

Определение 87 (Регулярное выражение). Регулярные выражения определяются следующим образом.

1. $\varnothing$ — регулярное выражение для пустого множества.
2. ε — регулярное выражение для множества $\{\varepsilon\}$.
3. Если $a \in \Sigma$, то a — регулярное выражение для множества $\{a\}$.
4. Если α — регулярное выражение для регулярного множества A и β — регулярное выражение для регулярного множества B, то $\alpha + \beta$, $(\alpha)(\beta)$,

$(\alpha)^*$ и $(\alpha)^+$ — регулярные выражения для регулярных множеств $A \cup B$, AB, A^*, A^+ соответственно.

В регулярных выражениях для операций приняты следующие приоритеты: самый высокий приоритет у операций итерации и положительной итерации, более низкий приоритет у операции конкатенации, и самый низкий приоритет у операции объединения.

Круглые скобки в регулярных выражениях можно опускать в случаях, если это не приведет к нарушению порядка выполнения операций.

Таким образом множество L_1 из примера 44 можно задать с помощью регулярного выражения $ab + ba$, а множество L_2 из примера 45 — с помощью выражения 01^*.

Для любого регулярного языка можно найти множество различных регулярных выражений определяющих его. Например легко проверить, что регулярные выражения $0 + 01^+$, $0(\varepsilon + 1 + 1^*)^*$ так же будут определять язык L_2 из примера 45. Регулярные выражения, определяющие одно и то же регулярное множество, называют эквивалентными.

Принимая во внимание, что регулярные выражения представляют собой запись операций над множествами, можно определить ряд закономерностей для эквивалентных преобразований регулярных выражений. Некоторые из них представлены в следующей лемме.

Лемма 1 (Эквивалентные регулярные выражения)**.** *Если α, β и γ — регулярные выражения, то справедливы следующие соотношения.*

1) $\alpha + \beta = \beta + \alpha$,	8) $\varnothing^* = \varepsilon$,
2) $\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$,	9) $\alpha(\beta\gamma) = (\alpha\beta)\gamma$,
3) $\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma$,	10) $(\alpha + \beta)\gamma = \alpha\gamma + \beta\gamma$,
4) $\alpha\varepsilon = \varepsilon\alpha = \alpha$,	11) $\varnothing\alpha = \alpha\varnothing = \varnothing$,
5) $\alpha^* = \alpha + \alpha^*$,	12) $(\alpha^*)^* = \alpha^*$,
6) $\alpha + \alpha = \alpha$,	13) $\alpha + \varnothing = \alpha$,
7) $\alpha^+ = \alpha\alpha^*$,	14) $\alpha(\beta\alpha)^* = (\alpha\beta)^*\alpha$.

Доказательство леммы напрямую следует из свойств операций над языками.

Пример 46. Рассмотрим регулярное выражение $\gamma = (0^* + \varepsilon)^*(1 + 1^+)$. По определению операции итерации имеем $\gamma = ((0^+ + \varepsilon) + \varepsilon)^*(1 + 1^+)$. В силу соотношения 2 получим $\gamma = (0^+ + \varepsilon + \varepsilon)^*(1 + 1^+)$. Применяем соотношение 6, получая $\gamma = (0^+ + \varepsilon)^*(1 + 1^+)$. Применяем определение операции положительной итерации, получим $\gamma = (0^*)^*(1+1^+)$. По соотношению 12 $\gamma = 0^*(1+1^+)$. Из 3 — $\gamma = 0^*1(\varepsilon+1^*)$. Применяя соотношения 2, 6 и определение операции итерации получим $\gamma = 0^*11^*$. По определению операции положительной итерации $\gamma = 0^*1^+$.

Общее преобразование можно провести намного быстрее если помнить, что операции конкатенации и объединения — это операции над множествами. □

Определение 88 (Уравнение с регулярными коэффициентами). Уравнением с регулярными коэффициентами называется уравнение вида

$$X = f(X),$$

где f — регулярное выражение, а X — переменная.

Решением уравнения с регулярными коэффициентами является множество слов, подставляя которое вместо имени переменной в левую и правую части уравнения получают тождество.

Пример 47. Рассмотрим уравнение с регулярными коэффициентами

$$X = 0X1 + \varepsilon.$$

Решением данного уравнения является множество слов $X = \{0^n1^n \mid n \geqslant 0\}$. Действительно, если подставим это решение в правую часть уравнения, то получим

$$0X1 + \varepsilon = 0\{0^n1^n \mid n \geqslant 0\}1 + \varepsilon = \{0^{n+1}1^n \mid n \geqslant 0\}1 + \varepsilon = \\ \{0^{n+1}1^{n+1} \mid n \geqslant 0\} + \varepsilon = \{0^k1^k \mid k \geqslant 1\} + \varepsilon = \{0^k1^k \mid k \geqslant 0\} = X.$$

Нас будут интересовать уравнения с регулярными коэффициентами, решением которых являются регулярные множества, т. е. множества, которые можно задать с помощью регулярных выражений.

Определение 89 (Леволинейное уравнение с регулярными коэффициентами). Леволинейным уравнением с регулярными коэффициентами называется уравнение вида

$$X = X\alpha + \beta,$$

где X — переменная, а α, β — регулярные выражения без переменных.

Лемма 2 (О решении леволинейного уравнения с регулярными коэффициентами)**.** *Решением леволинейного уравнения с регулярными коэффициентами*

$$X = X\alpha + \beta, \tag{6.1}$$

является значение

$$X = \beta\alpha^*. \tag{6.2}$$

Доказательство. Подставив решение (6.2) в обе части (6.1), получим

$$\beta\alpha^* = \beta\alpha^*\alpha + \beta.$$

Преобразуем правую часть:

$$\beta\alpha^* = \beta(\alpha^+ + \varepsilon)$$

и далее

$$\beta\alpha^* = \beta\alpha^*.$$

Получили тождество. □

Замечание 4. В случае, если в (6.1) множество, определенное регулярным выражением α, содержит ε, то уравнение (6.1) имеет бесконечно много решений вида

$$X = (\beta + \gamma)\alpha^*,$$

для любого языка γ (не обязательно регулярного).

В случае наличия множества решений одного уравнения рассматривают в качестве единственного решения так называемую минимальную неподвижную точку.

Определение 90 (Минимальная неподвижная точка уравнения с регулярными коэффициентами). Минимальной неподвижной точкой уравнения с регулярными коэффициентами $X = f(X)$ является такое X', что при наличии любого другого решения X'' выполняется соотношение $X' \subseteq X''$.

Минимальной неподвижной точкой леволинейного уравнения с регулярными коэффициентами является решение (6.2).

Пример 48. Пусть задано уравнение

$$X = X0 + X1(11 + 0)^+ + X + 0 + 01.$$

Вынесем X за скобку, получим

$$X = X(0 + 1(11 + 0)^+ + \varepsilon) + 0 + 01.$$

Здесь $\alpha = 0 + 1(11 + 0)^+ + \varepsilon$, а $\beta = 0 + 01$. Минимальной неподвижной точкой уравнения будет

$$X = (0 + 01)(0 + 1(11 + 0)^+ + \varepsilon)^*.$$ □

Замечание 5. Аналогичным образом можно ввести праволинейное уравнение с регулярными коэффициентами — уравнение вида $X = \alpha X + \beta$. Минимальной неподвижной точкой такого уравнения будет $X = \alpha^*\beta$.

Определение 91 (Леволинейная система уравнений с регулярными коэффициентами). Стандартной леволинейной системой уравнений с регулярными коэффициентами называют систему вида

$$
\begin{aligned}
&X_1 = X_1\alpha_{11} + X_2\alpha_{12} + X_3\alpha_{13} + \cdots X_n\alpha_{1n} + \beta_1,\\
&X_2 = X_1\alpha_{21} + X_2\alpha_{22} + X_3\alpha_{23} + \cdots X_n\alpha_{2n} + \beta_2,\\
&\ldots\\
&X_n = X_1\alpha_{n1} + X_2\alpha_{n2} + X_3\alpha_{n3} + \cdots X_n\alpha_{nn} + \beta_n,
\end{aligned}
\tag{6.3}
$$

где X_i — переменные, а α_{ij}, β_i — регулярные выражения без переменных, $0 < i, j \leqslant n$.

Определение 92 (Минимальная неподвижная точка системы уравнений с регулярными коэффициентами). Минимальной неподвижной точкой системы уравнений с регулярными коэффициентами (6.3) является такое решение $X'_1, X'_2, \ldots, X'_n$, что при наличии любого другого решения $X''_1, X''_2, \ldots, X''_n$ выполняется соотношение $X'_i \subseteq X''_i$ для всех i, $1 \leqslant i \leqslant n$.

Пользуясь выражением (6.2) для решения леволинейного уравнения с регулярными коэффициентами можно решать систему уравнений (6.3) методом исключения переменных. То есть, например, из первого уравнения системы (6.3) можно получить

$$
\begin{aligned}
X_1 = (X_2\alpha_{12} + X_3\alpha_{13} + \cdots X_n\alpha_{1n} + \beta_1)\alpha_{11}^* = \\
X_2\alpha_{12}\alpha_{11}^* + X_3\alpha_{13}\alpha_{11}^* + \cdots X_n\alpha_{1n}\alpha_{11}^* + \beta_1\alpha_{11}^*.
\end{aligned}
$$

Тем самым, можем исключить переменную X_1 из остальных уравнений, подставив в них полученное выражение вместо X_1. Исключая последовательно все переменные получим для последней из них решение, не зависящее от других переменных. Проводя подстановки полученных решений в обратном порядке найдем решение для всей системы уравнений.

Можно доказать, что такой метод исключения состояний выдает минимальную неподвижную точку леволинейной системы уравнений с регулярными коэффициентами.

Пример 49. Рассмотрим следующую систему уравнений.

$$
\begin{aligned}
&X_1 = X_2 0 + X_1 1 + \varepsilon,\\
&X_2 = X_3 0 + X_2 1,\\
&X_3 = X_1 0 + X_3 1.
\end{aligned}
$$

Первое уравнение представим в виде $X_1 = (X_2 0 + \varepsilon)1^*$, подставим во второе и третье уравнение и раскроем скобки. Получим новую систему

$$
\begin{aligned}
&X_1 = X_2 01^* + 1^*,\\
&X_2 = X_3 0 + X_2 1,\\
&X_3 = X_2 01^*0 + 1^*0 + X_3 1.
\end{aligned}
$$

Второе уравнение представим в виде $X_2 = X_3 01^*$, подставим в третье уравнение, получим

$$\begin{aligned}
&X_1 = X_2 01^* + 1^*,\\
&X_2 = X_3 01^*,\\
&X_3 = X_3 01^* 01^* 0 + 1^* 0 + X_3 1.
\end{aligned}$$

Получим решение для X_3 и подставим его в первое и второе уравнения. Получим

$$\begin{aligned}
&X_1 = X_2 01^* + 1^*,\\
&X_2 = 1^* 0 (01^* 01^* 0 + 1)^* 01^*,\\
&X_3 = 1^* 0 (01^* 01^* 0 + 1)^*.
\end{aligned}$$

Полученное решение для X_2 подставим в X_1.

$$\begin{aligned}
&X_1 = 1^* 0 (01^* 01^* 0 + 1)^* 01^* 01^* + 1^*,\\
&X_2 = 1^* 0 (01^* 01^* 0 + 1)^* 01^*,\\
&X_3 = 1^* 0 (01^* 01^* 0 + 1)^*.
\end{aligned}$$

Полученные регулярные выражения для X_i, $i = 1, 2, 3$, есть решение исходной системы. □

Теорема 19 (О регулярности языков типа 3)**.** *Язык определяется регулярным выражением тогда и только тогда, когда это леволинейный язык.*

Доказательство. Пусть L определен регулярным выражением над алфавитом $\Sigma = \{a_1, \ldots, a_r\}$. Это означает, по определению, что L получен из конечных языков

$$\varnothing, \{a_1\}, \ldots, \{a_r\},$$

с помощью конечного числа регулярных операций. По теореме 5 следует, что L — язык типа 3.

В обратную сторону. Пусть L — язык типа 3, определенный грамматикой $G = (N, \Sigma, P, S)$. Представим каждое правило $A \to \alpha \in P$ в виде уравнения $A = \alpha$ относительно переменных — нетерминальных символов. Сложим правые части уравнений, имеющих одинаковые левые части. Получим систему уравнений с регулярными коэффициентами относительно множества переменных N. Решая эту систему получим регулярное выражение для нетерминала S — регулярное выражение для языка, определенного грамматикой G. □

Пример 50. Рассмотрим регулярное выражение $(0+1)^*2$. В этом регулярном выражении присутствуют операции объединения, итерации и конкатенации. Построим праволинейную грамматику определяющую тот же язык.

Для языков $\{0\}$, $\{1\}$ и $\{2\}$ определим грамматики (по Теореме 4)

$$\begin{aligned} G_0 &= (\{S_0\}, \{0,1,2\}, \{S_0 \to 0\}, S_0),\\ G_1 &= (\{S_1\}, \{0,1,2\}, \{S_1 \to 1\}, S_1),\\ G_2 &= (\{S_2\}, \{0,1,2\}, \{S_2 \to 2\}, S_2), \end{aligned}$$

соответственно.

Построим для регулярного выражения $0+1$ грамматику

$$G_3 = (\{S_3, S_0, S_1\}, \{0,1,2\}, P_3, S_3)$$

по грамматикам G_0 и G_1 так как описано в Теореме 5. Здесь P_3 будет состоять из продукций:

$$S_3 \to S_0 \mid S_1, \qquad S_0 \to 0, \qquad S_1 \to 1.$$

По Теореме 5 для регулярного выражения $(0+1)^*$ по грамматике G_3 построим грамматику $G_4 = (\{S_4, S_3, S_0, S_1\}, \{0,1,2\}, P_4, S_4)$, где P_4 — это следующее множество правил:

$$S_4 \to \varepsilon \mid S_3, \qquad S_3 \to S_0 \mid S_1, \qquad S_0 \to 0 \mid S_3 0, \qquad S_1 \to 1 \mid S_3 1.$$

Согласно Теореме 5 для регулярного выражения $(0+1)^*2$ по грамматикам G_4 и G_2 построим грамматику

$$G_5 = (\{S_4, S_3, S_0, S_1, S_2\}, \{0,1,2\}, P_5, S_2),$$

где P_5 — это следующее множество правил:

$$S_2 \to S_4 2, \quad S_4 \to \varepsilon \mid S_3, \quad S_3 \to S_0 \mid S_1, \quad S_0 \to 0 \mid S_3 0, \quad S_1 \to 1 \mid S_3 1.$$

Полученная грамматика является леволинейной грамматикой, определяющей тот же язык, что и исходное регулярное выражение. $\square$

Пример 51. Рассмотрим грамматику

$$\begin{aligned} S &\to A0 \mid B1 \mid \varepsilon,\\ A &\to A0 \mid B1 \mid 2,\\ B &\to B0 \mid A1. \end{aligned}$$

Заменим множество правил на систему уравнений с регулярными коэффициентами

$$\begin{aligned}
S &= A0 + B1 + \varepsilon,\\
A &= A0 + B1 + 2,\\
B &= B0 + A1.
\end{aligned}$$

Выражаем B из последнего уравнения $B = A10^*$ и подставляем в первое во второе уравнения, получим

$$\begin{aligned}
S &= A0 + A10^*1 + \varepsilon,\\
A &= A0 + A10^*1 + 2,\\
B &= A10^*.
\end{aligned}$$

Выражаем A из второго уравнения $A = 2(0 + 10^*1)^*$ и подставляем в первое уравнение. Получим

$$\begin{aligned}
S &= 2(0 + 10^*1)^*0 + 2(0 + 10^*1)^*10^*1 + \varepsilon,\\
A &= 2(0 + 10^*1)^*,\\
B &= A10^*.
\end{aligned}$$

Можно преобразовать выражение для S к виду $S = 2(0 + 10^*1)^+ + \varepsilon$. Это выражение и будет являться регулярным выражением определяющим тот же язык, что и исходная праволинейная грамматика. □

Теорема 20 (О замкнутости языков типа 3 относительно зеркального отражения и подстановки)**.** *Класс леволинейных языков замкнут относительно зеркального отражения, подстановки и произвольного гомоморфизма.*

Доказательство. Пусть L — леволинейный язык, определенный регулярным выражением γ. Тогда L^R определяется регулярным выражением γ', полученным из γ заменой $(\alpha\beta)$ на $(\beta\alpha)$ для всевозможных α и β, т. е. заменой порядка умножения на противоположный во всех конкатенациях.

Так как любой леволинейный язык можно определить регулярным выражением, операция подстановки заключается в подстановке одних регулярных выражений в другие. □

6.2. Самовставленные языки

Определение 93 (Линейная грамматика). Грамматика $G = (N, \Sigma, P, S)$ называется *линейной*, если любая продукция в множестве P находится в одной из следующих форм

$$A \to \alpha B \beta, \qquad (*)$$
$$A \to \gamma,$$

где $A, B \in N$, $\alpha, \beta, \gamma \in \Sigma^*$. Если в линейной грамматике во всех правилах вида (*) $\alpha = \varepsilon$, то грамматика называется леволинейной. Если в линейной грамматике во всех правилах вида (*) $\beta = \varepsilon$, то грамматика называется праволинейной.

Теорема 21 (О праволинейности языков типа 3)**.** *Классы леволинейных и праволинейных языков совпадают.*

Доказательство. Пусть G — порождающая леволинейная грамматика. Обозначим через G^R грамматику полученную из G заменой правой части каждого правила на ее зеркальное отражение. Тогда G^R — праволинейная грамматика. Более того, G^R порождает язык $(L(G))^R$.

И в обратную сторону. Если G — праволинейная грамматика, то G^R — грамматика типа 3, порождающая язык $(L(G))^R$.

На основании теоремы 20 Теорема 21 доказана. □

Определение 94 (Самовставленная грамматика). Контекстно-свободная грамматика $G = (N, \Sigma, P, S)$ называется *самовставленной* (самовложенной), если для некоторых $A \in N$ и $\alpha, \beta \in \Sigma^+$ найдется вывод $A \Rightarrow_G^* \alpha A \beta$
.

Определение 95 (Самовставленный язык). Язык типа 2 называется *самовставленным* (самовложенным) языком, если все грамматики его определяющие — самовставленные.

Теорема 22 (О несамовставленных языках типа 2)**.** *Язык является языком типа 3 если и только если он несамовставленный язык типа 2.*

Доказательство. Так как нет самовставленных грамматик типа 3, делаем заключение, что если язык самовставленный, он не является языком типа 3.

В обратную сторону, пусть L порождается несамовставленной КС-грамматикой $G = (N, \Sigma, P, S)$. Не теряя общности, предположим, что для любого $A \in N$ найдется вывод по правилам грамматики $S \Rightarrow^* \alpha A \beta$ для некоторых $\alpha, \beta \in (N + \Sigma)^*$ (т. к. иначе все продукции, включающие A могут быть удалены из множества правил P без изменения языка L). Рассмотрим 2 случая.

1. Пусть для каждого $A \in N$ найдется вывод из $A \Rightarrow^* \alpha S \beta$ для некоторых $\alpha, \beta \in (N + \Sigma)^*$. Любая продукция в P, содержащая нетерминалы в правой части имеет одну из форм

$$A \to \alpha B \beta, \tag{6.4}$$
$$A \to \alpha B, \tag{6.5}$$
$$A \to B \beta, \tag{6.6}$$
$$A \to B, \tag{6.7}$$

   где $A, B \in N$, $\alpha, \beta \in (N + \Sigma)^+$.
   Если P содержит продукцию вида (6.4), то у нас должен иметься вывод

$$A \Rightarrow \alpha B \beta \Rightarrow^* \alpha \alpha_1 S \beta_1 \beta \Rightarrow^* \alpha \alpha_1 \alpha_2 A \beta_2 \beta_1 \beta,$$

   для некоторых цепочек $\alpha_1, \alpha_2, \beta_1, \beta_2 \in (N + \Sigma)^*$. Так как $\alpha \neq \varepsilon$ и $\beta \neq \varepsilon$, то получили, что грамматика G самовставленная, что невозможно. Такое же противоречие получаем, когда грамматика G содержит правила вида (6.5) и (6.6) одновременно. Таким образом если P содержит правила в форме (6.5), то во всех правилах вида (6.5) цепочка α — терминальная. Это означает, что грамматика G — праволинейная. Аналогично, если P содержит правила в форме (6.6), то грамматика G — леволинейная. В любом случае G — грамматика, определяющая язык типа 3 по Теореме 21. Очевидно, что $L(G)$ — язык типа 3, если в P нет продукций вида (6.4)-(6.6).
2. Пусть существует нетерминал $A' \in N$ такой, что не существует вывода $A' \Rightarrow^* \alpha S \beta$ ни для каких $\alpha, \beta \in (N + \Sigma)^*$. Доказательство того, что язык $L(G)$ — язык типа 3 проведем по индукции по числу n нетерминалов в грамматике G.
   Если $n = 1$ случай 2 невозможен, т. к. $S \Rightarrow^* S$, а значит язык $L(G)$ — язык типа 3, по случаю 1.
   Предположим, что для $n = k$ предположение верно и $L(G)$ — язык типа 3.
   Пусть в грамматике $k + 1$ нетерминал. Рассмотрим грамматику

$$G_1 = (N - \{S\}, \Sigma, P', A'),$$

   полученную из G удалением всех продукций, содержащих S, а также рассмотрим грамматику

$$G_2 = (N - \{A'\}, \Sigma + \{A'\}, P'', S),$$

   полученную из G удалением всех продукций, содержащих A' в левой части.

Оба языка $L(G_1)$ и $L(G_2)$ являются языками типа 3 по предположению индукции или по условию случая 1.

Очевидно, что $L(G)$ — результат подстановки $L(G_1)$ вместо A' в $L(G_2)$. По теореме 20 язык $L(G)$ — язык типа 3.

□

Теорема 23 (Об эквивалентности представлений языков типа 3)**.** *Для языка L следующие условия эквивалентны.*

1. *L — язык типа 3.*
2. *L порожден праволинейной грамматикой.*
3. *L определяется регулярным выражением.*
4. *L допускаем конечным детерминированным автоматом.*
5. *L допускаем конечным недетерминированным автоматом.*
6. *L — несамовставленный язык типа 2.*

6.3. Лемма о накачке для регулярных языков

Теорема 24 (Лемма о накачке (о разрастании) регулярных языков)**.** *Пусть L — регулярный язык. Существует константа n, зависящая от L, для которой каждую цепочку $w \in L$, удовлетворяющую неравенству $|w| \geqslant n$, можно разбить на три цепочки $w = xyz$ так, что выполняются следующие условия.*

1. *$y \neq \varepsilon$.*
2. *$|xy| \leqslant n$.*
3. *Для любого $k \geqslant 0$ цепочка xy^kz также принадлежит L.*

Доказательство. Пусть L — регулярный язык. Тогда $L = L(M)$ для некоторого детерминированного конечного автомата $M = (Q, \Sigma, P, q_0, F)$. Пусть $\#Q = n$. Рассмотрим произвольную цепочку w длиной не менее n, скажем $w = a_1a_2a_3 \dots a_m$, где $m \geqslant n$ и каждый a_i — входной символ. Пусть $p_0 = q_0, p_1, p_2, \dots, p_m$ — последовательность состояний, через которую переходит автомат при приеме входной цепочки w. Т. е.

$$p_0w \Rightarrow p_1a_2a_3 \dots a_m \Rightarrow p_2a_3 \dots a_m \Rightarrow^* p_m.$$

Рассмотрим первые $n + 1$ состояний из последовательности p_i для $i = 0, 1, \dots, n$. Поскольку автомат M имеет n различных состояний, то найдутся два целых числа i и j ($0 \leqslant i < j \leqslant n$), при которых $p_i = p_j$. Теперь разобьем цепочку w на xyz.

1. $x = a_1a_2 \dots a_i$.
2. $y = a_{i+1}a_{i+2} \dots a_j$.
3. $z = a_{j+1}a_{j+2} \dots a_m$.

Таким образом, x приводит автомат в состояние p_i, y — из p_i обратно в p_i (так как $p_i = p_j$), а z — остаток цепочки w. Т. е.

$$p_0 xyz \Rightarrow^* p_i yz \Rightarrow^* p_j z \Rightarrow^* p_m.$$

Причем цепочка x может быть пустой при $i = 0$, а z — при $j = n = m$. Однако цепочка y не может быть пустой, т. к. i строго меньше j.

Когда на вход автомата поступает цепочка xy^kz для $k = 0$ автомат переходит из начального состояния q_0 в p_i, прочитав x. Поскольку $p_i = p_j$, то z переводит M из состояния p_i в допускающее состояние.

Если же $k > 0$, то при x автомат переходит из q_0 в p_i, затем, читая y^k циклически проходит через p_i, и, наконец, по z переходит в заключительное состояние. Таким образом, для любого $k \geqslant 0$ цепочка xy^kz также распознается автоматом M, а значит, принадлежит языку L. $\square$

С помощью леммы о накачке можно доказать нерегулярность некоторого предлагаемого языка.

Пример 52. Покажем, что язык $L = \{0^n1^n \mid n \geqslant 0\}$ нерегулярен. Предположим, что он регулярен. Тогда, согласно лемме о накачке регулярных языков, достаточно длинная цепочка $w = 0^m1^m$ этого языка может быть представлена в виде $w = xyz$, где $y \neq \varepsilon$, и любая цепочка xy^kz, $k \geqslant 0$, будет принадлежать языку L, а значит представима в виде 0^r1^r для какого-то соответствующего r. Предположим, что так оно и есть.

Пусть $y = 0^r1^s$, где r и s положительные числа, одновременно не обращающиеся в 0. Значит $w = 0^{m-r}0^r1^s1^{m-s}$. Цепочка $xy^2z = 0^{m-r}0^r1^s0^r1^s1^{m-s}$ также должна принадлежать языку L.

Предположим $r = 0$, то получим, что $xy^2z = 0^m1^s1^s1^{m-s} = 0^m1^{m+s}$, но из предположения $y \neq \varepsilon$ следует $s \neq m + s$, а значит цепочка $xy^2z = 0^m1^{m+s}$ не принадлежит языку L, а значит предположение $r = 0$ — неверно. Значит $r \neq 0$. Аналогично показывается $s \neq 0$.

Так как $r \neq 0$ и $s \neq 0$, то цепочка $xy^2z = 0^{m-r}0^r1^s0^r1^s1^{m-s}$ должна принадлежать L. Но это не так, из-за того, что в последней цепочке есть подцепочка 10, чего в цепочках языка L случаться не может.

Мы перебрали все возможные случаи r и s и не нашли подходящей пары, а значит нет такого представления $w = xyz$, а значит язык L нерегулярен. $\square$

Пример 53. Докажем нерегулярность языка L_{pr}, образованного всеми цепочками из единиц, длины которых — простые числа. Предположим, что L_{pr} — регулярен. Тогда должна существовать константа n, удовлетворяющая условиям леммы о накачке. Рассмотрим некоторое простое число $p \geqslant n + 2$. Такое число должно существовать, поскольку множество простых чисел бесконечно. Пусть $w = 1^p$.

Согласно лемме о накачке можно разбить цепочку $w = xyz$ так, что $y \neq \varepsilon$ и $|xy| \leqslant n$. Пусть $|y| = m$. Тогда $|xz| = p - m$. Рассмотрим цепочку $xy^{p-m}z$, которая по лемме о накачке должна принадлежать языку L_{pr}, если он действительно регулярен. Однако

$$|xy^{p-m}z| = |xz| + (p-m)|y| = p - m + (p-m)m = (m+1)(p-m).$$

Похоже, что число $|xy^{p-m}z|$ не простое, так как имеет два множителя $m+1$ и $p-m$. Однако нужно еще убедиться, что ни один из этих множителей не равен 1, потому что тогда $(m+1)(p-m)$ может быть простым. Из неравенства $y \neq \varepsilon$ следует $m \geqslant 1$ и $m+1 > 1$. Кроме того, $m = |y| \leqslant n$, а $p \geqslant n+2$, поэтому $p - m \geqslant 2$.

Пришли к противоречию. Таким образом язык L_{pr} нерегулярен. □

6.4. Деревья вывода и однозначность грамматик типа 2

Рассмотрим наглядный метод описания любого вывода в грамматике типа 2.

Определение 96 (Дерево вывода). Пусть $G = (N, \Sigma, P, S)$ — грамматика типа 2. Упорядоченное дерево есть *дерево вывода* (*дерево разбора*) для G, если

1) каждый узел имеет метку — символ из $N \cup \Sigma \cup \{\varepsilon\}$;
2) метка корня — S;
3) если узел имеет по крайней мере одного потомка, то его метка этого узла — нетерминальный символ;
4) если узлы $n_1, n_2, \ldots, n_k$ — прямые потомки узла n, перечисленные слева направо, с метками $A_1, A_2, \ldots, A_k$ соответственно, а метка узла n есть A, то в множестве продукций P присутствует правило $A \to A_1A_2\ldots A_k$. Причем узел n_i может быть помечен ε только в случае если n_i — единственный потомок узла n, и в P присутствует правило $A \to \varepsilon$.

Пример 54. Рассмотрим грамматику $G = (\{S, A\}, \{a, b\}, P, S)$, где $P = \{S \to aAS, S \to a, A \to SbA, A \to ba, A \to SS\}$. На рис. 6.1 вывод цепочки $aabbaa$ в грамматике G представлен в виде дерева. □

Определение 97 (Крона дерева вывода). *Кроной* дерева вывода называют цепочку, составленную из меток всех листьев этого дерева, перечисленных слева направо.

В последнем примере цепочка $aabbaa$ является кроной приведенного дерева вывода.

Определение 98 (Сечение дерева вывода). *Сечением* дерева вывода называют набор узлов этого дерева такой, что:

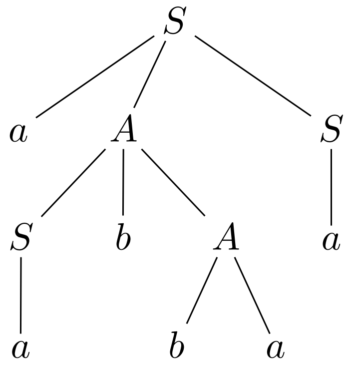


Рис. 6.1. Дерево вывода $S \Rightarrow aAS \Rightarrow aSbAS \Rightarrow aSbAa \Rightarrow aabAa \Rightarrow aabbaa$

1) никакие два узла в сечении не принадлежат одному и тому же пути от корня до листа;
2) нельзя добавить ни одного узла в сечение, чтобы не нарушилось правило 1.

В любом дереве множество состоящее только из корня дерева является сечением этого дерева. Также, множество всех листьев дерева является сечением.

В последнем примере сечениями являются наборы $\{S\}$, $\{a, a, b, b, a, a\}$, $\{a, A, S\}$, $\{a, A, a\}$.

Определение 99 (Крона сечения). *Кроной сечения* дерева вывода называют цепочку, составленную из меток всех узлов в сечении, перечисленных слева направо.

Перечисленным выше сечениям соответствуют кроны S, $aabbaa$, aAS, aAa.

Имеет место следующее утверждение.

Утверждение 6. *Пусть $G = (N, \Sigma, P, S)$ — КС-грамматика. Для цепочки $\alpha \in (N + \Sigma)^*$ существует вывод $S \Rightarrow_G^* \alpha$ тогда и только тогда, когда существует дерево вывода в грамматике G с кроной сечения α.*

Определение 100 (Неоднозначная грамматика). Контекстно-свободную грамматику $G = (N, \Sigma, P, S)$ называют *неоднозначной*, если найдется хотя бы одна выводимая цепочка $\alpha \in (N + \Sigma)^*$, для которой существуют два разных дерева вывода с кроной сечения α. В противном случае грамматику называют *однозначной*

Пример 55. Рассмотрим грамматику, которая представляет арифметические выражения типичного языка программирования (в несколько упрощенном виде). Ограничимся только операциями сложения и умножения, а в качестве аргументов возьмем только идентификаторы, составленные из букв a и b, цифр 0 и 1. Рассмотрим грамматику

$$G = (\{E, I\}, \{+, *, (,), a, b, 0, 1\}, P, E),$$

в которой множество P составляют следующие правила:

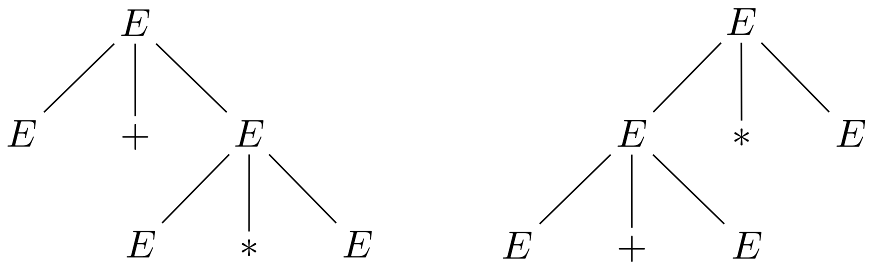
$$\begin{aligned} E &\to I \mid E+E \mid E*E \mid (E), \\ I &\to a \mid b \mid Ia \mid Ib \mid I0 \mid I1. \end{aligned} \tag{6.8}$$

Рассмотрим выводимую цепочку $E + E * E$. Для этой цепочки существует два различных вывода в грамматике G:

$$E \Rightarrow E + E \Rightarrow E + E * E, \tag{6.9}$$

$$E \Rightarrow E * E \Rightarrow E + E * E. \tag{6.10}$$

Здесь в первом выводе второе E заменяется на $E*E$, а во втором — первое E на $E + E$. На рис. 6.2 показаны два действительно различных дерева разбора.



(a) ($б$)

Рис. 6.2. Два дерева вывода с одной и той же кроной

Разница между этими двумя выводами значительна. Когда рассматривается структура выражений, первый вывод говорит, что второе и третье выражения перемножаются, и результат складывается с первым. Вместе с тем, второй вывод задает сложение первых двух выражений и умножение результата на третье. Более конкретно, первый вывод задает, что $1 + 2 * 3$ группируется как $1 + (2 * 3) = 7$, а второй — что группирование имеет вид $(1 + 2) * 3 = 9$. Очевидно, что первое из них (но не второе) соответствует нашему понятию о правильном группировании арифметических выражений.

Поскольку грамматика (6.8) дает две различные структуры любой цепочке терминалов, порождаемой заменой трех выражений в $E + E * E$ идентификаторами, для обеспечения уникальности структуры она не подходит. В частности, хотя она может давать цепочкам как арифметическим выражениям правильное группирование, она также дает им неправильное. Для того, чтобы использовать эту грамматику выражений в компиляторе, мы должны изменить ее, обеспечив правильное группирование. $\square$

С другой стороны, само по себе существование различных выводов одной и той-же цепочки еще не означает неоднозначности грамматик.

Пример 56. Для грамматики (6.8) рассмотрим два вывода цепочки $a + b$:

$$E \Rightarrow E + E \Rightarrow I + E \Rightarrow a + E \Rightarrow a + I \Rightarrow a + b,$$
$$E \Rightarrow E + E \Rightarrow E + I \Rightarrow I + I \Rightarrow I + b \Rightarrow a + b.$$

Заметим, что настоящей разницы между структурами, заданными этими двумя выводами, нет. В действительности, оба этих вывода приводят к одному и тому же дереву вывода. □

Не существует алгоритма, способного определить, является ли произвольная КС-грамматика неоднозначной.

На практике, для многих конструкций, возникающих в обычных языках программирования, существует техника устранения неоднозначности. Так проблема с грамматикой (6.8) типична, и мы исследуем устранение ее неоднозначности.

Есть следующие две причины неоднозначности в грамматике (6.8).

1. Не учитываются приоритеты операторов, описываемых грамматикой. Необходимо обеспечить, чтобы в однозначной грамматике была допустимой только структура, показанная на рис. 6.2, *а*.
2. Последовательность одинаковых операторов может группироваться как слева, так и справа. Поскольку операторы $+$ и $*$ ассоциативны, не имеет значения, группируем мы их слева или справа, но для исключения неоднозначности нам нужно выбрать что-то одно.

Решение проблемы установления приоритетов состоит в том, что вводится несколько разных нетерминалов, каждый из которых представляет выражение, имеющее один и тот же уровень «связывающей мощности». В частности, для грамматики (6.8) это решение может иметь следующий вид.

1. *Сомножитель*, или *фактор*, — это выражение, которое не может быть разделено на части никаким примыкающим оператором: ни $*$ ни $+$. Сомножителями в нашем языке выражений являются только следующие выражения:
   а) идентификаторы. Буквы идентификатора невозможно разделить путем присоединения оператора;
   б) выражения в скобках, независимо от того, что находится между ними.
2. *Слагаемое* или *терм*, — это выражение, которое не может быть разорвано оператором $+$. В нашем примере терм представляет собой произведение одного или нескольких сомножителей.
3. *Выражение* будет обозначать любое возможное выражение.

Следующая грамматика представляет собой тот же язык, что и грамматика (6.8), но является однозначной:

$$
\begin{aligned}
&E \to T \mid E + T,\\
&T \to F \mid T * F,\\
&F \to I \mid (E),\\
&I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1.
\end{aligned}
\tag{6.11}
$$

Так, например, для цепочки $a + a * a$ в этой грамматике существует только одно дерево разбора, показанное на рисунке 6.3.

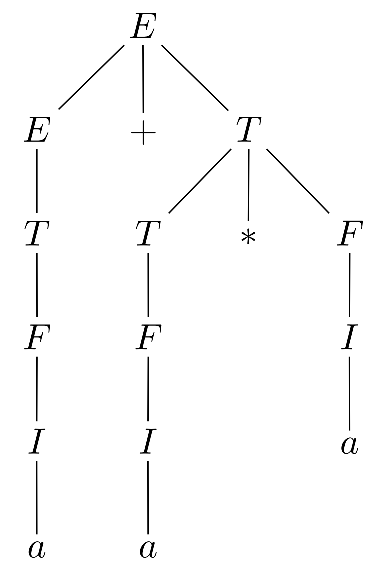


Рис. 6.3. Дерево вывода для цепочки $a + a * a$ и грамматики (6.11)

Определение 101 (Существенно неоднозначный язык). КС-язык L называется *существенно неоднозначным*, если все его грамматики неоднозначны. Если хотя бы одна грамматика языка L является однозначной, то L является однозначным языком.

Теорема 25 (Об однозначности языков типа 3)**.** *Все языки типа 3 однозначны.*

Доказательство. Пусть L — язык типа 3, над терминальным алфавитом Σ. По теореме 23 $L = L(M)$, для некоторого КДА

$$M = (Q, \Sigma, P, q_0, F).$$

Отсюда, язык L порождается грамматикой

$$G = (Q, \Sigma, P_1, q_0),$$

где

$$P_1 = \{q \to ar \mid q, r \in Q,\ a \in \Sigma,\ qa \to r \in P\} \cup \{q \to \varepsilon \mid q \in F\}.$$

Грамматика G, очевидно, является однозначной. □

6.5. Преобразования КС-грамматик

Определение 102 (ε-правило). В грамматике типа 2 *ε-правилом* называют любое правило вида $A \to \varepsilon$, где A — нетерминальный символ.

Определение 103 (A-правило). В грамматике типа 2 *A-правилом* называют любое правило вида $A \to \alpha$, где A — нетерминальный символ, α — произвольная цепочка.

Определение 104 (Цепное правило). В грамматике типа 2 *цепным правилом* называют любое правило вида $A \to B$, где A и B — нетерминальные символы.

Определение 105 (Грамматика без ε-правил). Грамматику типа 2 $G = (N, \Sigma, P, S)$ называют *грамматикой без ε правил*, если в ней отсутствуют ε-правила, или присутствует правило $S \to \varepsilon$, при условии, что S не встречается в правых частях правил.

Теорема 26 (О грамматиках без ε-правил)**.** *Для любого языка типа 2 найдется его определяющая грамматика без ε-правил.*

Доказательство. Пусть L — язык типа 2, а $G = (N, \Sigma, P, S)$ — КС-грамматика его определяющая.

Найдем множество

$$U = \{A \in N \mid A \Rightarrow^* \varepsilon\}. \tag{6.12}$$

Для этого определим

$$\begin{aligned} U_1 &= \{A \mid A \to \varepsilon \in P\}, \\ U_{i+1} &= U_i \cup \{A \mid A \to \alpha \in P,\ \alpha \in U_i^*\}, \quad i \geqslant 1. \end{aligned}$$

Так как U_i представляют собой монотонно увеличивающиеся подмножества множества N, найдется число k такое, что $U_k = U_{k+v}$, $v = 1, 2, \ldots$. Положим $U = U_k$.

Таким образом, $\varepsilon \in L(G)$ тогда и только тогда, когда $S \in U$.

Построим грамматику $G' = (N, \Sigma, P', S)$, в которой продукция $A \to \alpha$ принадлежит множеству P' тогда и только тогда, когда $\alpha \neq \varepsilon$ и найдется продукция $A \to \beta \in P$ такая, что α получается из β удалением 0 или более символов из U.

Справедливо, что $L(G') = L(G) - \{\varepsilon\}$. Действительно, включение $L(G') \subseteq L(G) - \{\varepsilon\}$ очевидно из (6.12). Обратное включение устанавливается при рассмотрении грамматики

$$G_1 = (N, \Sigma, P \cup P', S).$$

Ясно, что $L(G) \subseteq L(G_1)$. Если $w \in L(G)$ и $w \neq \varepsilon$, то $w \in L(G')$, т. к. все применения продукций $A \to \varepsilon$ могут быть заменены некоторыми применениями продукций из P'.

Если $\varepsilon \in L(G)$, построим грамматику G'':

$$G'' = (N \cup \{S'\}, \Sigma, P' \cup \{S' \to \varepsilon, S' \to S\}, S').$$

По построению $L(G'') = L(G') \cup \{\varepsilon\}$.

Как грамматика G', так и грамматика G'' — грамматики без ε-правил.

□

Следствие 26.1. *Любой язык типа 2 является языком типа 1.*

Пример 57. Рассмотрим грамматику

$$S \to AB, \qquad A \to aAA \mid \varepsilon, \qquad B \to bBB \mid \varepsilon.$$

Применим к ней метод, предложенный в доказательстве последней теоремы.

Сначала, определим, что $U = \{A, B, S\}$.

Правило $S \to AB$ влечет за собой включение в новое множество правил P' правил $S \to AB \mid A \mid B$. Из-за правила $A \to aAA$ включаем в P' правила $A \to aAA \mid aA \mid a$. Процесс повторяем для всех правил исходной грамматики, не являющихся ε-правилами. В итоге получим множество P' в следующем виде:

$$\begin{aligned} S &\to AB \mid A \mid B, \\ A &\to aAA \mid aA \mid a, \\ B &\to bBB \mid bB \mid b. \end{aligned}$$

Так как $S \in U$, то включаем новый начальный нетерминал S' в множество нетерминалов и добавляем два правила $S' \to S \mid \varepsilon$.

Окончательная грамматика:

$$\begin{aligned} S' &\to S \mid \varepsilon, \\ S &\to AB \mid A \mid B, \\ A &\to aAA \mid aA \mid a, \\ B &\to bBB \mid bB \mid b. \end{aligned}$$

□

Теорема 27 (О грамматиках без цепных правил)**.** *Для любого языка типа 2 найдется грамматика без цепных правил его определяющая.*

Доказательство. Пусть L — язык типа 2, а $G = (N, \Sigma, P, S)$ — грамматика его определяющая. Согласно теореме 26 можно предположить, что G — грамматика без ε-правил.

Для каждого $X \in N$ найдем

$$U(X) = \{B \in N \mid B \Rightarrow^* X\}. \tag{6.13}$$

Для этого определим

$$\begin{aligned} U_1(X) &= \{X\}, \\ U_{i+1}(X) &= U_i \cup \{A \mid A \to B \in P,\ B \in U_i(X)^*\}, \quad i \geqslant 1. \end{aligned}$$

Так как $U_i(X)$ представляют собой монотонно увеличивающиеся подмножества множества N, найдется число k такое, что $U_k = U_{k+v}$, $v = 1, 2, \ldots$. Положим $U(X) = U_k(X)$.

Построим грамматику $G' = (N, \Sigma, P', S)$, в которой продукция $A \to \beta$ принадлежит множеству P' тогда и только тогда, когда это не цепное правило и найдется нетерминал B такой, что существует правило $B \to \beta \in P$ и $A \in U(B)$.

Покажем, что $L(G) \subseteq L(G')$. Действительно, рассмотрим цепочку $w \in L(G)$. Построим дерево вывода для этой цепочки по грамматике G. Заметим, что по принципу построения грамматики G' все нецепные правила грамматики G включаются в P'. Применение цепного правила $A \to B$ при выводе цепочки w выльется в появление в дереве вывода узла с меткой A, имеющего единственного потомка с меткой B. Наличие в дереве вывода цепи $A_0, A_1, \ldots, A_k$ для $k \geqslant 1$ соответствовует последовательности применений цепных правил $A_0 \to A_1, \ldots, A_{k-1} \to A_k$. Пусть A_0 — начальный, а A_k — конечный узел такой цепи, т. е. узел A_0 появился не в результате применения цепного правила, и к нетерминалу A_k применяется не цепное правило. Допустим, что к нетерминалу A_k применяется правило $A_k \to \alpha$, $\alpha \in (N + \Sigma)^*$. Согласно 6.13 $A_0 \in U(A_k)$. Принимая это во внимание и в связи с тем, что $A_k \to \alpha \in P$, можем сделать вывод, что правило $A_0 \to \alpha$ должно содержаться в P'. Заменим поддерево A_0-$\cdots$-A_k-α на A_0-α. Такая замена будет соответствовать применению правила грамматики G'. Крона дерева от такой замены не изменится. Повторим этот процесс для каждой цепи меток-нетерминалов в нашем дереве. Получим дерево, построенное по правилам грамматики G' с той же кроной. Так как кроной дерева является цепочка w, делаем вывод, что любое слово языка $L(G)$ содержится в $L(G')$.

В обратную сторону. Рассмотрим вывод цепочки $w \in L(G')$. Пусть шаг вывода $\alpha_i \Rightarrow_{G'} \alpha_{i+1}$ происходит в результате применения правила $A \to \beta \in P'$. Но наличие такого правила в P' предполагает, что $A \in U(B)$ для некоторого нетерминального символа B (возможно $B = A$) и в грамматике G присутствует правило $B \to \beta$. Факт $A \in U(B)$ означает, что $A \Rightarrow_G B$. Следовательно $\alpha_i \Rightarrow^*_G \alpha_{i+1}$. Так как шаг вывода был выбран произвольно, $S \Rightarrow^*_G w$. В силу произвольного выбора цепочки w $L(G') \subseteq L(G)$.

Таким образом $L(G') = L(G)$. □

Определение 106 (Нормальная форма Хомского). КС-грамматика $G = (N, \Sigma, P, S)$ называется *грамматикой в нормальной форме Хомского*, если в ней нет ε-правил и любое правило, отличное от $S \to \varepsilon$ имеет вид:

1. $A \to BC$, где $B, C \in N$,
2. $A \to a$, где $a \in \Sigma$.

Теорема 28 (О грамматиках в нормальной форме Хомского)**.** *Для любого языка типа 2 найдется грамматика в нормальной форме Хомского.*

Доказательство. Пусть L — язык типа 2, а $G = (N, \Sigma, P, S)$ — грамматика его определяющая. Согласно теореме 26 и теореме 27 можно, не теряя общности предположить, что G — грамматика без ε-правил и без цепных правил.

Тогда P может состоять из продукций вида

$$S \to \varepsilon, \tag{6.14}$$

$$A \to a, \quad a \in \Sigma, \tag{6.15}$$

$$A \to X_1X_2, \quad X_1, X_2 \in (N + \Sigma), \tag{6.16}$$

$$A \to X_1X_2 \dots X_k, \quad k > 2,\ X_1, \dots, X_k \in (N + \Sigma). \tag{6.17}$$

Построим грамматику $G' = (N', \Sigma, P', S)$, в которой в множество P' включены

1) все правила из P вида (6.14) и (6.15);
2) для каждого правила из P вида (6.16) — правила вида

$$A \to X'_1X'_2,$$

где $X'_i = X_i$, если $X_i \in N$ и X'_i — новый нетерминал в противном случае;
3) для каждого правила из P вида (6.17) — правила вида

$$A \to X'_1Y_1, \qquad Y_1 \to X'_2Y_2, \qquad \dots, \qquad Y_{k-2} \to X'_{k-1}X'_k,$$

где все Y_i — новые нетерминалы, а $X'_i = X_i$, если $X_i \in N$ и X'_i — новый нетерминал в противном случае;

4) правила $X'_i \to X_i$, для всех новых нетерминалов X'_i, введенных в пунктах 2–3.

Легко проверить, что $L(G') = L(G)$. □

6.6. Лемма о накачке для КС-языков

Лемма 3 (О длине выводимой цепочки)**.** *Пусть дано дерево разбора для грамматики $G = (N, \Sigma, P, S)$ в нормальной форме Хомского. Пусть кроной дерева является цепочка $\alpha \in \Sigma^+$ (Рис. 6.4).*

Если n — наибольшая длина пути от корня к листьям, то $|\alpha| \leqslant 2^{n-1}$.

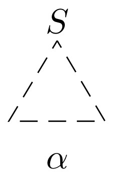


Рис. 6.4. Дерево вывода цепочки α по грамматике G

Доказательство. По индукции: Пусть $n = 1$. Длина пути есть число ребер, т. е. на 1 меньше числа узлов, участвующих в этом пути. Таким образом, дерево с максимальной длиной пути состоит из корня и листа, отмеченного терминалом (Рис. 6.5). Цепочка α является этим терминалом, и $|\alpha| = 1$. Так как $2^{n-1} = 2^0 = 1$, то предположение леммы для $n = 1$ верно. Пусть $n > 1$. Корень дерева использует правило, которое должно

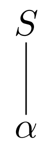


Рис. 6.5. Шаг индукции $n = 1$

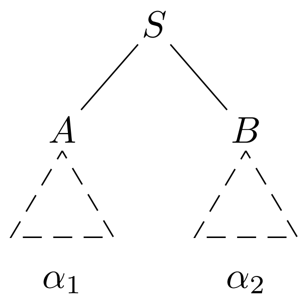


Рис. 6.6. Шаг индукции $n > 1$

иметь вид $S \to AB$ (Рис. 6.6), т. к. $n > 1$ (т. е. нельзя начать дерево, использовав продукцию с терминалом). Ни один из путей в поддеревьях с корнями в A и B не может иметь длину больше, чем $n-1$. По предположению индукции эти два поддерева имеют кроны длины не более 2^{n-2}. Крона всего дерева представляет собой конкатенацию этих двух крон, а следовательно имеет длину не превышающую $2^{n-2}+2^{n-2}=2^{n-1}$. $\square$

Теорема 29 (Лемма о накачке (о разрастании) для КС-языков)**.** *Пусть L — контекстно-свободный язык. Тогда существует такая константа n, что если z — произвольная цепочка из L, длина которой не меньше n, то можно записать $z = uvwxy$, причем выполняются следующий условия.*

1. $|vwx| \leqslant n$ *(средняя часть не слишком длинная);*
2. $vx \neq \varepsilon$ *(хотя бы одна из цепочек v и x не пуста);*
3. $uv^iwx^iy \in L, \quad \forall i \geqslant 0.$

Доказательство. Для L найдем грамматику $G = (N, \Sigma, P, S)$ в нормальной форме Хомского. Пусть $\#N = m$.

Возьмем $n = 2^m$. Предположим, что $z \in L$, $|z| \geqslant n$. По лемме 3 любое дерево разбора с кроной z имеет путь длинной не менее $m+1$.

Пусть самый длинный путь имеет длину $k+1$, $k \geqslant m$. Тогда на этом пути встречается $k+1$ нетерминал $A_0, A_1, A_2, \ldots, A_k$. Но N содержит всего m нетерминалов, поэтому хотя бы два из $m+1$-го нетерминала на пути должны совпадать. Пусть $A_i = A_j = A$, где $k-m \leqslant i \leqslant j \leqslant k$.

Тогда дерево разбора можно разделить так, как показано на рис. 6.7. Цепочка w является кроной поддерева с корнем A_j. Цепочки v и x — это цепочки соответственно справа и слева от w в кроне большого поддерева с корнем A_i. Цепных правил в грамматике нет, а поэтому v и x не могут одновременно быть ε (хотя одна из них может). Наконец u и yобразуют части z, лежащие соответственно слева и справа от поддерева с корнем A_i.

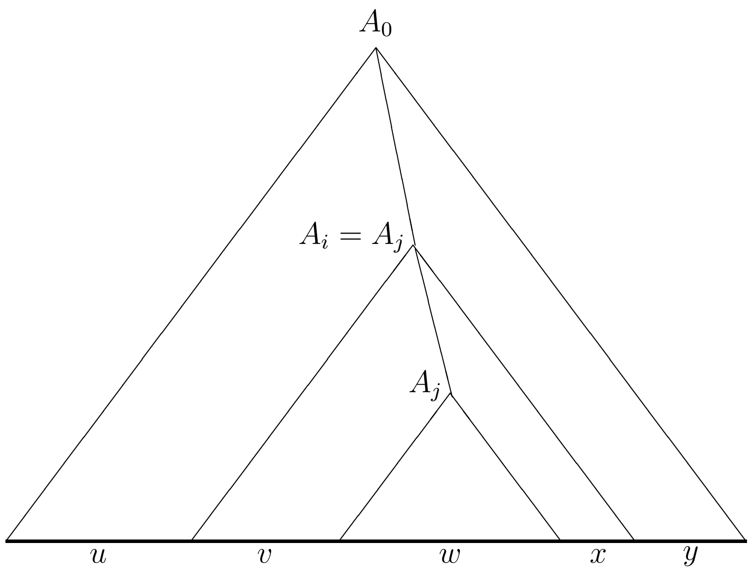


Рис. 6.7. Дерево разбора для исходной цепочки

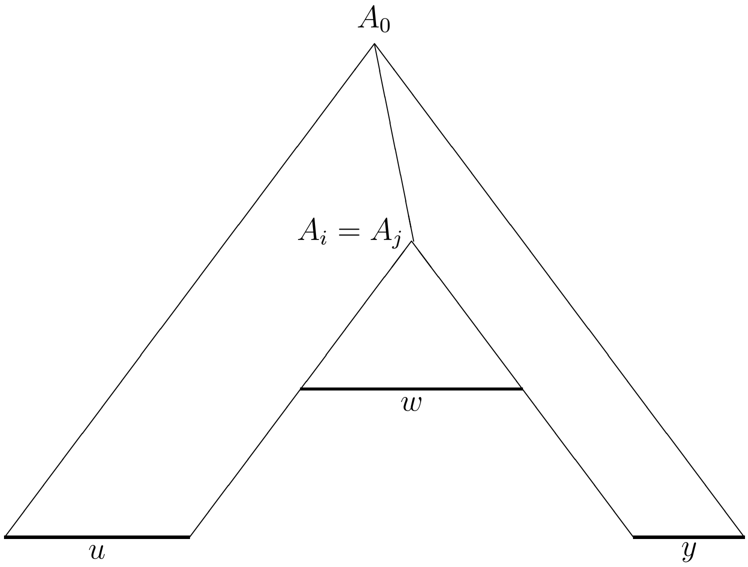


Рис. 6.8. Накачивание цепочек v и x 0 раз

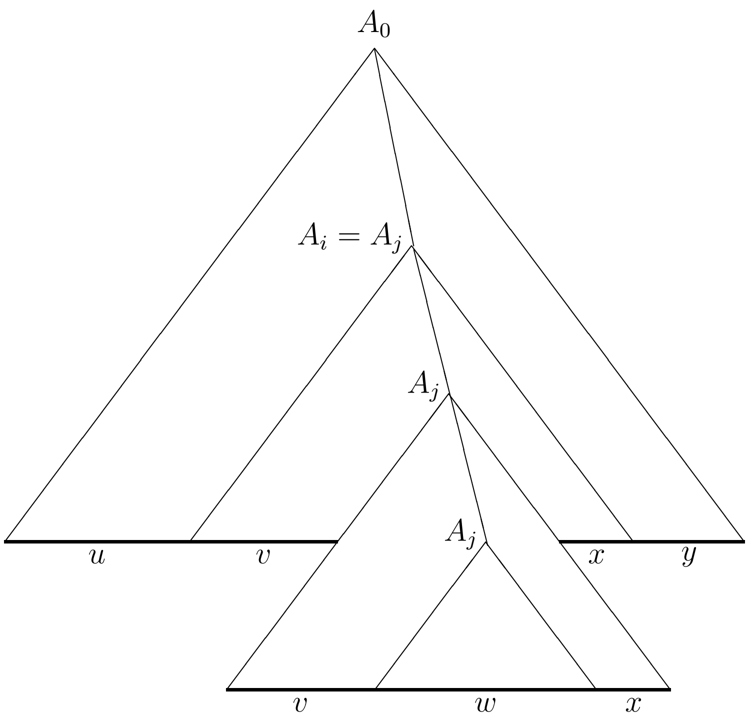


Рис. 6.9. Накачивание цепочек v и x 2 раза

Если $A_i = A_j = A$, то по исходному дереву можно построить новое дерево разбора. Сначала можно заменить поддерево с корнем A_j, имеющее крону vwx, поддеревом с корнем A_j, у которого крона w. Это допустимо, т. к. корни обоих поддеревьев отмечены одним и тем же нетерминалом A. Полученное дерево представлено на рис. 6.8. Оно соответствует случаю $i = 0$.

Другая возможность представлена на рис. 6.9. Там поддерево с корнем

A_j заменено поддеревом с корнем A_i. Это допустимо, т. к. корни поддеревьев совпадают. Кроной такого дерева является цепочка uv^2wx^2y. Можно далее продолжать процесс замены поддеревьев большими поддеревьями с целью получения кроны uv^iwx^iy для любого i. Итак в G существуют деревья разбора для всех таких цепочек.

Мы выбирали A_i с условием $k-i \leqslant m$. Таким образом самый длинный путь в поддереве с корнем A_i имеет длину не более чем $m+1$. По лемме 3 поддерево с корнем A_i имеет крону, длина которой не больше, чем $2^m = n$. Тем самым все утверждения теоремы доказаны. □

Контрольные вопросы

1. Что из себя представляет регулярное выражение?
2. Как регулярное выражение определяет язык?
3. Перечислите операции над регулярными выражениями в порядке возрастания приоритетов.
4. Что является решением леволинейной системы с регулярными коэффициентами?
5. Сформулируйте алгоритм решения системы уравнений с регулярными коэффициентами?
6. Какой класс языков определяется регулярными выражениями?
7. Как получить регулярное выражение для языка определенного грамматикой типа 3?
8. Как показать, что класс языков типа 3 замкнут относительно операции зеркального отражения?
9. Продемонстрируйте, что класс языков типа 3 замкнут относительно подстановки.
10. Приведите пример самовставленной грамматики.
11. Приведите пример несамовставленной грамматики типа 2, не являющейся леволинейной или праволинейной грамматикой.
12. Как можно доказать нерегулярность языка?
13. Какой язык называется однозначным?
14. Какое правило принято называть A-правилом?
15. Какую грамматику принято называть грамматикой без ε-правил?
16. Какое правило грамматики называется цепным?
17. Сформулируйте алгоритм приведения КС-грамматике к форме без цепных правил.
18. В чем примечательность грамматики в нормальной форме Хомского?
19. Какой основной прием применяется при приведении грамматики к нормальной форме Хомского?
20. Какие признаки могут указывать, что язык не является контекстно-свободным?