МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский Нижегородский государственный
университет им. Н.И. Лобачевского»

Л.П. Жильцова, Т.Г. Смирнова

ОСНОВЫ ТЕОРИИ АВТОМАТОВ И ФОРМАЛЬНЫХ ЯЗЫКОВ В ПРИМЕРАХ И ЗАДАЧАХ

Учебно-методическое пособие

Рекомендовано методической комиссией
института информационных технологий, математики и механики
для студентов ННГУ, обучающихся по направлениям подготовки
020302 «Фундаментальная информатика и информационные технологии»,
090303 «Прикладная информатика (в информационной сфере)»,
090304 «Программная инженерия»

Нижний Новгород
2017

УДК 519.17+519.711.4
ББК В182
Ж-72

Ж−72 Жильцова Л.П., Смирнова Т.Г. Основы теории автоматов и формальных языков в примерах и задачах: учебно-методическое пособие. – [электронный ресурс] – Нижний Новгород: Нижегородский госуниверситет, 2017. – 64 с.

Фонд электронных образовательных ресурсов ННГУ

Рецензент: к. ф.-м. наук, доцент **А.Л.Калашников**

В учебно-методическом пособии рассматриваются основные понятия и алгоритмы теории автоматов и формальных языков. Особое внимание уделяется разделам, относящимся к теории конечных автоматов и регулярных языков. Изучение каждой темы сопровождается необходимым теоретическим материалом и примерами решения типовых задач. Предлагаются также задачи для самостоятельного решения.

Учебно-методическое пособие предназначено для студентов института информационных технологий, математики и механики направлений подготовки «Фундаментальная информатика и информационные технологии», «Прикладная информатика (в информационной сфере)», «Программная инженерия», изучающих курсы «Теория автоматов и формальных языков» и «Дискретная математика».

УДК 519.17+519.711.4
ББК В182

Глава 1. Формальные языки и грамматики

1.1. Языки и основные операции над языками

Алфавит – это любое множество символов (букв). Будем рассматривать конечные алфавиты.

Примеры алфавитов: русский алфавит, двоичный алфавит $\{0,1\}$.

Словом (или ***цепочкой***) в алфавите A называется конечная последовательность символов множества A. Например, $\alpha = 011000101110$ – слово в алфавите $\{0,1\}$.

Длина слова – количество символов в нем. Через $|\alpha|$ будем обозначать длину слова α, а через $|\alpha|_a$ – число вхождений символа a в α. Например, длина слова $\alpha = 011000101110$ равна 12, тогда $|\alpha| = 12$, причем $|\alpha|_0 = 6$ и $|\alpha|_1 = 6$.

Пустое слово – слово, не содержащее ни одного символа. Будем обозначать его через λ. Длина пустого слова равна 0.

Конкатенация (или ***сцепление***) слов x и y – приписывание слова y в конец слова x. Например, $x = ab$, $y = cd$, $xy = abcd$.

Для любого слова x всегда $x\lambda = \lambda x = x$.

Обращение слова – слово, записанное в обратном порядке. Например, для слова $x = abcd$ обращение равно $dcba$. Обращение слова x будем обозначать x^R. Таким образом, $x^R = dcba$.

Язык в алфавите A – произвольное множество слов в алфавите A. Например, $L = \{a, ab, ca\}$ – конечный язык в алфавите $A = \{a, b, c\}$, содержащий три слова, $L = \{0, 00, 000, ...\}$ – бесконечный язык в алфавите $\{0\}$, $\varnothing$ – пустой язык, $\{\lambda\}$ – язык, состоящий из одного пустого слова. Отметим, что $\varnothing$ и $\{\lambda\}$ – разные языки.

A^* – множество, содержащее все слова в алфавите A, включая пустое слово. Если $A = \{0,1\}$, то $A^* = \{\lambda, 0, 1, 00, 01, 10, 11, 000, \ldots\}$. Каждый язык L в алфавите A является подмножеством A^*, т.е. $L \subseteq A^*$.

A^+ – множество всех непустых слов в алфавите A.

Пусть L_1 и L_2 – языки в алфавите A, тогда для них определены теоретико-множественные операции:

$L_1 \cup L_2 = \{x \mid x \in L_1$ или $x \in L_2\}$ – ***объединение*** языков L_1 и L_2;

$L_1 \cap L_2 = \{x \mid x \in L_1$ и $x \in L_2\}$ – ***пересечение*** языков L_1 и L_2;

$L_1 - L_2 = \{x \mid x \in L_1 \text{ и } x \notin L_2\}$ – ***разность*** языков L_1 и L_2;

$\overline{L} = A^* - L = \left\{ x \mid x \in A^* \text{ и } x \notin L \right\}$ – ***дополнение*** языка L.

Пусть L_1 – язык в алфавите A_1, L_2 – язык в алфавите A_2, тогда язык $L_1L_2 = \{xy \mid x \in L_1, y \in L_2\}$ называется ***конкатенацией*** (или ***сцеплением***, ***произведением***) языков L_1 и L_2.

$L^0 = \{\lambda\}$ – ***нулевая степень*** языка L;

$L^n = LL^{n-1}$, где $n \ge 1$, – n***-ая степень*** языка L;

$L^* = \bigcup\limits_{n \ge 0} L^n$ – ***итерация*** (***звездочка*** или ***замыкание Клини***) языка L.

Позитивная итерация языка L, обозначаемая L^+, – это язык $\bigcup\limits_{n \ge 1} L^n$.

$L^+ = LL^* = L^*L$, и $L^* = L^+ \cup \{\lambda\}$.

Задачи и упражнения

1.1.1. Может ли язык L^* или L^+ быть пустым?

1.1.2. При каких условиях L^* и L^+ конечны?

1.1.3. Верно ли, что $L^+ = L^* - \{\lambda\}$?

1.1.4. Для заданных языков L_1 и L_2 найти $L_1 \cup L_2$ и $L_1 \cap L_2$:

а) $L_1 = \left\{ b^i a \mid i \ge 0 \right\}$, $L_2 = \left\{ a^i b \mid i \ge 0 \right\}$;

б) $L_1 = \left\{ b^i a^j \mid i \ge 0, j \ge 0 \right\}$, $L_2 = \left\{ a^i b^j \mid i \ge 0, j \ge 0 \right\}$;

в) $L_1 = \left\{ \alpha \mid \alpha \in \{0,1\}^*, |\alpha|_0 = |\alpha|_1 \right\}$, $L_2 = \left\{ 0^i 1^j \mid i \ge 0, j \ge 0 \right\}$;

г) $L_1 = \{ \alpha \mid \alpha \in \{0,1\}^*, |\alpha|_1 \le 2$, и если содержит 1, то и начинается с 1 $\}$,
$L_2 = \left\{ \beta\beta^R \,\middle|\, \beta \in \{0,1\}^* \right\}$;

д) $L_1 = \left\{ \alpha \mid \alpha = a^{2i}, i \ge 0 \right\}$, $L_2 = \left\{ \beta\beta \mid \beta \in \{a,b\}^* \right\}$.

1.1.5. Для заданных языков L_1 и L_2 найти $L_1 - L_2$:

а) $L_1 = \left\{ a^i \mid i \ge 0 \right\}$, $L_2 = \left\{ b^i a \mid i \ge 0 \right\}$;

б) $L_1 = \left\{ b^i a \mid i \ge 0 \right\}$, $L_2 = \left\{ a^i \mid i \ge 0 \right\}$;

в) $L_1 = \left\{ a^i b^j \mid i \ge 0, j \ge 0 \right\}$, $L_2 = \left\{ b^i a^j \mid i \ge 0, j \ge 0 \right\}$;

г) $L_1 = \left\{ b^i a^j \mid i \ge 0, j \ge 0 \right\}$, $L_2 = \left\{ a^i b^j \mid i \ge 0, j \ge 0 \right\}$;

д) $L_1 = \{a, b\}^*$, $L_2 = \left\{ a^i \mid i \ge 0 \right\}$.

1.1.6. Для заданного языка L в алфавите $A = \{a, b\}$ найти $\overline{L}$:

а) $L = \{a^i \mid i \ge 0\}$;

б) $L = \{a^i b a^j b a^k \mid i, j, k \ge 0\}$;

в) $L = \{b a^i b \mid i \ge 0\}$;

г) $L = \{\alpha \mid |\alpha|_a = 2k + 1,\ k \ge 1\}$;

д) $L = \left\{ \alpha \mid |\alpha|_a = 2i \text{ и } |\alpha|_b = 2j,\ i \ge 1,\ j \ge 0 \right\}$.

1.1.7. Для заданных языков L_1 и L_2 найти конкатенацию $L_1 L_2$ и $L_2 L_1$:

а) $L_1 = \left\{ 01, 001, 0001 \right\}$, $L_2 = \left\{ 10, 100 \right\}$;

б) $L_1 = \left\{ a^i \mid i \ge 1 \right\}$, $L_2 = \left\{ a^i b \mid i \ge 1 \right\}$;

в) $L_1 = \left\{ a^i \mid i \ge 1 \right\}$, $L_2 = \left\{ b^i a \mid i \ge 0 \right\}$;

г) $L_1 = \left\{ a^i \mid i \ge 1 \right\}$, $L_2 = \{a, b\}^*$;

д) $L_1 = \left\{ a^i \mid i \ge 0 \right\}$, $L_2 = \{a, b\}^*$;

е) $L_1 = \{a, b\}^*$, $L_2 = \left\{ a^i \mid i \ge 1 \right\}$;

ж) $L_1 = \{a, b\}^*$, $L_2 = \left\{ a^i \mid i \ge 0 \right\}$.

1.1.8. Для заданного языка L найти L^*:

а) $L = \left\{ a \right\}$;

б) $L = \left\{ a^i \mid i \ge 0 \right\}$;

в) $L = \left\{ a^i \mid i \ge 1 \right\}$;

г) $L = \left\{ a^i b^j \mid i \ge 0, j \ge 0 \right\}$;

д) $L = \left\{ a^i b^j \mid i \ge 0, j \ge 1 \right\}$;

е) $L = \left\{ a^i b^j \mid i \ge 1, j \ge 0 \right\}$;

ж) $L = \left\{ a^i b^j \mid i \ge 1, j \ge 1 \right\}$.

1.1.9. В алфавите $A = \{0, 1\}$ заданы языки $L_1 = \{0, 1, 00, 11, 000, 111, ...\}$, $L_2 = \{0,1\}^*$, $L_3 = \{w \mid w \in A^*, |w|_1 = 1\}$, $L_4 = \{w \mid w \in A^*, |w|_1 = 2\}$ и $L_5 = \{w \mid w \in A^*, |w|_1 \geq 2\}$. Найти:

а) $L_1 \cup L_2$, $L_3 \cup L_4$, $L_3 \cup L_5$, $L_4 \cup L_5$;

б) $L_1 \cap L_2$, $L_1 \cap L_3$, $L_1 \cap L_4$, $L_1 \cap L_5$, $L_3 \cap L_4$;

в) $L_1 - L_2$; $L_1 - L_3$, $L_3 - L_4$, $L_4 - L_5$, $L_5 - L_4$;

г) $\overline{L_1}$, $\overline{L_2}$, $\overline{L_3}$, $\overline{L_5 - L_4}$;

д) $L_1 L_2$, $L_3 L_4$, $L_4 L_3$;

е) L_2^*, L_3^*, L_4^*.

1.2. Иерархия Хомского

Грамматики образуют важный класс генераторов языков.

Грамматикой называется система $G = \langle V_T, V_N, P, S \rangle$, где V_T, V_N – непересекающиеся конечные множества терминальных и нетерминальных символов (терминалов и нетерминалов) соответственно; $S \in V_N$ – некоторый выделенный символ, называемый ***аксиомой грамматики***, P – конечное множество правил. Терминальные символы называются также основными, а нетерминальные – вспомогательными символами.

Каждое правило в P представляет собой пару слов (α, β), где $\alpha \in (V_T \cup V_N)^*$ и содержит хотя бы один нетерминальный символ, и $\beta \in (V_T \cup V_N)^*$. Правило (α, β) будем записывать в виде $\alpha \to \beta$.

Для слов α и β из $(V_T \cup V_N)^*$ будем говорить, что β ***непосредственно выводимо*** из α (и записывать $\alpha \Rightarrow \beta$), если $\alpha = \alpha_1 \delta \alpha_2$, $\beta = \alpha_1 \gamma \alpha_2$ для некоторых $\alpha_1, \alpha_2 \in (V_T \cup V_N)^*$, и в грамматике G имеется правило $\delta \to \gamma$.

Рефлексивное транзитивное замыкание отношения $\Rightarrow$ назовем ***отношением выводимости*** и обозначим через $\Rightarrow *$.

Язык, порождаемый грамматикой G, определим как множество всех слов в терминальном алфавите, выводимых из аксиомы грамматики S: $L(G) = \{\alpha : S \Rightarrow * \alpha, \alpha \in V_T^*\}$.

Пример 1.2.1. Рассмотрим грамматику $G = \langle \{0,1\}, \{S, A\}, P, S \rangle$, где P состоит из правил:

$S \to 0A1$

$0A \to 00A1$

$A \to \lambda$ (λ – пустое слово).

Нетерминальными символами являются S и A, терминальными – 0 и 1.

Покажем, что слово $\alpha = 0011$ принадлежит $L(G)$. Для этого построим вывод α из аксиомы S: $S \Rightarrow 0A1 \Rightarrow 00A11 \Rightarrow 0011$. Поэтому $S \Rightarrow {}^*\alpha$. Поясним, что на первом шаге к аксиоме S было применено правило $S \to 0A1$ и получено слово $0A1$; на втором шаге к подслову $0A$, входящему в слово $0A1$, применено правило $0A \to 00A1$, и на третьем шаге – правило $A \to \lambda$. В результате описанного вывода получено слово $0011 \in V_T^*$, поэтому $\alpha \in L(G)$. Можно показать, что $L(G) = \{0^n 1^n \mid n \ge 1\}$.

Для обозначения n правил с одинаковой левой частью

$$\alpha \to \beta_1$$
$$\alpha \to \beta_2$$
$$\ldots$$
$$\alpha \to \beta_n$$

применяют сокращенную запись $\alpha \to \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n$.

Грамматики можно классифицировать по виду их правил. Пусть $G = \langle V_T, V_N, P, S \rangle$ – грамматика.

Грамматика G называется

1) ***праволинейной***, если каждое правило из P имеет вид $A \to xB$ или $A \to x$, где $A, B \in V_N$, $x \in V_T^*$;
2) ***контекстно-свободной (или бесконтекстной)***, если каждое правило из P имеет вид $A \to \alpha$, где $A \in V_N$, $\alpha \in (V_N \cup V_T)^*$;
3) ***контекстно-зависимой (или неукорачивающей)***, если каждое правило из P имеет вид $\alpha \to \beta$, где $|\alpha| \le |\beta|$.
4) грамматика, не удовлетворяющая ни одному из указанных ограничений, называется ***грамматикой общего вида*** (или без ограничений).

Определенные выше четыре класса грамматик и соответствующих им языков называют ***иерархией Хомского.***

Пример 1.2.2. Грамматика $G_1 = \langle \{0,1\}, \{S\}, P, S \rangle$, где P состоит из правил: $S \to 0S \mid 1S \mid \lambda$, – праволинейная. Она порождает язык $\{0,1\}^*$.

Пример 1.2.3. Пусть $G_2 = \langle \{ a, +, *, (,) \}, \{E, T, F\}, P, E \rangle$, где P состоит из правил

$$E \to E + T \mid T$$
$$T \to T * F \mid F$$
$$F \to (E) \mid a.$$

Грамматика G_2 – контекстно-свободная. Пример вывода в этой грамматике:

$$E \Rightarrow E + T \Rightarrow T + T \Rightarrow F + T \Rightarrow a + T \Rightarrow a + T * F \Rightarrow a + F * F \Rightarrow$$
$$\Rightarrow a + a * F \Rightarrow a + a * a.$$

Язык $L(G_2)$ представляет собой множество арифметических выражений, построенных из символов $a, +, *,$ (и).

Пример 1.2.4. Пусть $G_3 = \langle \{a, b, c\}, \{S, B, C\}, P, S \rangle$, где P состоит из правил

$$S \to aSBC \mid abC$$
$$CB \to BC$$
$$bB \to bb$$
$$bC \to bc$$
$$cC \to cc.$$

Грамматика G_3 – контекстно-зависимая. Пример вывода в этой грамматике:
$S \Rightarrow aSBC \Rightarrow aabCBC \Rightarrow aabBCC \Rightarrow aabbCC \Rightarrow aabbcC \Rightarrow aabbcc.$

G_3 порождает язык $\{ a^n b^n c^n \mid n \geq 1 \}$.

Пример 1.2.5. Пусть $G_4 = \langle \{a, b\}, \{S, A, B, C, D\}, P, S \rangle$, где P состоит из правил

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

G_4 – грамматика общего вида. Пример вывода в этой грамматике:

$$S \Rightarrow CD \Rightarrow aCAD \Rightarrow abCBAD \Rightarrow abBAD \Rightarrow abBaD \Rightarrow$$
$$\Rightarrow abaBD \Rightarrow ababD \Rightarrow abab.$$

Грамматика G_4 порождает язык $\{ \omega\omega \mid \omega \in \{a, b\}^* \}$.

Задачи и упражнения

1.2.1. Пусть $L \subseteq \{a,b,c\}^*$. Построить праволинейные грамматики для языка L, если каждое слово из L:

а) начинается на букву c;
б) заканчивается на букву c;
в) содержит букву c;
г) содержит ровно одну букву c;
д) содержит не более одной буквы c;
е) содержит не менее двух букв c;
ж) содержит не более двух букв c;
з) содержит ровно две буквы c;
и) содержит подслово cc;
к) содержит подслово abc;
л) начинается с подслова abc;
м) заканчивается подсловом abc.

1.2.2. Построить контекстно-свободные грамматики для следующих языков:

а) $L = \{0^i 1^j \mid i = j\}$;
б) $L = \{0^i 1^j \mid i > j\}$;
в) $L = \{0^i 1^j \mid i \geq j\}$;
г) $L = \{0^i 1^j \mid i < j\}$;
д) $L = \{0^i 1^j \mid i \leq j\}$.

1.2.3. Пусть $L = \left\{ \alpha\alpha^R \,\middle|\, \alpha \in \{0,1\}^* \right\}$. Построить контекстно-свободную грамматику, порождающую язык L.

1.2.4. Пусть $L \subseteq \{a,b\}^*$ и каждое слово из L содержит одинаковое число букв a и b. Построить контекстно-свободную грамматику, порождающую L.

1.2.5. Пусть L − язык выражений из правильно расставленных скобок. Построить контекстно-свободную грамматику, порождающую язык L.

Пояснение. В выражении из правильно расставленных скобок для любого префикса выражения число открывающих скобок должно быть не меньше числа закрывающих скобок, а для выражения в целом число открывающих скобок равно числу закрывающих скобок. Например, (( ) (( ) ( ))) ( ) − выражение из правильно расставленных скобок.

1.2.6. Построить контекстно-зависимую грамматику для языка $L = \{ a^m b^n a^m b^n \mid m,n \geq 1\}$.

1.2.7. Пусть L – язык идентификаторов, которые могут содержать буквы латинского алфавита i, j, k, l, m, n и цифры 0 и 1, могут быть любой длины, но должны начинаться с буквы. Построить порождающую грамматику для L.

1.2.8. Пусть L – язык идентификаторов, которые могут содержать буквы латинского алфавита и цифры 0 и 1, должны содержать от одного до шести символов и начинаться с букв i, j, k, l, m, n. Построить порождающую грамматику для языка L.

1.2.9. Пусть L – язык вещественных констант с фиксированной точкой (примеры констант: 2, 2., 10.8, 3.14159). Построить порождающую грамматику для языка L.

1.2.10. Пусть L – язык вещественных констант с плавающей точкой (пример константы: 6.625E-27). Построить порождающую грамматику для L.

1.2.11. Описать язык, порождаемый грамматикой с правилами $S \to bSS \mid a$.

1.2.12. Описать язык, порождаемый грамматикой $G = \langle \{a, b\}, \{S, A\}, P, S \rangle$, где P состоит из правил:

$$\begin{aligned} &S \to aS \mid bS \mid bA \\ &A \to a. \end{aligned}$$

1.2.13. Описать язык, порождаемый грамматикой $G = \langle \{a, b\}, \{S, A\}, P, S \rangle$, где P состоит из правил:

$$\begin{aligned} &S \to aS \mid bS \mid aA \\ &A \to aA \mid bA \mid \lambda. \end{aligned}$$

1.2.14. Описать язык, порождаемый грамматикой $G = \langle \{a, b\}, \{S, A, B\}, P, S \rangle$, где P состоит из правил:

$$\begin{aligned} &S \to aB \\ &A \to aB \\ &B \to bA \mid b. \end{aligned}$$

1.2.15. Описать язык, порождаемый грамматикой $G = \langle \{a, b\}, \{S, A, B\}, P, S \rangle$, где P состоит из правил:

$$\begin{aligned} &S \to AB \\ &A \to a \\ &B \to Bb \mid \lambda. \end{aligned}$$

1.2.16. Описать язык, порождаемый грамматикой $G = \langle \{a, b\}, \{S, A, B\}, P, S \rangle$, где P состоит из правил:

$$\begin{aligned} &S \to AB \\ &A \to aA \mid \lambda \\ &B \to Bb \mid \lambda. \end{aligned}$$

Глава 2. Конечные автоматы и регулярные языки

2.1. Конечные автоматы и конечно-автоматные языки

Недетерминированный конечный автомат – это система $M = \langle Q, A, q_0, \delta, F \rangle$, где

Q – конечное множество состояний автомата,

A – конечный алфавит,

$q_0 \in Q$ – начальное состояние автомата,

$\delta : Q \times A \to 2^Q$ – функция переходов конечного автомата,

$F \subseteq Q$ – множество заключительных ("хороших") состояний автомата.

Автомат M называется ***детерминированным***, если множество $\delta(q, a)$ содержит не более одного состояния для любых $q \in Q$ и $a \in A$. Если $\delta(q, a)$ всегда содержит точно одно состояние, то автомат M называется ***полностью определенным***.

Конфигурация автомата M – это пара $(q, w) \in Q \times A^*$. Конфигурация (q_0, w) называется ***начальной***, а пара (q, λ), где $q \in F$, называется ***заключительной*** (или ***допускающей***).

Определим на множестве всех конфигураций конечного автомата M бинарное отношение $\vdash$ (***такт работы*** автомата M) следующим образом. Если $\delta(q, a)$ содержит q', тогда $(q, aw) \vdash (q', w)$ для всех $w \in A^*$.

Запись $C \vdash^0 C'$ означает, что $C = C'$, а $C_0 \vdash^k C_k$ (для $k \geq 1$) – что существуют такие конфигурации $C_1, \ldots, C_{k-1}$, что $C_i \vdash C_{i+1}$ для всех $0 \leq i < k$. Бинарное отношение $\vdash^*$ определяется как рефлексивное и транзитивное замыкание отношения $\vdash$, т.е. $C \vdash^* C'$ означает, что $C \vdash^k C'$ для некоторого $k \geq 0$.

Будем говорить, что автомат M ***распознает*** (или ***допускает***) слово w, если $(q_0, w) \vdash^* (q, \lambda)$ для некоторого $q \in F$.

Язык, определяемый (или ***распознаваемый***, или ***допускаемый***) автоматом M (обозначается $L(M)$), – это множество входных цепочек, допускаемых конечным автоматом M, т.е.

$$L(M) = \left\{ w \,\middle|\, w \in A^* \text{ и } (q_0, w) \vdash^* (q, \lambda) \text{ для некоторого } q \in F \right\}.$$

Язык L называется ***конечно-автоматным***, если для него можно построить распознающий конечный автомат.

Пример 2.1.1. Недерминированный конечный автомат M, допускающий те и только те слова из 0 и 1, которые заканчиваются на 11, может быть формально задан как $\langle \{q_0, q_1, q_2\}, \{0,1\}, q_0, \delta, \{q_2\} \rangle$, где функция переходов δ определяется соотношениями: $\delta(q_0, 0) = \{q_0\}$, $\delta(q_0, 1) = \{q_0, q_1\}$, $\delta(q_1, 1) = \{q_2\}$, $\delta(q_1, 0) = \delta(q_2, 0) = \delta(q_2, 1) = \varnothing$. Для слова 1011 имеем следующую последовательность возможных конфигураций: $(q_0, 1011) \vdash$ $\vdash (q_0, 011) \vdash (q_0, 11) \vdash (q_1, 1) \vdash (q_2, \lambda)$.

Так как $(q_0, 1011) \vdash^* (q_2, \lambda)$, заключаем, что слово $1011 \in L(M)$.

Для описания конечных автоматов существуют два основных способа задания: таблица переходов и диаграмма переходов.

Таблица переходов недетерминированного конечного автомата представляет собой обычное табличное представление функции переходов δ, которая двум своим аргументам $q \in Q$ и $a \in A$ ставит в соответствие значение $\delta(q, a)$. Строки таблицы соответствуют состояниям, а столбцы − входным символам. На пересечении строки, соответствующей состоянию $q \in Q$, и столбца, соответствующего входному символу $a \in A$, находится подмножество $\delta(q, a)$. Если $\delta(q, a) = \varnothing$, то будем говорить, что $\delta(q, a)$ не определено. Заметим, что в случае детерминированного конечного автомата $\delta(q, a)$ представляет собой всегда одноэлементное подмножество, поэтому договоримся в этом случае вместо $\delta(q, a) = \{p\}$ писать просто $\delta(q, a) = p$ (без фигурных скобок).

Диаграмма переходов конечного автомата есть ориентированный граф, каждая вершина которого соответствует некоторому состоянию $q \in Q$, причем вершины, соответствующие заключительным состояниям (состояниям из F), выделяются двойным кружком. Пусть $\delta(q, a) = \{p_1, \ldots, p_k\}$ для некоторого состояния q из Q и входного символа a из A, тогда диаграмма переходов должна содержать дуги, помеченные символом a, из вершины q в каждую из вершин $p_1, \ldots, p_k$.

Пример 2.1.2. Опишем недерминированный конечный автомат M, рассмотренный в примере 2.1.1, с помощью таблицы переходов и диаграммы переходов. В таблице 2.1.1 представлена таблица переходов этого конечного автомата, заметим, что звездочкой здесь отмечено заключительное состояние $q_2 \in F$, а на рис. 2.1.1 изображена диаграмма переходов автомата M.

Таблица 2.1.1

Таблица переходов автомата M из примера 2.1.1

Состояния	Вход	
	0	1
q_0	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\varnothing$	$\{q_2\}$
$*q_2$	$\varnothing$	$\varnothing$

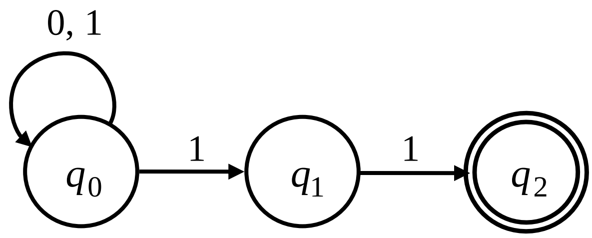


Рис. 2.1.1. Диаграмма переходов автомата из примера 2.1.1

Пример 2.1.3. Построить конечный автомат, распознающий $L \subseteq \{0,1\}^*$, при условии, что слово α принадлежит языку L тогда и только тогда, когда в этом слове содержится подслово 10.

Для того, чтобы решить, содержит ли входная последовательность подслово 10, автомат M должен уметь распознавать следующие важные моменты относительно прочитанных им входных данных.

1) Последовательность 10 была прочитана, тогда с этого момента автомат будет находиться только в заключительных состояниях.

2) Последовательность 10 еще не была считана, и на предыдущем шаге на вход либо ничего не подавалось, либо был считан символ 0. В этом случае автомат M не перейдет в заключительное состояние до тех пор, пока им не будут считаны символы 1, а затем сразу 0.

3) Последовательность 10 еще не была считана, но на предыдущем шаге был считан символ 1. Значит, если на данном шаге читается символ 0, тогда подслово 10 будет прочитано, и с этого момента автомат будет находиться только в заключительных состояниях.

Каждое из этих условий может быть представлено как некоторое состояние. Условию 2) соответствует начальное состояние q_0, условию 3) − состояние q_1, а условию 1) – заключительное состояние q_2.

Детерминированный конечный автомат M, распознающий этот язык, может быть формально задан как система $\langle\{q_0, q_1, q_2\}, \{0,1\}, q_0, \delta, \{q_2\}\rangle$, где функция переходов δ может быть задана либо таблицей переходов, либо диаграммой переходов. В таблице 2.1.2 представлена таблица переходов конечного автомата M, диаграмма переходов изображена на рис. 2.1.2.

Таблица 2.1.2

Таблица переходов автомата M из примера 2.1.3

Состояния	Вход	
	0	1
q_0	q_0	q_1
q_1	q_2	q_1
$*q_2$	q_2	q_2

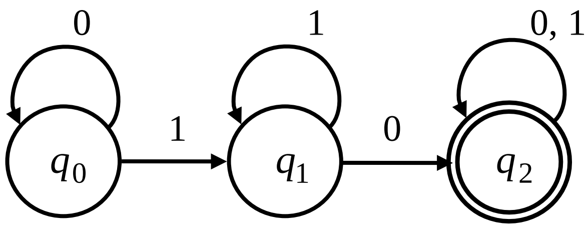


Рис. 2.1.2. Диаграмма переходов автомата M из примера 2.1.3

Задачи и упражнения

В задачах **2.1.1** − **2.1.18** требуется построить конечный автомат, распознающий язык L в алфавите $A = \{a, b, c\}$.

2.1.1. Слово α принадлежит языку L тогда и только тогда, когда в этом слове не встречается буква a.

2.1.2. Слово α принадлежит языку L тогда и только тогда, когда в этом слове встречается хотя бы одна буква a.

2.1.3. Слово α принадлежит языку L тогда и только тогда, когда в этом слове буква a встречается ровно два раза.

2.1.4. Слово α принадлежит языку L тогда и только тогда, когда в этом слове буква a встречается не менее двух раз.

2.1.5. Слово α принадлежит языку L тогда и только тогда, когда в этом слове буква a встречается четное число раз

2.1.6. Слово α принадлежит языку L тогда и только тогда, когда в этом слове буква a встречается нечетное число раз.

2.1.7. Слово α принадлежит языку L тогда и только тогда, когда в этом слове буква a не встречается дважды подряд.

2.1.8. Слово α принадлежит языку L тогда и только тогда, когда в нем содержится подслово $abbc$.

2.1.9. Слово α принадлежит языку L тогда и только тогда, когда в нем содержится подслово $bbaa$.

2.1.10. Слово α принадлежит языку L тогда и только тогда, когда в нем содержится подслово $cbca$.

2.1.11. Слово α принадлежит языку L тогда и только тогда, когда в нем подслово ac встречается не более двух раз.

2.1.12. Слово α принадлежит языку L тогда и только тогда, когда в этом слове первая буква совпадает с последней.

2.1.13. Слово α принадлежит языку L тогда и только тогда, когда в этом слове не содержится подслово cab.

2.1.14. Слово α принадлежит языку L тогда и только тогда, когда оно содержит четное число букв.

2.1.15. Слово α принадлежит языку L тогда и только тогда, когда если оно содержит букву a, то содержит и букву b.

2.1.16. Слово α принадлежит языку L тогда и только тогда, когда каждая буква алфавита встречается в нем четное число раз.

2.1.17. Слово α принадлежит языку L тогда и только тогда, когда в этом слове буква a встречается четное число раз, а буква b − нечетное число раз.

2.1.18. Слово α принадлежит языку L тогда и только тогда, когда оно заканчивается на букву, ранее встречавшуюся в слове, причем $|\alpha| \geq 1$.

2.1.19. Построить конечный автомат, распознающий множество слов в алфавите $A = \{a, b\}$, длина которых кратна 3.

2.1.20. Построить конечный автомат, распознающий множество слов в алфавите $A = \{a, b\}$, длина которых кратна 4.

2.1.21. Построить конечный автомат, распознающий множество слов в алфавите $A = \{ a, b \}$, длина которых кратна 5.

2.1.22. Построить конечный автомат, распознающий множество слов в алфавите $A = \{ a, b \}$, длина которых кратна 6.

2.1.23. Построить конечный автомат, распознающий числа, кратные 4.

2.1.24. Построить конечный автомат, распознающий числа, кратные 5.

2.1.25. Построить конечный автомат, распознающий числа, кратные 6.

2.1.26. Построить конечный автомат, распознающий числа, кратные 7.

2.1.27. Построить конечный автомат, распознающий числа, кратные 8.

2.1.28. Построить конечный автомат, распознающий числа, кратные 12.

2.1.29. Построить конечный автомат, распознающий числа, кратные 25.

2.1.30. Построить конечный автомат, распознающий числа, кратные 50.

2.1.31. Построить конечный автомат, распознающий числа, кратные 100.

2.1.32. Построить конечный автомат, распознающий числа, кратные 125.

2.2. Переход от недетерминированного конечного автомата к детерминированному

Один из наиболее важных результатов теории конечных автоматов состоит в том, что класс языков, определяемых недерминированными конечными автоматами, совпадает с классом языков, определяемых полностью определенными детерминированными конечными автоматами.

Для многих языков построить допускающий недерминированный конечный автомат гораздо легче, чем детерминированный.

Рассмотрим ***алгоритм детерминизации*** конечного автомата.

Пусть $M = \langle Q, A, q_0, \delta, F \rangle$ − недетерминированный конечный автомат, распознающий язык $L(M)$. Требуется построить детерминированный конечный автомат $M' = \langle Q', A, q_0', \delta', F' \rangle$, такой, что $L(M') = L(M)$. Положим $Q' = 2^Q$, т.е. состояниями автомата M' являются подмножества состояний автомата M, $q_0' = \{q_0\}$, $\delta'(S, a) = \bigcup_{p \in S} \delta(p, a)$ для каждого множества $S \subseteq Q$ и каждой входной буквы $a \in A$. Таким образом, чтобы найти $\delta'(S, a)$, мы рассматриваем все состояния p в S, затем ищем те состояния Q, в которые можно попасть из состояния p по букве a, после чего берем объединение множеств найденных состояний по всем состояниям p.

Множество заключительных состояний детерминированного конечного автомата $F' = \{S \subseteq Q : S \cap F \neq \varnothing\}$, т.е. состоит из всех подмножеств состояний Q, содержащих хотя бы одно "хорошее" состояние из F.

Два автомата называются ***эквивалентными***, если они распознают один и тот же язык.

Теорема 2.2.1. *Для любого недерминированного конечного автомата с n состояниями всегда можно построить эквивалентный ему детерминированный конечный автомат, имеющий не более чем 2^n состояний.*

Пример 2.2.1. Рассмотрим недерминированный конечный автомат M, допускающий те и только те слова из 0 и 1, которые заканчиваются на 11. Этот автомат был построен в примере 2.1.1, диаграмма переходов этого автомата изображена на рис. 2.1.1. Построим эквивалентный ему детерминированный конечный автомат M'.

Недетерминированный конечный автомат M имеет три состояния, поэтому детерминированный автомат M' имеет не более $2^3 = 8$ состояний, представляющих собой различные подмножества, составленные из элементов множества $Q = \{q_0, q_1, q_2\}$. С помощью алгоритма детерминизации построим полную таблицу переходов автомата M' (см. табл. 2.2.1).

Таблица 2.2.1

Таблица переходов автомата M' из примера 2.2.1

Состояния	Вход	
	0	1
$\varnothing$	$\varnothing$	$\varnothing$
$\{q_0\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_1\}$	$\varnothing$	$\{q_2\}$
$*\{q_2\}$	$\varnothing$	$\varnothing$
$\{q_0, q_1\}$	$\{q_0\}$	$\{q_0, q_1, q_2\}$
$*\{q_0, q_2\}$	$\{q_0\}$	$\{q_0, q_1\}$
$*\{q_1, q_2\}$	$\varnothing$	$\{q_2\}$
$*\{q_0, q_1, q_2\}$	$\{q_0\}$	$\{q_0, q_1, q_2\}$

Просматривая строки этой таблицы, несложно понять, что из начального состояния $\{q_0\}$ автомата M' за конечное число шагов можно попасть только в три из восьми состояний: $\{q_0\}$, $\{q_0, q_1\}$ и $\{q_0, q_1, q_2\}$. Остальные пять состояний из начального недостижимы, поэтому их можно исключить из таблицы.

Переобозначим полученные состояния: $\{q_0\}$ обозначим за q'_0, $\{q_0, q_1\}$ – за q'_1, $\{q_0, q_1, q_2\}$ – за q'_2. Состояние q'_0 является начальным, состояние q'_2 – заключительным. В таблице 2.2.2 представлена таблица переходов детерминированного автомата M' с учетом лишь достижимых состояний.

Таблица 2.2.2

Таблица переходов автомата M' из примера 2.2.1

Состояния	Вход	
	0	1
q'_0	q'_0	q'_1
q'_1	q'_0	q'_2
$*q'_2$	q'_0	q'_2

На рис. 2.2.1 изображена диаграмма переходов полученного детерминированного конечного автомата, распознающего двоичные слова, которые заканчиваются на 11. Заметим, что он имеет всего три состояния. Это число случайно оказалось равным числу состояний недетерминированного автомата M, изображенного на рис. 2.1.1, по которому строился этот детерминированный автомат M'. Но детерминированный конечный автомат на рис. 2.2.1 имеет шесть переходов, а автомат на рис. 2.1.1 – лишь четыре.

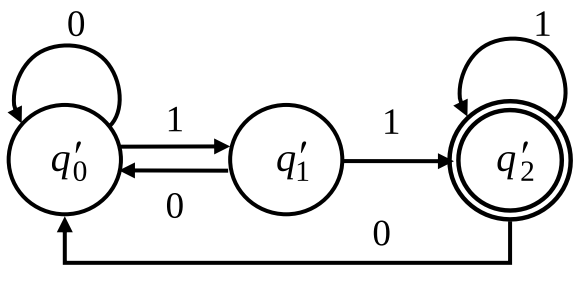


Рис. 2.2.1. Диаграмма переходов детерминированного автомата из примера 2.2.1

Замечание. В дальнейшем детерминированный полностью определенный конечный автомат будем называть просто конечным автоматом.

Задачи и упражнения

2.2.1. Преобразовать недетерминированный автомат M (см. табл. 2.2.3) в эквивалентный ему детерминированный автомат.

Таблица 2.2.3

Таблица переходов автомата M из задачи 2.2.1

Состояния	Вход	
	a	b
$^*q_0$	$\{q_0\}$	$\{q_1\}$
q_1	$\{q_1, q_2\}$	$\{q_0\}$
q_2	$\{q_3\}$	$\{q_2\}$
q_3	$\{q_1\}$	$\{q_3\}$

2.2.2. Преобразовать недетерминированный автомат M (см. табл. 2.2.4) в эквивалентный ему детерминированный автомат.

Таблица 2.2.4

Таблица переходов автомата M из задачи 2.2.2

Состояния	Вход	
	a	b
$^*q_0$	$\{q_1\}$	$\{q_2\}$
q_1	$\{q_0, q_3\}$	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_3\}$
q_3	$\{q_0\}$	$\{q_2, q_3\}$

2.2.3. Преобразовать недетерминированный автомат M (см. табл. 2.2.5) в эквивалентный ему детерминированный автомат.

Таблица 2.2.5

Таблица переходов автомата M из задачи 2.2.3

Состояния	Вход	
	a	b
q_0	$\{q_0\}$	$\{q_1, q_2\}$
q_1	$\{q_2, q_3\}$	$\{q_1\}$
q_2	$\{q_2\}$	$\{q_0, q_1\}$
$^*q_3$	$\{q_1, q_2\}$	$\{q_3\}$

2.2.4. Преобразовать недетерминированный автомат M (см. табл. 2.2.6) в эквивалентный ему детерминированный автомат.

Таблица 2.2.6

Таблица переходов автомата M из задачи 2.2.4

Состояния	Вход		
	a	b	c
q_0	$\{q_1\}$	$\varnothing$	$\varnothing$
q_1	$\{q_2\}$	$\{q_0\}$	$\{q_0, q_1\}$
$^*q_2$	$\{q_0\}$	$\{q_2\}$	$\{q_2\}$

2.2.5. По недетерминированному конечному автомату, который распознает все слова в алфавите $A=\{a,b,c\}$, заканчивающиеся на букву c, построить эквивалентный ему детерминированный автомат.

2.2.6. По недетерминированному конечному автомату, распознающему все слова в алфавите $A=\{a,b\}$, которые начинаются и заканчиваются на одну букву, построить эквивалентный ему детерминированный автомат.

2.2.7. По недетерминированному конечному автомату, распознающему все слова в алфавите $A=\{a,b\}$, заканчивающиеся на букву, которая уже встречалась ранее, построить эквивалентный ему полностью определенный детерминированный автомат.

2.2.8. По недетерминированному конечному автомату, распознающему множество чисел в десятичной системе счисления, делящихся на 25, построить эквивалентный ему детерминированный конечный автомат.

2.2.9. По недетерминированному конечному автомату, распознающему множество чисел в десятичной системе счисления, делящихся на 125, построить эквивалентный ему детерминированный конечный автомат.

2.3. Замкнутость конечно-автоматных языков относительно теоретико-множественных операций

Теорема 2.3.1. *Пустой язык, универсальный язык, любой конечный язык в произвольном фиксированном алфавите* $A=\{a_1, ..., a_n\}$ *являются конечно-автоматными языками.*

Теорема 2.3.2. *Класс конечно-автоматных языков замкнут относительно основных теоретико-множественных операций – объединения, пересечения и дополнения.*

Рассмотрим алгоритмы построения соответствующих автоматов.

Пусть L_1 и L_2 – конечно-автоматные языки, распознаваемые конечными автоматами $M^1 = \left\langle Q^1, A, q_0^1, \delta^1, F^1 \right\rangle$ и $M^2 = \left\langle Q^2, A, q_0^2, \delta^2, F^2 \right\rangle$ соответственно.

1) ***Операция объединения.*** Автомат $M^{\cup} = \left\langle Q, A, q_0, \delta, F^{\cup} \right\rangle$, распознающий язык $L_1 \cup L_2$, строим следующим образом. Полагаем $Q = Q^1 \times Q^2$, т.е. каждое состояние автомата $M^{\cup}$ содержит две компоненты, левую и правую. Начальным состоянием нового автомата считаем $\left(q_0^1, q_0^2\right)$, а функцию переходов δ определяем следующим образом:

$$\delta\left(\left(q_i^1, q_j^2\right), a_k\right) = \left(\delta^1\left(q_i^1, a_k\right), \delta^2\left(q_j^2, a_k\right)\right).$$

Входное слово α принадлежит объединению языков L_1 и L_2 тогда и только тогда, когда после его обработки автомат $M^{\cup}$ окажется в состоянии, у которого либо первая компонента принадлежит совокупности F^1, либо вторая компонента принадлежит совокупности F^2.Таким образом, следует положить:

$$F^{\cup} = \left(F^1 \times Q^2\right) \cup \left(Q^1 \times F^2\right).$$

Все компоненты автомата $M^{\cup}$ определены, его построение закончено.

2) ***Операция пересечения.*** Автомат $M^{\cap} = \left\langle Q, A, q_0, \delta, F^{\cap} \right\rangle$, распознающий язык $L_1 \cap L_2$, отличается от $M^{\cup}$ только последней своей компонентой. Здесь получаем, что $F^{\cap} = F^1 \times F^2$.

Пример 2.3.1. Для конечных автоматов (рис. 2.3.1), распознающих языки L_1 и L_2, построить конечные автоматы, распознающие объединение этих языков и их пересечение. На рис. 2.3.2 и 2.3.3 изображены конечные автоматы, распознающие языки $L_1 \cup L_2$ и $L_1 \cap L_2$ соответственно.

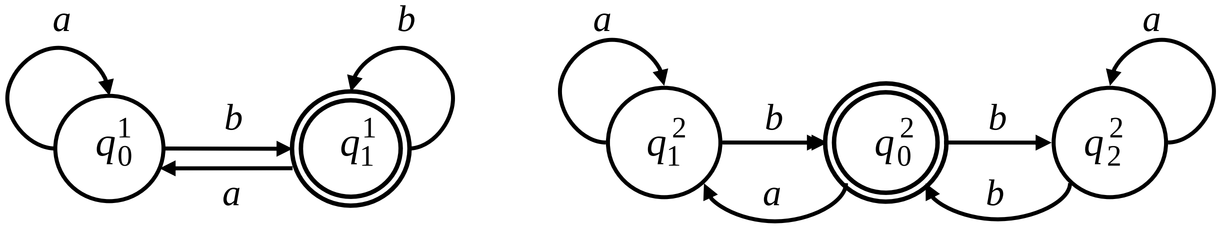


Рис. 2.3.1. Диаграммы конечных автоматов, распознающих L_1 и L_2

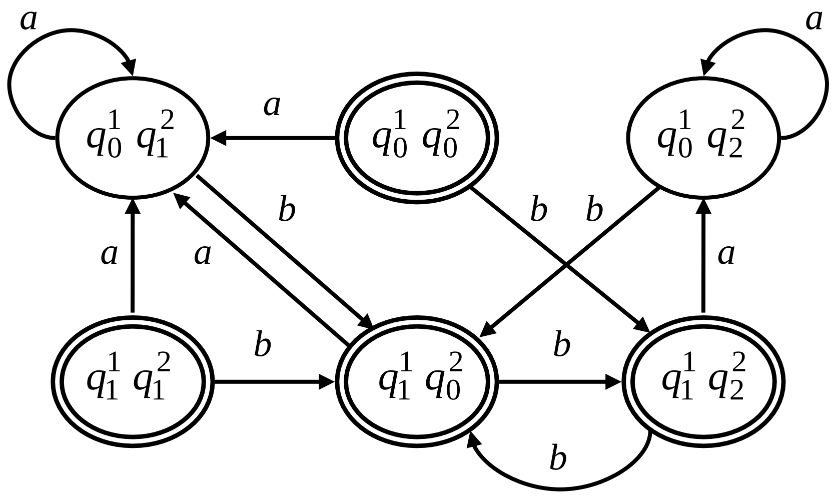


Рис. 2.3.2. Диаграмма конечного автомата, распознающего $L_1 \cup L_2$

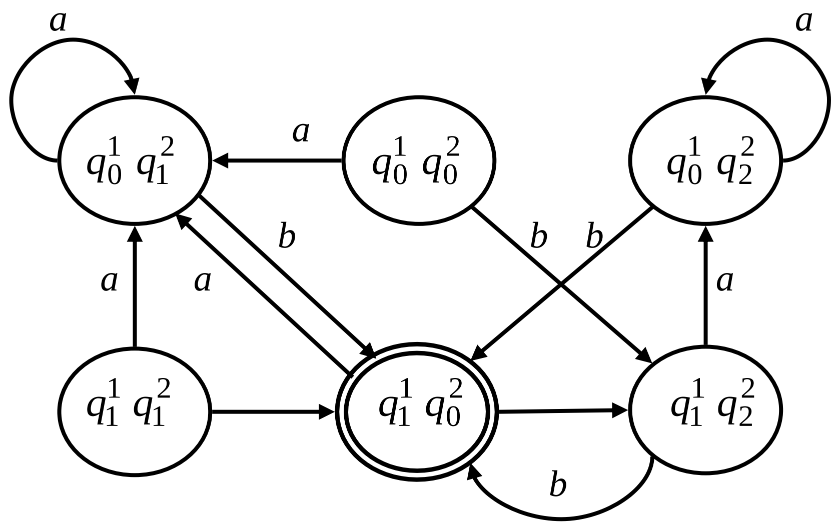


Рис. 2.3.3. Диаграмма конечного автомата, распознающего $L_1 \cap L_2$

3) ***Операция дополнения.*** Пусть $M = \langle Q, A, q_0, \delta, F \rangle$ – конечный автомат, распознающий язык L. Произвольное слово $\alpha \in A^*$ принадлежит языку $\overline{L} = A^* - L$ тогда и только тогда, когда после его обработки автомат M оказывается в состоянии, не принадлежащем F. Тогда $\overline{L} = L(N)$, где $N = \langle Q, A, q_0, \delta, Q - F \rangle$, т.е. для того, чтобы получить конечный автомат, распознающий дополнение регулярного языка, надо в автомате, распознающем исходный язык, поменять местами "хорошие" и "плохие" состояния.

Пример 2.3.2. Язык L распознается конечным автоматом, диаграмма переходов которого представлена на рис. 2.3.4. Построить конечный автомат, распознающий его дополнение.

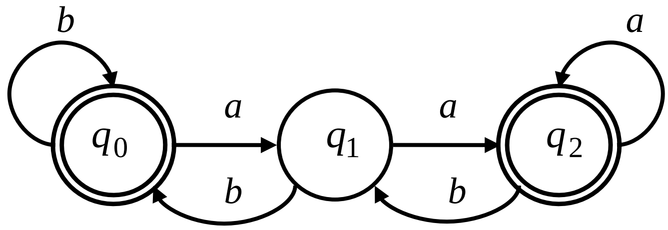


Рис. 2.3.4. Диаграмма конечного автомата, распознающего язык L

На рис. 2.3.5 представлен конечный автомат, распознающий дополнение языка L.

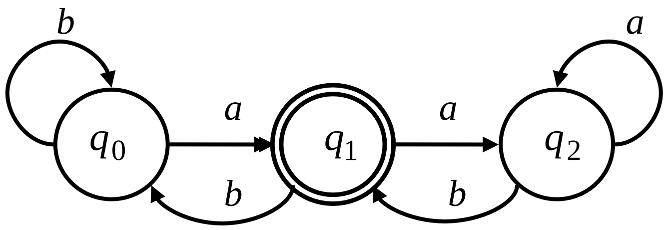


Рис. 2.3.5. Диаграмма конечного автомата, распознающего язык $\overline{L}$

Отметим, что класс конечно-автоматных языков замкнут также относительно операции разности, так как $L_1 - L_2 = L_1 \cap \overline{L}_2$.

Рассмотрим теперь алгоритм построения недетерминированного конечного автомата, распознающего объединение двух конечно-автоматных языков. Этот алгоритм можно применять вне зависимости от того, являются ли детерминированными автоматы, определяющие исходные языки.

Пусть L_1 и L_2 – конечно-автоматные языки, распознаваемые конечными автоматами $M^1 = \left\langle Q^1, A, q_0^1, \delta^1, F^1 \right\rangle$ и $M^2 = \left\langle Q^2, A, q_0^2, \delta^2, F^2 \right\rangle$ соответственно. Для построения $M^{\cup}$ добавим в диаграмму новое состояние q_0. По каждой букве $x \in A$ из состояния q_0 проводим две дуги с надписанной буквой x: $\left(q_0, \delta^1\left(q_0^1, x\right)\right)$ и $\left(q_0, \delta^2\left(q_0^2, x\right)\right)$. Начальным состоянием $M^{\cup}$ является состояние q_0, а множеством заключительных состояний F автомата является множество $F^1 \cup F^2$. В случае, если либо $q_0^1 \in F^1$, либо $q_0^2 \in F^2$, тогда $q_0 \in F$.

Пример 2.3.3. Построить конечный автомат, распознающий числа, каждое из которых кратно 2 или 5, над алфавитом $A=\{0,2,3,5\}$.

На рис. 2.3.6 представлена диаграмма переходов детерминированного конечного автомата M_1, распознающего числа, кратные 2, а на рис. 2.3.7 – диаграмма автомата M_2, распознающего числа, кратные 5.

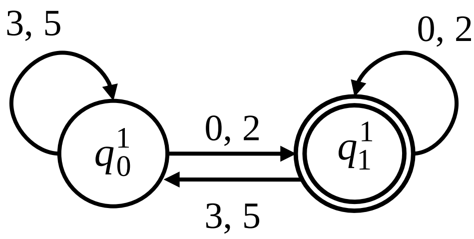


Рис. 2.3.6. Диаграмма переходов M_1

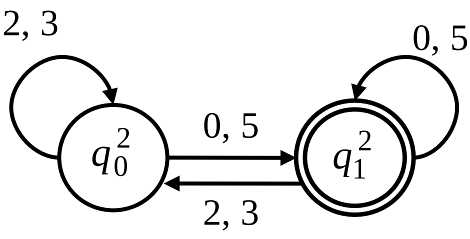


Рис. 2.3.7. Диаграмма переходов M_2

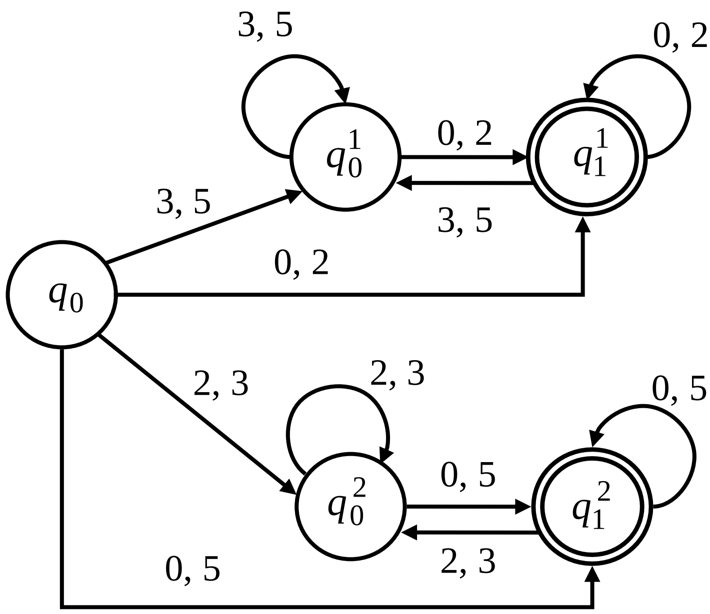


Рис. 2.3.8. Диаграмма автомата, распознающего числа, кратные 2 или 5

На рис. 2.3.8 построена диаграмма переходов недетермированного конечного автомата, распознающего множество десятичных чисел над алфавитом $A=\{0,2,3,5\}$, каждое из которых кратно 2 или 5.

Задачи и упражнения

2.3.1. Языки L_1 и L_2 распознаются конечными автоматами, диаграммы переходов которых изображены на рис. 2.3.9. Построить конечные автоматы, распознающие языки $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 - L_2$ и $L_2 - L_1$.

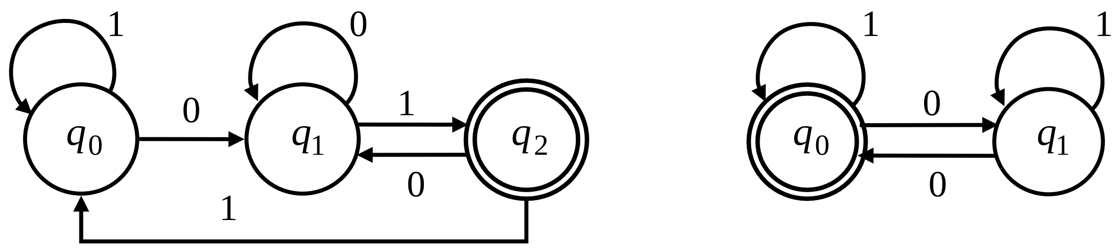


Рис. 2.3.9. Диаграммы переходов автоматов к задаче 2.3.1

2.3.2. Языки L_1 и L_2 распознаются конечными автоматами, диаграммы переходов которых изображены на рис. 2.3.10. Построить конечные автоматы, распознающие языки $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 - L_2$ и $L_2 - L_1$.

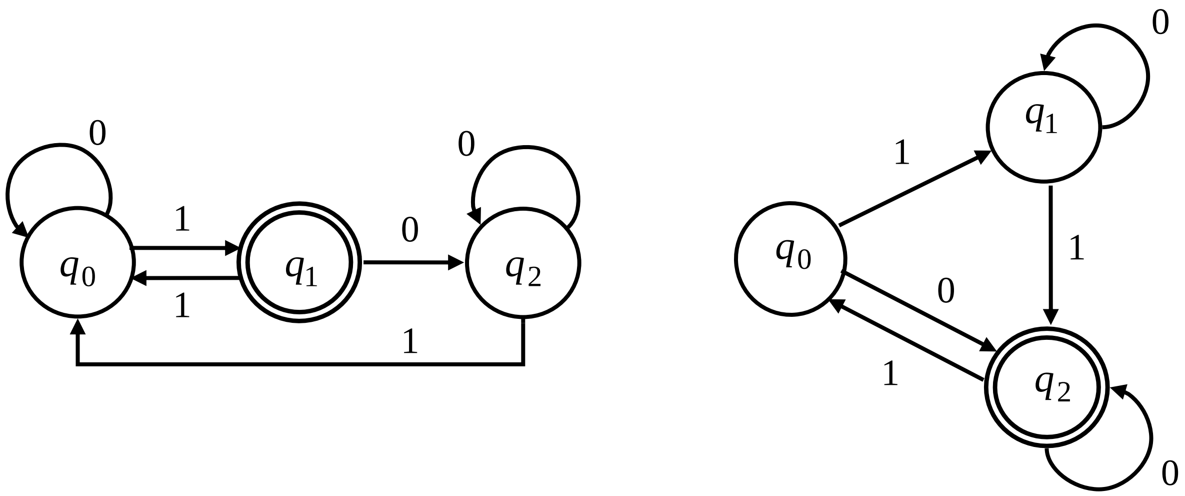


Рис. 2.3.10. Диаграммы переходов автоматов к задаче 2.3.2

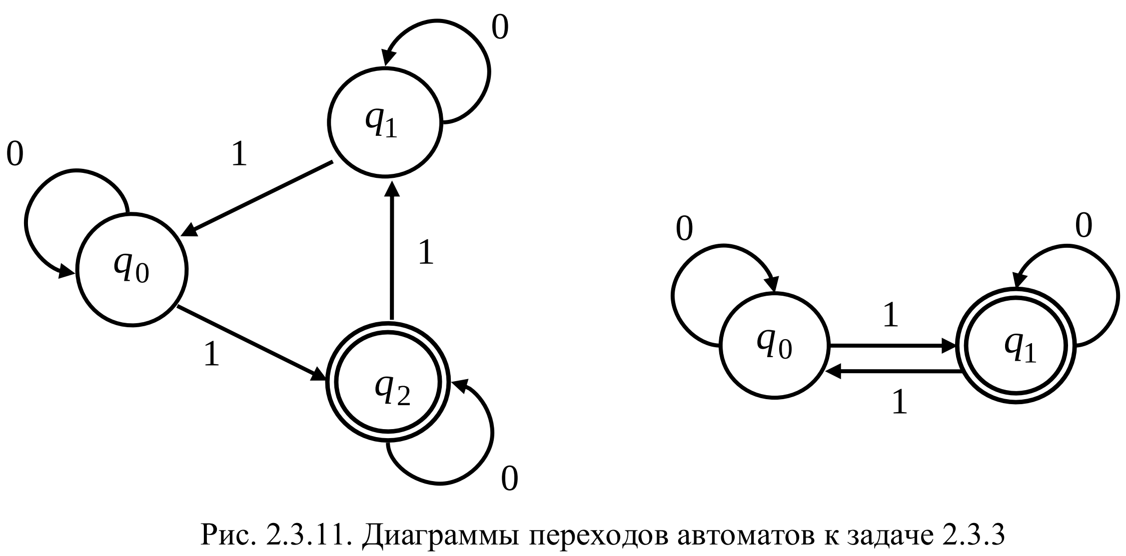


Рис. 2.3.11. Диаграммы переходов автоматов к задаче 2.3.3

2.3.3. Языки L_1 и L_2 распознаются конечными автоматами, диаграммы переходов которых изображены на рис. 2.3.11. Построить конечные автоматы, распознающие языки $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 - L_2$ и $L_2 - L_1$.

2.3.4. Языки L_1 и L_2 распознаются конечными автоматами, диаграммы переходов которых изображены на рис. 2.3.12. Построить конечные автоматы, распознающие языки $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 - L_2$ и $L_2 - L_1$.

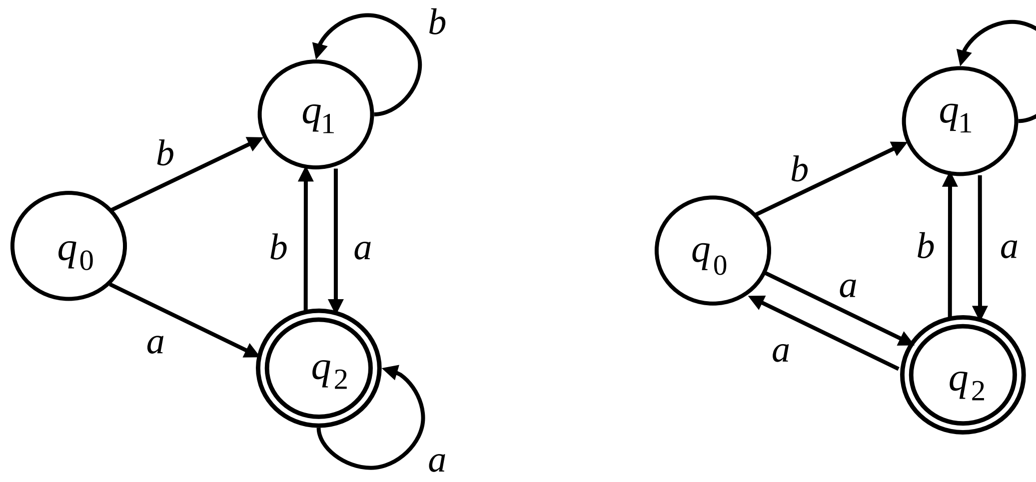


Рис. 2.3.12. Диаграммы переходов автоматов к задаче 2.3.4

2.4. Замкнутость конечно-автоматных языков относительно конкатенации, возведения в степень и итерации

Теорема 2.4.1. *Класс конечно-автоматных языков замкнут относительно операции конкатенации.*

Пусть L_1 и L_2 – конечно-автоматные языки, распознаваемые конечными автоматами $M^1 = \left\langle Q^1, A, q_0^1, \delta^1, F^1 \right\rangle$ и $M^2 = \left\langle Q^2, A, q_0^2, \delta^2, F^2 \right\rangle$ соответственно. При построении конечного автомата $M^\circ = \left\langle Q, A, q_0, \delta, F \right\rangle$, распознающего конкатенацию языков $L_1 L_2$, возможны следующие случаи.

Случай 1. $q_0^1 \notin F^1$, т.е. пустое слово не входит в язык L_1. Через W обозначим совокупность всех дуг (вместе с надписанными над ними буквами) в диаграмме переходов автомата M^1, которые своими концами имеют вершины подмножества F^1, т.е. все дуги вида $\left(q_i^1, q_j^1\right)$, где $q_j^1 \in F^1$. Для каждой принадлежащей множеству W дуги $\left(q_i^1, q_j^1\right)$ с надписанной над ней буквой x входного алфавита строим дугу-дубликат $\left(q_i^1, q_0^2\right)$ с той же надписанной буквой. Так реализуется «сцепка» диаграмм переходов конечных автоматов M^1 и M^2, результатом которой является диаграмма переходов конструируемого недетерминированного конечного автомата M°, положим теперь $q_0 = q_0^1$, $F = F^2$. Согласно выполненному построению, до перехода по некоторой дуге-дубликату в состояние q_0^2 конечный автомат M° функционирует как автомат M^1, после перехода в q_0^2 и до завершения работы – как M^2.

Случай 2. $q_0^1 \in F^1, q_0^2 \notin F^2$, т.е. в язык L_1 входит пустое слово, а в язык L_2 – не входит. Для этого случая «сцепка» диаграмм переходов конечных автоматов M^1 и M^2, результатом которой является диаграмма переходов конструируемого недетерминированного конечного автомата M°, реализуется в два шага.

1 шаг. Проводятся все дуги-дубликаты, как для случая 1.

2 шаг. Для каждой дуги $\left(q_0^2, q_i^2\right)$ с надписанной над ней буквой x входного алфавита проводится дополнительная дуга $\left(q_0^1, q_i^2\right)$ с той же надписанной над ней буквой x.

Как и в случае 1, $q_0 = q_0^1$ и $F = F^2$.

Случай 3. $q_0^1 \in F^1, q_0^2 \in F^2$, т.е. пустое слово входит и в язык L_1, и в язык L_2, значит, пустое слово принадлежит и конкатенации языков $L_1 L_2$. Алгоритм построения недетерминированного автомата M в этом случае реализуется в два шага, как и в случае 2. Однако, здесь $q_0 = q_0^1$ и $F = F^2 \cup \left\{q_0^1\right\}$.

Пример 2.4.1. Языки L_1 и L_2 распознаются конечными автоматами, диаграммы которых представлены на рис. 2.4.1. Построить конечный автомат, распознающий конкатенацию этих языков.

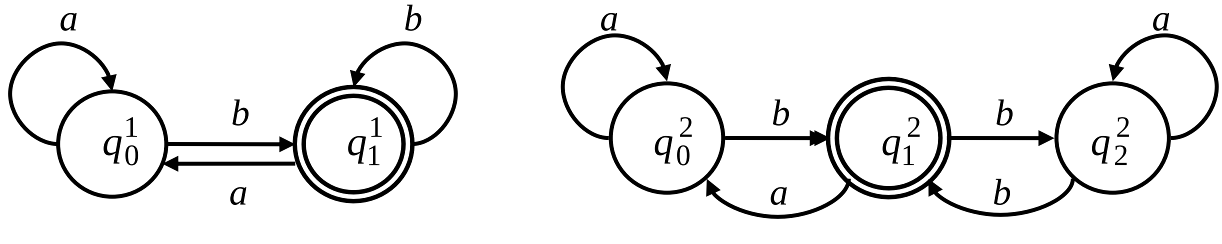


Рис. 2.4.1. Диаграммы переходов конечных автоматов, распознающих L_1 и L_2

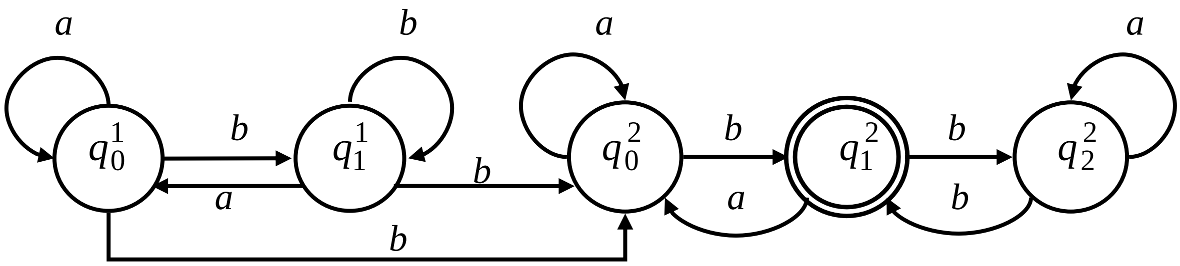


Рис. 2.4.2. Диаграмма переходов конечного автомата, распознающего L_1L_2

Так как $q_0^1 \notin F^1$, имеем случай 1. На рис. 2.4.2 представлена диаграмма переходов недетерминированного конечного автомата, распознающего конкатенацию языков L_1L_2.

Пример 2.4.2. Для конечных автоматов (рис. 2.4.3), распознающих языки L_1 и L_2, построить конечный автомат, распознающий конкатенацию этих языков.

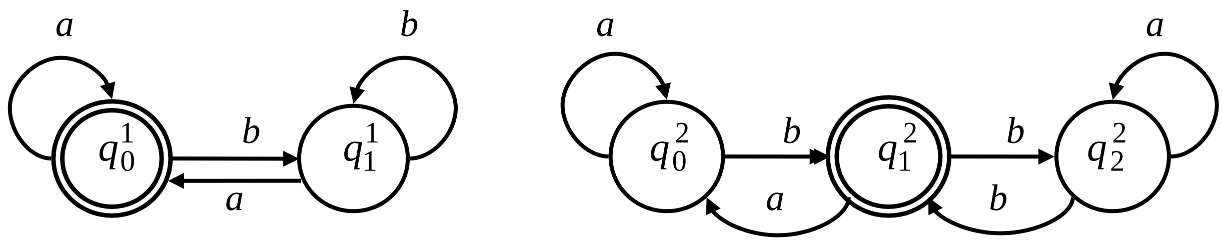


Рис. 2.4.3. Диаграммы переходов конечных автоматов, распознающих L_1 и L_2

Так как $q_0^1 \in F^1, q_0^2 \notin F^2$, имеем случай 2. На рис. 2.4.4 представлена диаграмма переходов недетерминированного конечного автомата, распознающего конкатенацию языков L_1L_2.

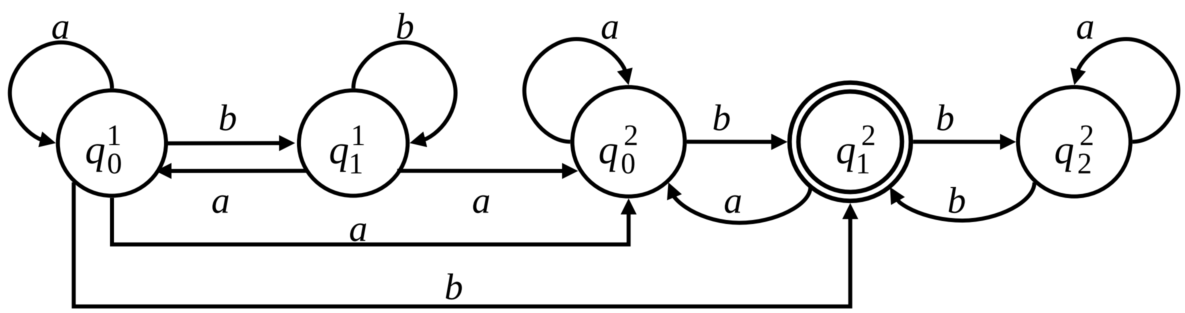


Рис. 2.4.4. Диаграмма переходов конечного автомата, распознающего L_1L_2

Пример 2.4.3. Для конечных автоматов (рис. 2.4.5), распознающих языки L_1 и L_2, построить конечный автомат, распознающий конкатенацию этих языков.

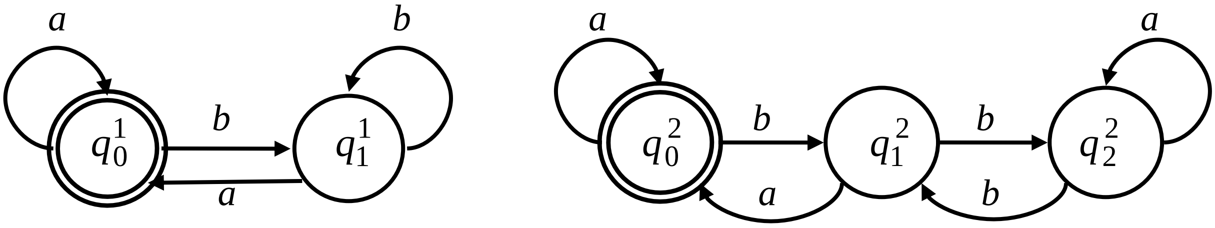


Рис. 2.4.5. Диаграммы переходов конечных автоматов, распознающих L_1 и L_2

Так как $q_0^1 \in F^1, q_0^2 \in F^2$, имеем случай 3. На рис. 2.4.6 представлена диаграмма переходов недетерминированного конечного автомата, распознающего конкатенацию языков $L_1 L_2$.

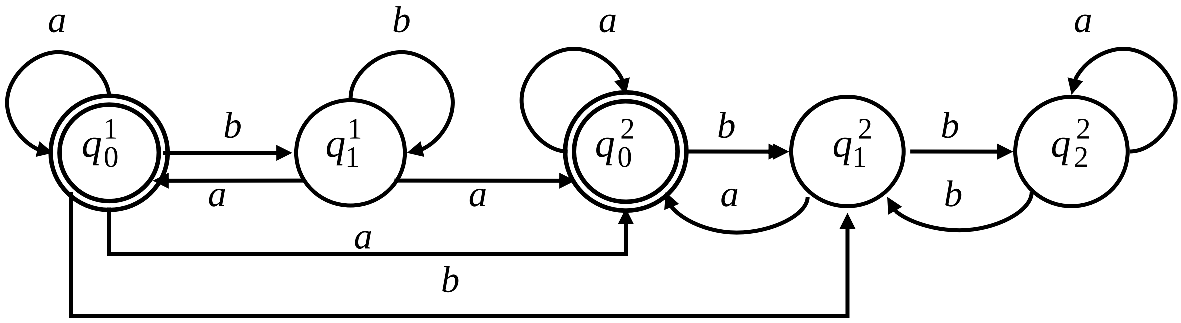


Рис. 2.4.6. Диаграмма переходов конечного автомата, распознающего $L_1 L_2$

Теорема 2.4.2. *Класс конечно-автоматных языков замкнут относительно операции возведения в любую целую неотрицательную степень.*

Построение конечного автомата, реализующего язык L^2, осуществляется по алгоритму построения конечного автомата, распознающего конкатенацию LL, соответственно построение конечного автомата, распознающего язык L^n при любом $n \geq 3$, осуществляется по алгоритму построения конечного автомата, распознающего конкатенацию $L^{n-1}L$.

Пример 2.4.4. По диаграмме конечного автомата (рис. 2.4.7), распознающего язык L, построить конечный автомат, распознающий язык L^3.

Построение конечного автомата, распознающего язык L^3, произведем в два этапа.

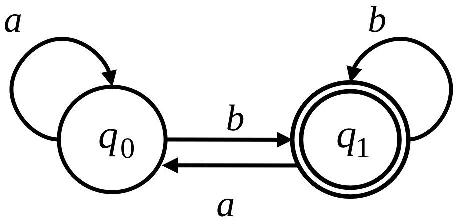


Рис.2.4.7. Диаграмма автомата к задаче 2.4.4.

Сначала построим диаграмму переходов конечного автомата, распознающего язык L^2 (рис. 2.4.8).

На рис. 2.4.9 представлена диаграмма переходов конечного автомата, распознающего язык L^3.

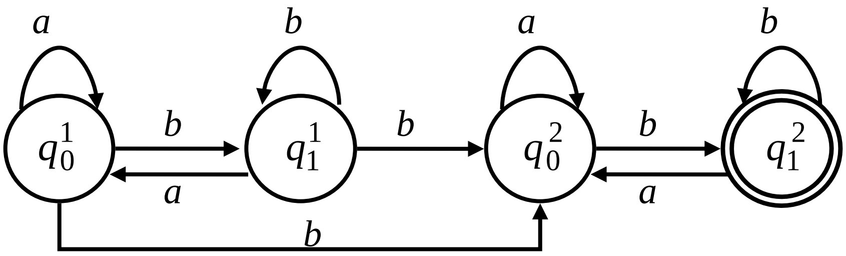


Рис.2.4.8. Диаграмма переходов автомата, распознающего L^2

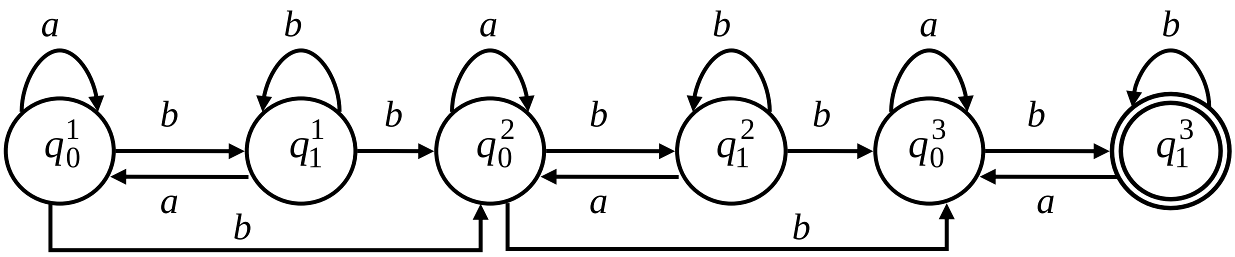


Рис. 2.4.9. Диаграмма переходов автомата, распознающего L^3

Теорема 2.4.3. *Класс конечно-автоматных языков замкнут относительно операции итерации.*

Пусть L − конечно-автоматный язык и $M = \langle Q, A, q_0, \delta, F \rangle$ – конечный автомат, распознающий L.

Рассмотрим алгоритм построения конечного автомата (вообще говоря, недетерминированного), распознающего язык L^*. Возможны два случая.

Случай 1. Начальное состояние q_0 автомата M − ***невозвратное***, т.е. отсутствуют дуги, входящие в q_0. Через W обозначим совокупность всех дуг (вместе с надписанными над ними буквами) в диаграмме переходов автомата M, которые своими концами имеют вершины подмножества F, т.е. все дуги вида (q_i, q_j), где $q_j \in F$. Для каждой принадлежащей множеству W дуги (q_i, q_j) с надписанной над ней буквой x входного алфавита строим дугу-дубликат (q_i, q_0) с той же надписанной буквой. Начальным состоянием автомата $M(L^*)$ считаем q_0, это же состояние считаем единственным "хорошим" состоянием в $M(L^*)$. Следует заметить, что, если в качестве "хороших" состояний автомата $M(L^*)$ считать множество F исходного автомата M, тогда получаемый автомат распознает язык L^+.

Случай 2. Начальное состояние q_0 автомата M не является невозвратным. Построение автомата, распознающего язык L^*, производим следующим образом.

Шаг 1. Для начального состояния q_0 автомата M вводим дублирующее состояние q_0^1. Все дуги (с надписанными над ними буквами), которые в диаграмме исходного автомата входят в вершину q_0, переориентируем в q_0^1, при этом каждая петля (q_0, q_0) преобразуется в дугу (q_0, q_0^1). Для каждой имеющейся дуги (q_0, q_j), где $q_j \in Q$, с надписанной над ней буквой, проводим дугу (q_0^1, q_j) с той же надписанной буквой, при этом для каждой ранее имевшейся петли (q_0, q_0) проводится, с той же надписанной буквой, петля (q_0^1, q_0^1). Множество "хороших" состояний построенного автомата M' считаем совпадающим с множеством "хороших" состояний исходного автомата M, причем, если $q_0 \in F$, тогда и $q_0^1 \in F$.

Шаг 2. Далее следует применить алгоритм, описанный для случая невозвратного начального состояния конечного автомата.

Замечание. Если начальное состояние q_0 не является невозвратным и $q_0 \in F$, то при построении конечного автомата, распознающего итерацию языка, можно сразу переходить к шагу 2.

Пример 2.4.5. Пусть $L \subseteq \{a, b, c\}^{+}$ – язык, состоящий из слов, которые начинаются на букву c и в каждом из которых буква c встречается ровно два раза. Построить конечный автомат, распознающий язык L^{*}.

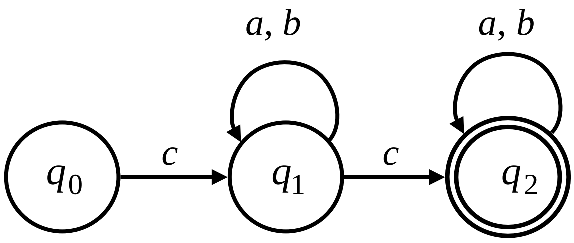


Рис. 2.4.10. Автомат, распознающий L

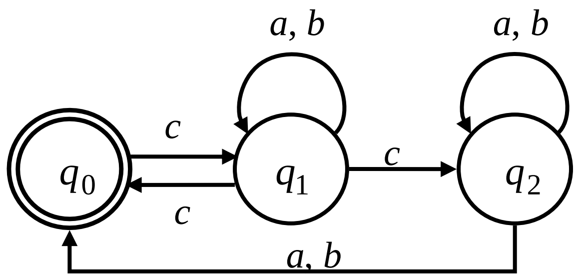


Рис. 2.4.11. Автомат, распознающий L^{*}

На рис. 2.4.10 приведена диаграмма переходов конечного автомата, распознающего язык L. Так как состояние q_0 исходного автомата является невозвратным, имеем случай 1. На рис. 2.4.11 приведена диаграмма переходов конечного автомата, распознающего язык L^{*}. Непустое слово α принимается автоматом, распознающим итерацию языка L, тогда и только тогда, когда буква c встречается в нем любое четное, отличное от нуля, число раз.

Пример 2.4.6. Язык L распознается конечным автоматом, диаграмма которого представлена на рис. 2.4.7. Построить конечный автомат, распознающий язык L^{*}.

Так как состояние q_0 исходного автомата не является невозвратным, имеем случай 2. На рис. 2.4.12 приведена диаграмма переходов конечного автома-

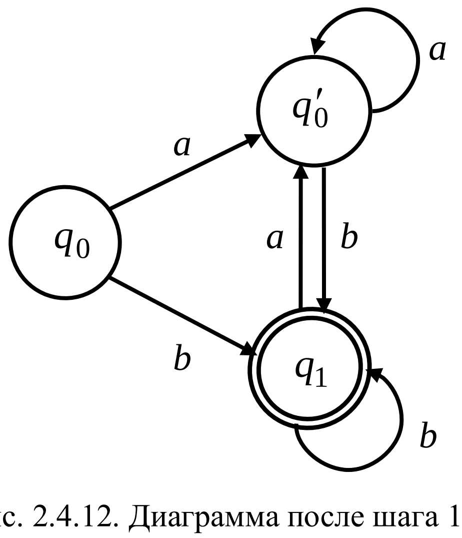


Рис. 2.4.12. Диаграмма после шага 1

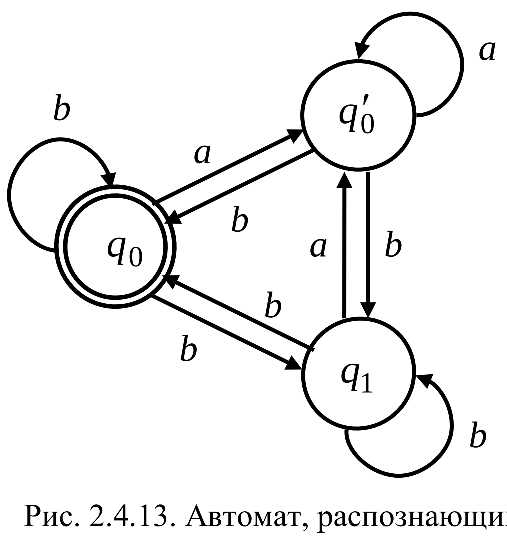


Рис. 2.4.13. Автомат, распознающий L^{*}

та, распознающего язык L, после выполнения шага 1, а на рис. 2.4.13 изображена диаграмма переходов конечного автомата, распознающего язык L^*.

Задачи и упражнения

2.4.1. Языки L_1 и L_2 распознаются конечными автоматами, диаграммы переходов которых изображены на рис. 2.4.14. Построить конечный автомат, распознающий язык L_1L_2. Рассмотреть следующие варианты задания совокупностей заключительных состояний исходных автоматов:

а) $F^1 = \{q_1, q_2\}$, $F^2 = \{q_2\}$;

б) $F^1 = \{q_0, q_2\}$, $F^2 = \{q_1\}$;

в) $F^1 = \{q_0\}$, $F^2 = \{q_0, q_2\}$.

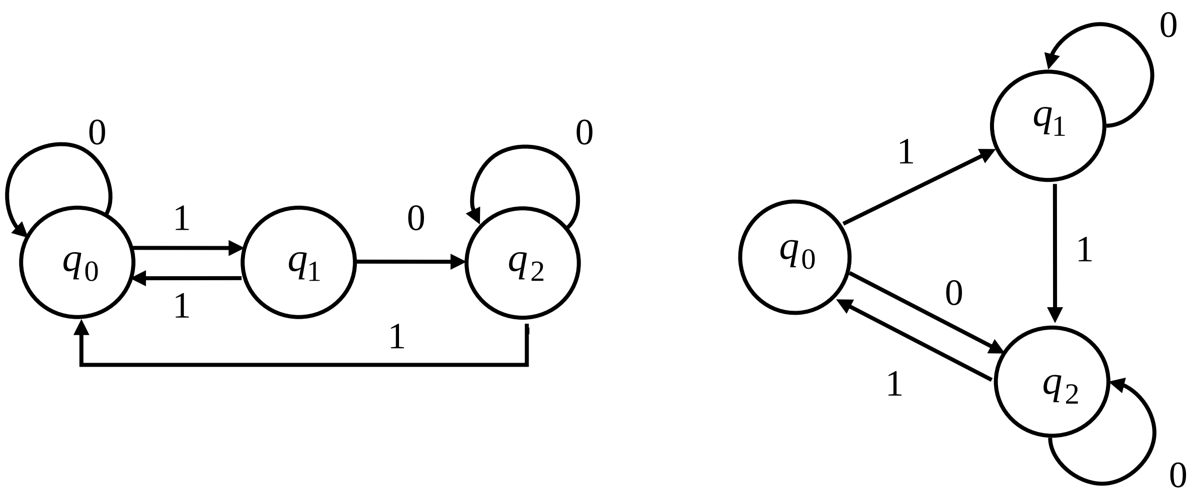


Рис. 2.4.14. Диаграммы автоматов, распознающих L_1 и L_2 к задаче 2.4.1

2.4.2. Языки L_1 и L_2 распознаются конечными автоматами, диаграммы переходов которых изображены на рис. 2.4.15. Построить конечный автомат, распознающий язык L_1L_2. Рассмотреть следующие варианты задания совокупностей заключительных состояний исходных автоматов:

а) $F^1 = \{q_1, q_2\}$, $F^2 = \{q_3\}$;

б) $F^1 = \{q_0, q_2\}$, $F^2 = \{q_1, q_3\}$;

в) $F^1 = \{q_0, q_1, q_2\}$, $F^2 = \{q_0, q_3\}$.

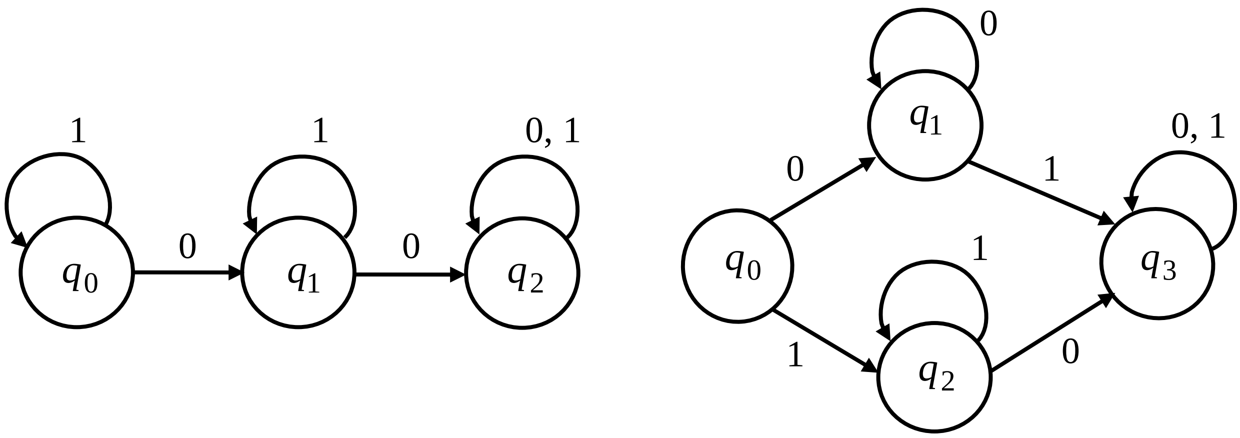


Рис. 2.4.15. Диаграммы автоматов, распознающих L_1 и L_2 к задаче 2.4.2

2.4.3. Языки L_1 и L_2 распознаются конечными автоматами, диаграммы переходов которых изображены на рис. 2.4.16. Построить конечный автомат, распознающий язык L_1L_2. Рассмотреть следующие варианты задания совокупностей заключительных состояний исходных автоматов:

а) $F^1 = \{q_1, q_3\}$, $F^2 = \{q_1\}$;

б) $F^1 = \{q_0, q_2\}$, $F^2 = \{q_1\}$;

в) $F^1 = \{q_0, q_1\}$, $F^2 = \{q_0\}$.

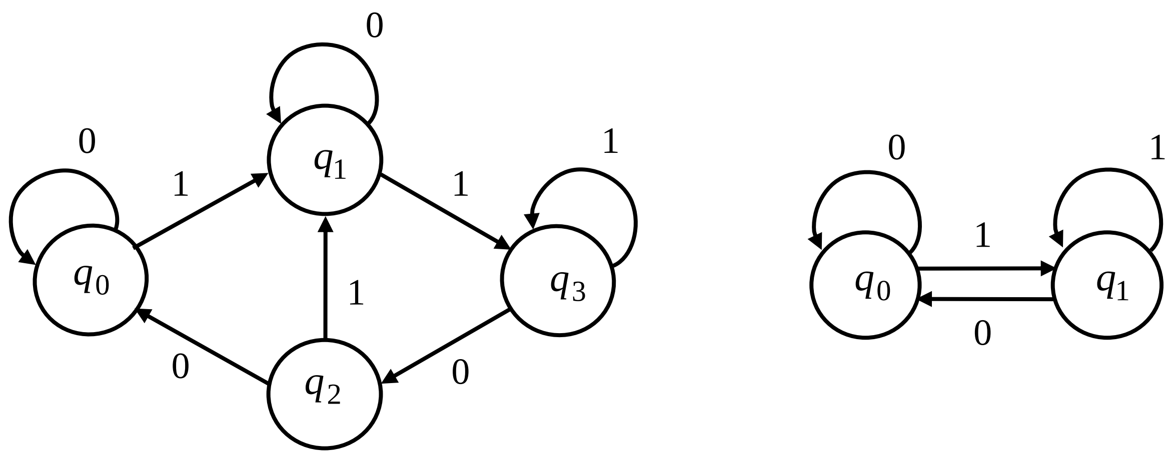


Рис. 2.4.16. Диаграммы автоматов, распознающих L_1 и L_2 к задаче 2.4.3

2.4.4. Языки L_1 и L_2 распознаются конечными автоматами, диаграммы переходов которых изображены на рис. 2.4.17. Построить конечный автомат, распознающий язык $L_1 L_2$. Рассмотреть следующие варианты задания совокупностей заключительных состояний исходных автоматов:

а) $F^1 = \{q_1, q_3\}$, $F^2 = \{q_2\}$;

б) $F^1 = \{q_0, q_2\}$, $F^2 = \{q_1\}$;

в) $F^1 = \{q_0, q_1\}$, $F^2 = \{q_0, q_2\}$.

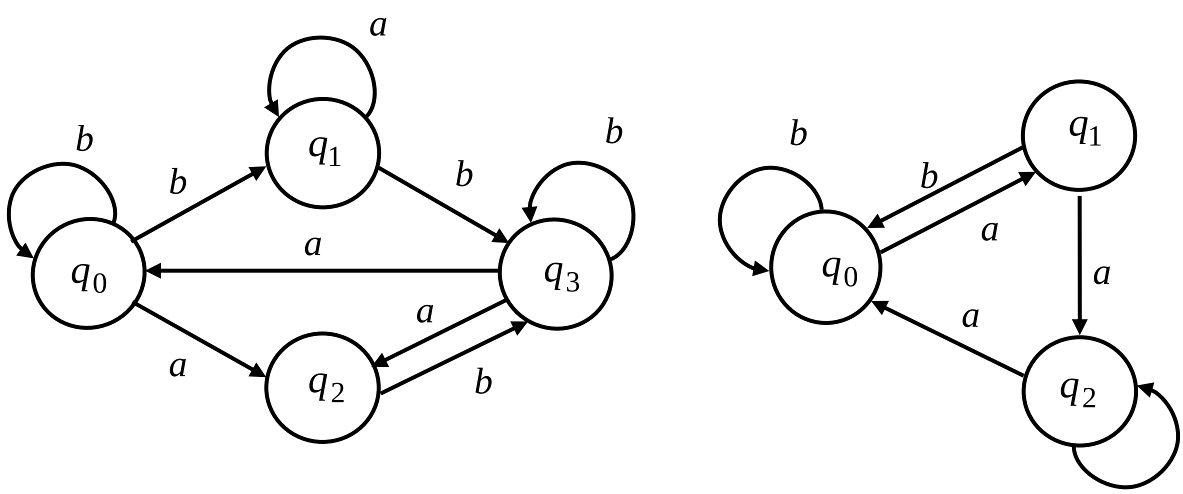
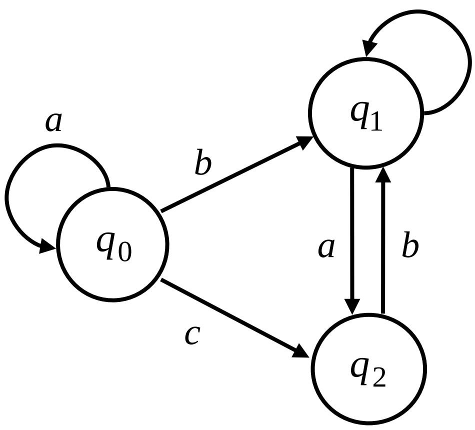


Рис. 2.4.17. Диаграммы автоматов, распознающих L_1 и L_2 к задаче 2.4.4

2.4.5. Язык L распознается конечным автоматом, диаграмма переходов которого изображена на рис. 2.4.18. Построить конечный автомат, распознающий язык L^3, если множество заключительных состояний автомата:

а) $F = \{q_1\}$;

б) $F = \{q_0, q_2\}$.



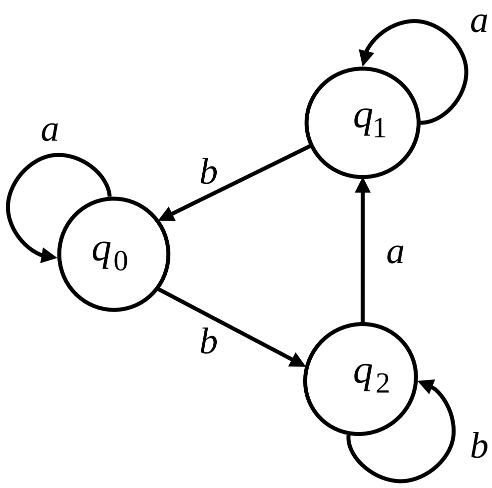


Рис. 2.4.18. Диаграмма к задаче 2.4.5

Рис. 2.4.19. Диаграмма к задаче 2.4.6

2.4.6. Язык L распознается конечным автоматом, диаграмма переходов которого изображена на рис. 2.4.19. Построить конечный автомат, распознающий язык L^3, если множество заключительных состояний автомата:

а) $F = \{q_1, q_2\}$;

б) $F = \{q_0\}$.

2.4.7. Язык L распознается конечным автоматом, диаграмма переходов которого изображена на рис. 2.4.20. Построить конечный автомат, распознающий итерацию этого языка, если множество заключительных состояний автомата:

а) $F = \{q_1, q_4\}$;

б) $F = \{q_0, q_2\}$;

в) $F = \{q_2, q_3\}$.

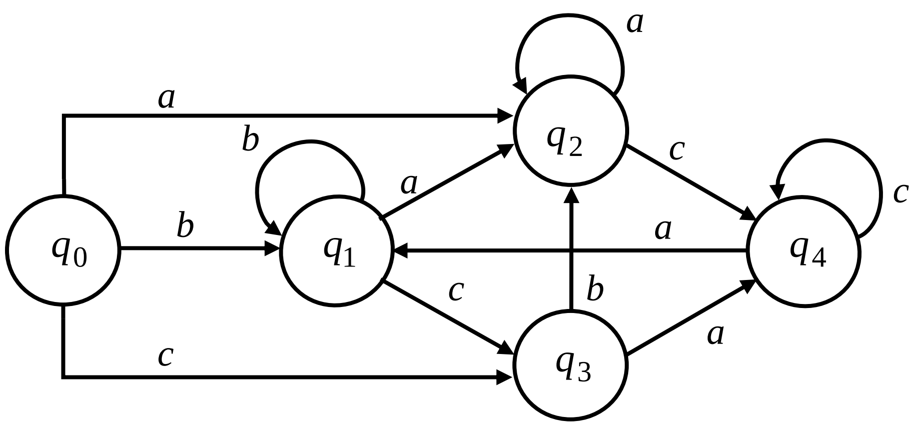


Рис. 2.4.20. Диаграмма переходов автомата к задаче 2.4.7

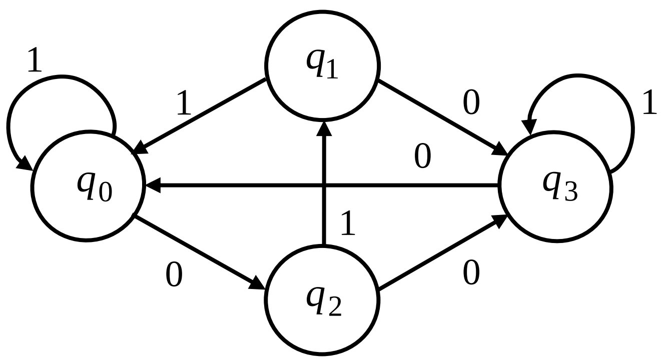


Рис. 2.4.21. Диаграмма переходов автомата к задаче 2.4.8

2.4.8. *Язык* L распознается конечным автоматом, диаграмма переходов которого изображена на рис. 2.4.21. Построить конечный автомат, распознающий итерацию этого языка, если множество заключительных состояний автомата:

а) $F = \{ q_1, q_3 \}$;

б) $F = \{ q_0, q_2 \}$;

в) $F = \{ q_2, q_3 \}$.

2.5. Лемма о разрастании для конечно-автоматных языков

Теорема 2.5.1. *Пусть* L *– конечно-автоматный язык. Существует такая константа p, что если* $w \in L$ *и* $|w| \geq p$*, то слово* w *можно записать в виде* xyz*, где* $0 < |y| \leq p$ *и* $xy^i z \in L$ *для всех* $i \geq 0$.

Лемма о разрастании полезна для доказательства того, что некоторые языки не являются конечно-автоматными.

Пример 2.5.1. Используя лемму, докажем, что язык $L = \left\{ a^n b a^n \mid n \geq 1 \right\}$ не является конечно-автоматным.

Доказательство проведем методом от противного. Предположим, что L − конечно-автоматный язык. Рассмотрим слово $w = a^p b a^p$, где p − константа из леммы. Очевидно, $|w| \geq p$. Поэтому w можно представить в виде $w = xyz$, где $|y| \geq 0$. Для y возможны следующие два случая:

а) $y = a^k$ для некоторого $k > 0$, для определенности предположим, что y входит в левую часть слова w. По лемме о разрастании слово $w' = xy^2 z$ также принадлежит языку L. Но в слове w' число вхождений буквы a слева больше числа вхождений a справа, следовательно, $w' \notin L$.

б) y содержит букву b. По лемме слово $w' = xz$ должно принадлежать L. Но в w' нет буквы b, поэтому $w' \notin L$.

В обоих случаях получили противоречие с предположением о том, что L − конечно-автоматный язык. Следовательно, язык L не является конечно-автоматным.

Задачи и упражнения

В задачах 2.5.1−2.5.14 доказать, что язык L не является конечно-автоматным:

2.5.1. $L=\left\{a^n b^n \mid n\ge 1\right\}$.

2.5.2. $L=\left\{a^n b^m \mid n,m\ge 1, n>m\right\}$.

2.5.3. $L=\left\{a^n b^m \mid n,m\ge 1, n\le m\right\}$.

2.5.4. $L=\left\{a^n b^m \mid n,m\ge 1, n\ne m\right\}$.

2.5.5. $L=\left\{a^n b^m c^k \mid n,m,k\ge 1, n\ne m\right\}$.

2.5.6. $L=\left\{a^n b^m c^k \mid n,m,k\ge 1, n\ne m, n\ne k, m\ne k\right\}$.

2.5.7. $L=\left\{a^n b^m c^k \mid n,m,k\ge 1, n\ge m\ge k\right\}$.

2.5.8. $L=\{\alpha \mid \alpha\in\{0,1\}^*$ и α содержит одинаковое число нулей и единиц$\}$.

2.5.9. $L=\{\alpha\alpha^R \mid \alpha\in\{0,1\}^*\}$.

2.5.10. $L=\{\alpha\alpha \mid \alpha\in\{0,1\}^*\}$.

2.5.11. $L=\left\{a^{n^2} \mid n\ge 1\right\}$.

2.5.12. $L=\left\{a^{n^3} \mid n\ge 1\right\}$.

2.5.13. $L=\{a^p \mid p -$ простое число$\}$.

2.5.14. $L=\{a^n \mid n=2^k, k=0,1,2,\dots\}$.

2.6. Минимизация конечного автомата

Пусть $M=\langle Q,A,q_0,\delta,F\rangle$ – конечный автомат. Конечный автомат M называется ***минимальным***, если не существует эквивалентного ему автомата с меньшим числом состояний.

Состояние q конечного автомата M называется недостижимым, если не существует слова α, переводящего автомат из начального состояния в состояние q.

Слово $\alpha\in A^*$ различает состояния q_1 и q_2, если $(q_1,\alpha)\vdash^*(q_3,\lambda)$, $(q_2,\alpha)\vdash^*(q_4,\lambda)$ и одно из состояний q_3 и q_4 принадлежит F, а другое нет.

Будем говорить, что состояния q_1 и q_2 $k-$неразличимы (и писать $q_1\equiv^k q_2$), если не существует слова $\alpha\in A^*$, различающего q_1 и q_2, для которого $|\alpha| \le k$. Будем говорить, что состояния q_1 и q_2 неразличимы (и писать $q_1 \equiv q_2$), если они $k-$неразличимы для любого $k \ge 0$.

Автомат M называется ***приведенным***, если в Q нет недостижимых состояний и нет двух неразличимых состояний.

Теорема 2.6.1. *Приведенный автомат является минимальным.*

Алгоритм построения минимального конечного автомата включает следующие этапы:

1. Поиск и удаление недостижимых состояний.
2. Построение классов эквивалентности по отношению $\equiv$.
3. Построение минимального автомата, состояниями которого являются классы эквивалентности.

Пример 2.6.1. Рассмотрим конечный автомат $M = \langle Q, A, q_0, \delta, F \rangle$, где $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$, $A = \{0,1\}$, $F = \{q_2, q_4\}$ и функция переходов δ определена таблицей 2.6.1.

Таблица 2.6.1

Таблица переходов автомата M из примера 2.6.1

Состояния	Вход	
	0	1
q_0	q_1	q_3
q_1	q_1	q_2
$^*q_2$	q_3	q_4
q_3	q_3	q_4
$^*q_4$	q_1	q_2
q_5	q_2	q_6
q_6	q_5	q_4

1) Состояния множества $\{q_5, q_6\}$ недостижимы из q_0, так как в них можно попасть только из того же множества $\{q_5, q_6\}$. После удаления состояний q_5 и q_6 получим таблицу 2.6.2. В общем случае недостижимые состояния могут быть найдены методом поиска в ширину.

2) Найдем классы эквивалентности отношения $\equiv$. Для этого последовательно будем строить классы эквивалентности отношения $\equiv^k$ для $k = 0,1,2,\ldots$, до тех пор, пока не выполнится условие $\equiv^k = \equiv^{k+1}$.

Таблица 2.6.2

Таблица переходов без недостижимых состояний

Состояния	Вход	
	0	1
q_0	q_1	q_3
q_1	q_1	q_2
$^*q_2$	q_3	q_4
q_3	q_3	q_4
$^*q_4$	q_1	q_2

При $k = 0$ всегда имеем два класса: F (класс 1) и $Q - F$ (класс 2).

При $k = 1$ построим таблицу 2.6.3, в которой для каждой буквы $a \in A$ и каждого состояния $q \in Q$ в столбце указан номер класса, которому принадлежит состояние $\delta(q,a)$.

Таблица 2.6.3

Определение классов эквивалентности при k=1

$a \in A$	1		2		
	q_2	q_4	q_0	q_1	q_3
0	2	2	2	2	2
1	1	1	2	1	1

По содержимому столбцов определяем, что класс 2 распадается на два новых класса эквивалентности. Поэтому для $k = 1$ получаем следующие классы эквивалентности: $\{q_2, q_4\}$ (класс 1), $\{q_0\}$ (класс 2), $\{q_1, q_3\}$ (класс 3).

Для $k = 2$ построим таблицу 2.6.4.

Таблица 2.6.4

Определение классов эквивалентности при k=2

$a \in A$	1		2	3	
	q_2	q_4	q_0	q_1	q_3
0	2	2	3	2	2
1	1	1	3	1	1

По таблице 2.6.4 устанавливаем, что новых классов эквивалентности не появилось, поэтому $\equiv^1 = \equiv^2 = \equiv$.

3) В минимальном автомате три состояния p, q и r, соответствующих классам эквивалентности: $\{q_0\}$, $\{q_1, q_3\}$, $\{q_2, q_4\}$. Начальное состояние –

$\{q_0\}$, заключительное состояние – $\{q_2, q_4\}$. Таблица переходов минимального автомата представлена в таблице 2.6.5.

Таблица 2.6.5

Таблица переходов минимального автомата

Состояния	Вход	
	0	1
p	q	q
q	q	r
$*r$	q	r

На рис.2.6.1 изображена диаграмма построенного минимального автомата.

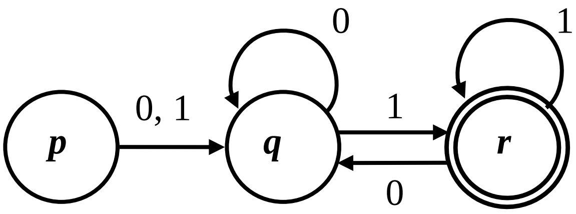


Рис.2.6.1. Диаграмма переходов минимального автомата

Задачи и упражнения

2.6.1. L – множество десятичных чисел над алфавитом {1,2,3}, кратных 6. В предположении, что пустое слово λ принадлежит L, построить минимальный автомат, допускающий L.

2.6.2. L – множество десятичных чисел над алфавитом {1,2,3}, кратных 6. В предположении, что $\lambda \notin L$, построить минимальный автомат, допускающий L.

2.6.3. L – множество десятичных чисел над алфавитом {0,1,2,3,4}, кратных 8. В предположении, что пустое слово λ принадлежит L, построить минимальный автомат, допускающий L.

2.6.4. L – множество десятичных чисел над алфавитом {1,2}, кратных 12. В предположении, что пустое слово λ принадлежит L, построить минимальный автомат, допускающий L.

2.6.5. L – множество десятичных чисел над алфавитом {1,2,3}, сумма цифр которых кратна 6. В предположении, что пустое слово λ принадлежит L, построить минимальный автомат, допускающий L.

2.6.6. L – множество десятичных чисел над алфавитом $\{1,2,3\}$, каждое из которых кратно 2 или кратно 3. В предположении, что пустое слово λ принадлежит L, построить минимальный автомат, допускающий L.

2.6.7. L – множество слов в алфавите $\{a,b\}$, длина которых кратна 6. Построить минимальный автомат, допускающий L.

2.6.8. L – множество слов в алфавите $\{a,b,c\}$, в которых число вхождений буквы a четно и число вхождений буквы b четно. Построить минимальный автомат, допускающий L.

2.6.9. L – множество слов в алфавите $\{a,b,c\}$, в которых число вхождений буквы a четно или число вхождений буквы b четно. Построить минимальный автомат, допускающий L.

2.6.10. Пусть $M = \langle Q, A, p, \delta, F \rangle$, где $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$, $A = \{0,1\}$, и функция переходов δ определена таблицей 2.6.6.

Таблица 2.6.6

Таблица переходов автомата M из задачи 2.6.10

Состояния	Вход	
	0	1
q_0	q_1	q_2
q_1	q_4	q_5
q_2	q_0	q_0
q_3	q_5	q_4
q_4	q_3	q_5
q_5	q_3	q_4

Построить минимальный автомат при следующих условиях:

а) $p = q_0, F = \{q_4, q_5\}$;

б) $p = q_1,\ F = \{q_4, q_5\}$;

в) $p = q_0,\ F = \{q_0\}$.

2.7. Регулярные языки; задача синтеза конечного автомата

Регулярное выражение в алфавите $A = \{a_1, ..., a_n\}$ определяется следующим образом:

1) символы $\varnothing$ и λ являются регулярными выражениями;

2) символ a_i, где $a_i \in A$ − любая буква алфавита A, является регулярным выражением;

3) если X и Y − регулярные выражения, тогда $(X \cup Y)$, $(X \cdot Y)$ и $(X)^*$ являются регулярными выражениями.

Регулярные выражения, записанные в соответствии с правилами 1-2, называются ***элементарными***, все остальные регулярные выражения − ***составными***.

Далее в записях регулярных выражений можно опускать внешние скобки и скобки, отсутствие которых не может привести к недоразумениям. Операция "звездочка" имеет самый высокий приоритет, далее по порядку приоритетности идет операция конкатенации, а затем − операция объединения. Вместо $X \cdot Y$ можно писать XY.

Каждое регулярное выражение R над алфавитом $A = \{ a_1, ..., a_n \}$ задает некоторый язык $L(R)$ над алфавитом A по следующим правилам:

1) $L(\varnothing) = \varnothing$ − пустой язык; $L(\lambda) = \{\lambda\}$ − язык, состоящий из одного пустого слова;

2) $L(a_i) = \{ a_i \}$ − язык, состоящий из одного однобуквенного слова a_i;

3) Если L_1 и L_2 − языки, определяемые регулярными выражениями X и Y, тогда $L(X \cup Y) = L_1 \cup L_2$, $L(X\,Y) = L_1\,L_2$ и $L(X^*) = L_1^*$.

Регулярный язык − язык, задаваемый регулярным выражением.

Пример 2.7.1. Пусть $A = \{a, b, c\}$ − алфавит, тогда регулярное выражение $(a \cup b \cup c)^*$ определяет множество всех слов в алфавите A.

Регулярное выражение $a(a \cup b \cup c)^* c$ определяет множество всех слов, начинающихся буквой a и заканчивающихся буквой c.

Регулярное выражение $(a \cup b \cup c)^* bb(a \cup b \cup c)^*$ определяет множество всех слов, содержащих хотя бы одно двукратное вхождение буквы b подряд.

Задача синтеза конечного автомата состоит в построении по произвольному регулярному выражению R конечного автомата $M(R)$.

Теорема 2.7.1. *Каждое регулярное выражение определяет конечно-автоматный язык.*

Действительно, каждое элементарное регулярное выражение ($\varnothing$, λ и a_i, где a_i − любая буква алфавита A) определяет конечно-автоматный язык; легко строятся конечные автоматы, распознающие эти языки. Составные регулярные выражения получаются из элементарных посредством применения операций объединения, конкатенации и итерации. Но каждая из этих операций не выводит за класс конечно-автоматных языков; алгоритмы синтеза конечных автоматов, распознающих получаемые посредством выполнения перечисленных операций языки, уже построены.

Задачи и упражнения

В задачах 2.7.1 − 2.7.22 требуется решить задачу синтеза конечного автомата по заданному регулярному выражению R.

2.7.1. $R=(a\cup b\cup c)^*$.

2.7.2. $R=a^*b^*c^*$.

2.7.3. $R=(aba\cup acba\cup bcba\cup ca)$.

2.7.4. $R=c(ab)^*ba^*c^*b$.

2.7.5. $R=a(ab^*c)^*a$.

2.7.6. $R=a(ab\cup bc)^*aa$.

2.7.7. $R=a^*(bc\cup ca)(ab\cup cb)^*$.

2.7.8. $R=a^*(bc\cup ca)\cup(ab\cup cb)^*$.

2.7.9. $R=(a\cup b\cup c)^*a(ab^*c)^*a$.

2.7.10. $R=(a\cup b\cup c)^*\cup a(ab^*c)^*a$.

2.7.11. $R=a(ab^*\cup acb)^*\cup b(cb^*)^*a$.

2.7.12. $R=a^*(bc\cup ca)(abc\cup cb)^*ac$.

2.7.13. $R=a(ab^*c)^*a\left((ca^*a\cup bc)^*\cup b(cb)^*a\right)$.

2.7.14. $R=(a^*bc\cup cd)^*(bac^*\cup aa^*c)^*$.

2.7.15. $R=(ba^*c\cup ca)^*\cup ab^*c^*$.

2.7.16. $R=\left((ba^*c\cup ca)^*\cup(a^*c)^*\right)^*$.

2.7.17. $R=(ba^*b^*c\cup cba^*)^*$.

2.7.18. $R=(ba^*c\cup a^*bc^*)^*\cup(ac^*\cup a^*c)^*$.

2.7.19. $R = \left(\left(abc^{*} \cup (ab)^{*}\right)^{*} ab\left(ab^{*} \cup b^{*}ca\right)^{*}\right)^{*}$.

2.7.20. $R = \left(\left(baa^{*}c \cup bd\right)^{*}\left(cd^{*}a \cup ab\right)^{*}\right)^{*}$.

2.7.21. $R = \left(\left((abc)^{*} ac\right)^{*} \cup bc^{*}ac^{*}b\right)^{*}$.

2.7.22. $R = \left(ca^{*}b \cup a^{*}bc^{*}\right)^{*} abc\left(bac \cup a \cup (cb)^{*}\right)^{*}$.

2.8. Задача анализа конечного автомата

Задача анализа конечного автомата состоит в построении регулярного выражения по заданному конечному автомату. Задачу анализа решают алгоритмы Мак-Ноттона – Ямады и В.М. Глушкова.

2.8.1. Алгоритм Мак-Ноттона – Ямады

Пусть L − конечно-автоматный язык и $M = \langle Q, A, q_0, \delta, F \rangle$ − допускающий его конечный автомат с $Q = \{q_0, q_1, q_2, \ldots q_m\}$. Определим язык L_{ij} следующим образом: $\alpha \in L_{ij}$ тогда и только тогда, когда $(q_i, \alpha) \vdash^{*} (q_j, \lambda)$. Через L_{ij}^{k} обозначим подмножество L_{ij} всех таких слов, при чтении которых в качестве промежуточных состояний могут встречаться только состояния из множества $\{q_0, q_1, q_2, \ldots q_k\}$. Очевидно, $L_{ij}^{m} = L$.

Построим регулярные выражения Z_{ij} $(i, j \in \{0,1,2,\ldots,m\})$ следующим образом. Если в диаграмме переходов автомата есть дуги из состояния q_i в состояние q_j, которым приписаны буквы $a_1,\ldots,a_r$, то при $i \neq j$ полагаем $Z_{ij} = a_1 \cup \ldots \cup a_r$, а если таких дуг нет, то $Z_{ij} = \varnothing$. При $i = j$ полагаем $Z_{ii} = a_1 \cup \ldots \cup a_r \cup \lambda$, если состоянию q_i соответствуют петли, которым приписаны буквы $a_1,\ldots,a_r$, и $Z_{ii} = \lambda$, если в состоянии q_i петли отсутствуют.

Через R_{ij}^{k} обозначим регулярное выражение, определяющее язык L_{ij}^{k}. Положим

$$R_{ij}^{0} = Z_{ij} \cup Z_{i0}\left(Z_{00}\right)^{*} Z_{0j} \text{ для } k = 0, \tag{2.8.1.1}$$

$$R_{ij}^{k} = R_{ij}^{k-1} \cup R_{ik}^{k-1}\left(R_{kk}^{k-1}\right)^{*} R_{kj}^{k-1} \text{ для } k \geq 1. \tag{2.8.1.2}$$

Язык $L = L(M)$ может быть представлен в виде $L(M) = \bigcup\limits_{j \in \Phi} L_{0j}$, где Φ – множество всех номеров состояний из F. Поэтому $L(M)$ определяется регулярным выражением $R(M) = \bigcup\limits_{j \in \Phi} R_{0j}^{m}$.

Пример 2.8.1.1. Решить задачу анализа для автомата M (см. рис. 2.8.1), используя алгоритм Мак-Ноттона – Ямады.

Поскольку M имеет множество состояний $Q = \{q_0, q_1\}$, начальное состояние q_0 и множество заключительных состояний $F = \{q_1\}$, справедливо равенство $L(M) = L_{01}^{1}$. Поэтому задача анализа состоит в построении регулярного выражения R_{01}^{1}.

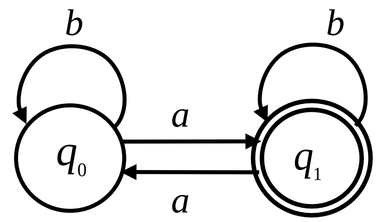


Рис.2.8.1. Диаграмма M для примера 2.8.1.1

На первом шаге построим выражения Z_{ij}:

$Z_{00} = b \cup \lambda$, так как есть единственная петля в состоянии q_0, и она помечена буквой b.

$Z_{01} = a$, так как есть единственная дуга из состояния q_0 в состояние q_1, и она помечена буквой a.

$Z_{10} = a$ по тем же причинам, что и Z_{01}.

$Z_{11} = b \cup \lambda$ по тем же причинам, что и Z_{00}.

На втором шаге построим выражения R_{ij}^{0} по формуле (2.8.1.1):

$$R_{00}^{0} = Z_{00} \cup Z_{00}\left(Z_{00}\right)^{*} Z_{00} = \left(Z_{00}\right)^{*} = \left(b \cup \lambda\right)^{*} = b^{*};$$

$$R_{01}^{0} = Z_{01} \cup Z_{00}\left(Z_{00}\right)^{*} Z_{01} = \left(Z_{00}\right)^{*} Z_{01} = \left(b \cup \lambda\right)^{*} a = b^{*} a;$$

$$R_{10}^{0} = Z_{10} \cup Z_{10}\left(Z_{00}\right)^{*} Z_{00} = Z_{10}\left(Z_{00}\right)^{*} = a\left(b \cup \lambda\right)^{*} = ab^{*};$$

$$R_{11}^{0} = Z_{11} \cup Z_{10}\left(Z_{00}\right)^{*} Z_{01} = b \cup \lambda \cup a\left(b \cup \lambda\right)^{*} a = \lambda \cup b \cup ab^{*} a.$$

На третьем шаге найдем R_{01}^{1} по формуле (2.8.1.2):

$$R_{01}^{1} = R_{01}^{0} \cup R_{01}^{0} \left(R_{11}^{0}\right)^{*} R_{11}^{0} = R_{01}^{0} \left(R_{11}^{0}\right)^{*} = b^{*}a \left(\lambda \cup b \cup ab^{*}a\right)^{*} = b^{*}a \left(b \cup ab^{*}a\right)^{*}.$$

Таким образом, язык, допускаемый автоматом M, описывается регулярным выражением $R(M) = R_{01}^{1} = b^{*}a\left(b \cup ab^{*}a\right)^{*}$.

Отметим большую вычислительную трудоемкость алгоритма Мак-Ноттона – Ямады.

2.8.2. *Алгоритм Глушкова*

Путем в конечном автомате $M = \langle Q, A, q_0, \delta, F \rangle$ назовем последовательность $\pi = q_{i_0} a_{j_1} q_{i_1} a_{j_2} \ldots q_{i_{k-1}} a_{j_k} q_{i_k}$, в которой $q_{i_0}, q_{i_1}, \ldots, q_{i_{k-1}}, q_{i_k}$ – состояния автомата M, $a_{j_1}, a_{j_2}, \ldots, a_{j_k}$ – буквы входного алфавита A, причем $\delta(q_{i_{s-1}}, a_{j_s}) = q_{i_s}$ для всех $s = 1,2,\ldots,k$. Путь называется ***простым***, если в нем все состояния различны, кроме, может быть, первого и последнего. Простой путь, в котором $q_{i_0} = q_{i_k}$, называется ***простым*** циклом. После удаления из простого цикла $\pi = q_x a_{j_1} q_{i_1} a_{j_2} \ldots q_{i_{k-1}} a_{j_k} q_x$ первого и последнего состояний получаем x***-цикл***. x-циклы называются также открытыми путями из q_x в q_x.

Начальным комплексом автомата M называется перечисленная с использованием знака $\cup$ совокупность всех простых путей, ведущих из начального состояния q_0 в состояния из множества $F - \{q_0\}$; при $q_0 \in F$ в начальный комплекс добавляется символ q_0 как отдельный путь.

Комплексом состояния q_i называется взятая в итерационные скобки совокупность всех i-циклов, соединенных с использованием знака $\cup$.

Начальный комплекс автомата M и комплекс произвольного состояния q_i будем обозначать $C_{нач}$ и C_i соответственно.

Пусть W – произвольное подмножество номеров состояний, $i \notin W$. ***Условным комплексом*** $C_i(W)$ состояния q_i называется взятая в итерационные скобки совокупность всех тех i-циклов, в записях которых нет состояний с номерами из W; i-циклы соединяются знаком $\cup$.

Если комплекс или условный комплекс состояния q_i не содержит ни одного i-цикла, то полагается $C_i = \lambda$ или, соответственно, $C_i(W) = \lambda$.

Введем ***операцию подстановки в пути начального комплекса***. Пусть $\pi = q_0 a_{j_1} q_{i_1} a_{j_2} q_{i_2} a_{j_3} q_{i_3} a_{j_4} \dots q_{i_{k-1}} a_{j_k} q_{i_k}$ – произвольный путь начального комплекса. Результатом операции подстановки является

$$P(\pi) = C_0 a_{j_1} C_{i_1}(0) a_{j_2} C_{i_2}(0, i_1) a_{j_3} C_{i_3}(0, i_1, i_2) a_{j_4} \dots$$
$$\dots C_{i_{k-1}}(0, i_1, i_2, \dots, i_{k-2}) a_{j_k} C_{i_k}(0, i_1, i_2, \dots, i_{k-1}).$$

Операция подстановки выполняется над всеми путями начального комплекса. Для вырожденного пути $\pi = q_0$ имеем $P(q_0) = C_0$.

Введем также ***операцию подстановки в открытые пути условных комплексов***. Пусть $\pi = a_{j_1} q_{i_1} a_{j_2} q_{i_2} a_{j_3} q_{i_3} \dots q_{i_{k-1}} a_{j_k}$ – произвольный открытый путь, входящий в $C_x(W)$. Результатом операции подстановки является

$$P(\pi) = a_{j_1} C_{i_1}(W \cup \{x\}) a_{j_2} C_{i_2}(W \cup \{x, i_1\}) a_{j_3} C_{i_3}(W \cup \{x, i_1, i_2\}) \dots$$
$$\dots C_{i_{k-1}}(W \cup \{x, i_1, i_2, \dots, i_{k-2}\}) a_{j_k}.$$

Операция подстановки выполняется над всеми открытыми путями условного комплекса. Комплекс состояния считаем частным случаем условного комплекса ($W = \varnothing$).

Под ***расширением*** понимаем результат замены каждого содержащегося в записи обозначения условного комплекса его значением.

Алгоритм анализа конечного автомата M состоит из двух этапов:

1) определение начального комплекса и комплекса всех состояний автомата;
2) последовательное построение ***записей-результатов***.

Запись-результат номер 0 – это запись начального комплекса.

Запись-результат номер 1 – результат применения операции подстановки в пути начального комплекса.

Запись-результат номер 2 (и каждая следующая четная запись) – результат применения к предыдущей записи операции расширения.

Запись-результат номер 3 (и каждая следующая нечетная запись) – результат применения операции подстановки к путям всех условных комплексов, содержащимся в предыдущей записи.

Процесс последовательного построения записей-результатов завершается, когда в последней полученной применением операции расширения записи отсутствуют символы состояний конечного автомата. Эта запись является искомым регулярным выражением.

Пример 2.8.2.1. Решить задачу анализа для автомата M (см. рис. 2.8.1), используя алгоритм Глушкова.

Начальный комплекс автомата M: $C_{нач} = q_0 a q_1$, так как существует единственный простой путь, ведущий из начального состояния q_0 в заключительное состояние q_1; этому пути соответствует входное слово a.

Комплекс состояния q_0: $C_0 = (b \cup a q_1 a)^*$, так как существуют два простых цикла, начинающихся и заканчивающихся в состоянии q_0; этим циклам соответствуют входные слова b и aa.

Комплекс состояния q_1: $C_1 = (b \cup a q_0 a)^*$ по тем же причинам.

Выпишем последовательность записей-результатов:

0) $q_0 a q_1$ − начальный комплекс;

1) $C_0 a C_1(0)$ − результат операции подстановки в запись с номером 0;

2) $(b \cup a q_1 a)^* a b^*$ − результат операции расширения, примененной к записи 1;

3) $(b \cup a C_1(0) a)^* a b^*$ − результат операции подстановки в запись с номером 2;

4) $\left(b \cup a b^* a\right)^* a b^*$ − результат операции расширения, примененной к записи 3.

Полученное на шаге 4 выражение не содержит знаков состояний автомата, поэтому оно является искомым. Язык, допускаемый автоматом M, описывается регулярным выражением $\left(b \cup a b^* a\right)^* a b^*$.

Заметим, что регулярные выражения для одного и того же автомата, полученные по алгоритмам Мак-Ноттона − Ямады и Глушкова, различаются, однако нетрудно убедиться, что они описывают один и тот же язык. Это подтверждает тот факт, что один и тот же язык может быть описан различными регулярными выражениями.

Пример 2.8.2.2. Решить задачу анализа для автомата M (см. рис. 2.8.2), используя алгоритм Глушкова.

Построим сначала начальный комплекс автомата: $C_{нач} = q_0 \cup q_0 a q_1$.

Заметим, что оба пути начального комплекса имеют общее начало, поэтому q_0 вынесем за скобки для сокращения длины записи, получим $C_{нач} = q_0(\lambda \cup a q_1)$.

Запишем теперь комплексы всех состояний рассматриваемого конечного автомата M:

$$C_0 = \left(b \cup c \cup aq_1a \cup aq_1bq_2a\right)^*;$$
$$C_1 = \left(c \cup aq_0a \cup bq_2b \cup bq_2aq_0a\right)^*;$$
$$C_2 = \left(c \cup bq_1b \cup aq_0aq_1b\right)^*.$$

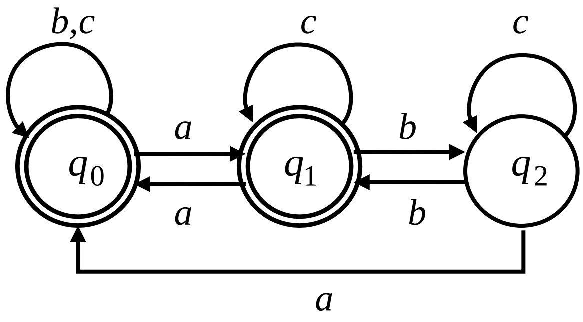


Рис. 2.8.2. Диаграмма M для примера 2.8.2.2

Последовательность записей-результатов имеет вид:

0) $q_0(\lambda \cup aq_1)$;

1) $C_0(\lambda \cup aC_1(0))$;

2) $\left(b \cup c \cup aq_1a \cup aq_1bq_2a\right)^*\left(\lambda \cup a\left(c \cup bq_2b\right)^*\right)$;

3) $\left(b \cup c \cup aC_1(0)a \cup aC_1(0)bC_2(0,1)a\right)^*\left(\lambda \cup a\left(c \cup bC_2(0,1)b\right)^*\right)$;

4) $\left(b \cup c \cup a\left(c \cup bq_2b\right)^* a \cup a\left(c \cup bq_2b\right)^* bc^*a\right)^*\left(\lambda \cup a\left(c \cup bc^*b\right)^*\right)$;

5) $\left(b \cup c \cup a\left(c \cup bC_2(0,1)b\right)^* a \cup a\left(c \cup bC_2(0,1)b\right)^* bc^*a\right)^* \cdot$

$\cdot\left(\lambda \cup a\left(c \cup bc^*b\right)^*\right)$;

6) $\left(b \cup c \cup a\left(c \cup bc^*b\right)^* a \cup a\left(c \cup bc^*b\right)^* bc^*a\right)^*\left(\lambda \cup a\left(c \cup bc^*b\right)^*\right) =$

$= \left(b \cup c \cup a\left(c \cup bc^*b\right)^*\left(a \cup bc^*a\right)\right)^*\left(\lambda \cup a\left(c \cup bc^*b\right)^*\right).$

Получили выражение, не содержащее символов состояний автомата, значит, это и есть искомое регулярное выражение, т.е.

$$R(M) = \left(b \cup c \cup a\left(c \cup bc^*b\right)^*\left(a \cup bc^*a\right)\right)^*\left(\lambda \cup a\left(c \cup bc^*b\right)^*\right).$$

Задачи и упражнения

2.8.1. По диаграмме конечного автомата M (см. рис. 2.8.3) построить регулярное выражение, используя алгоритм Мак-Ноттона – Ямады.

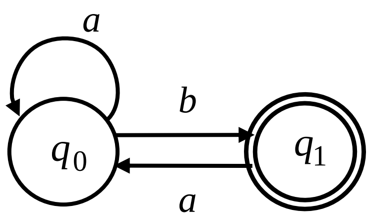


Рис. 2.8.3. Диаграмма автомата M к задаче 2.8.1

2.8.2. По диаграмме конечного автомата M (см. рис. 2.8.4) построить регулярные выражения с помощью алгоритма Мак-Ноттона – Ямады и алгоритма Глушкова.

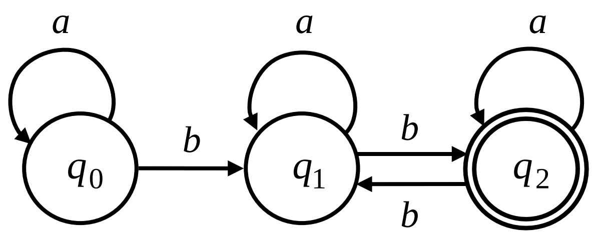


Рис. 2.8.4. Диаграмма автомата M к задаче 2.8.2

2.8.3. По диаграмме конечного автомата M (см. рис. 2.8.5) построить регулярные выражения с помощью алгоритма Мак-Ноттона – Ямады и алгоритма Глушкова.

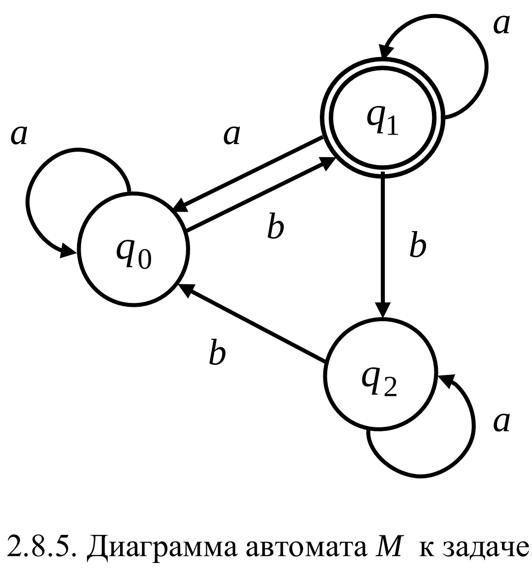


Рис. 2.8.5. Диаграмма автомата M к задаче 2.8.3

2.8.4. Используя алгоритм Глушкова, построить регулярное выражение по диаграмме конечного автомата M, изображенного на рис. 2.8.6.

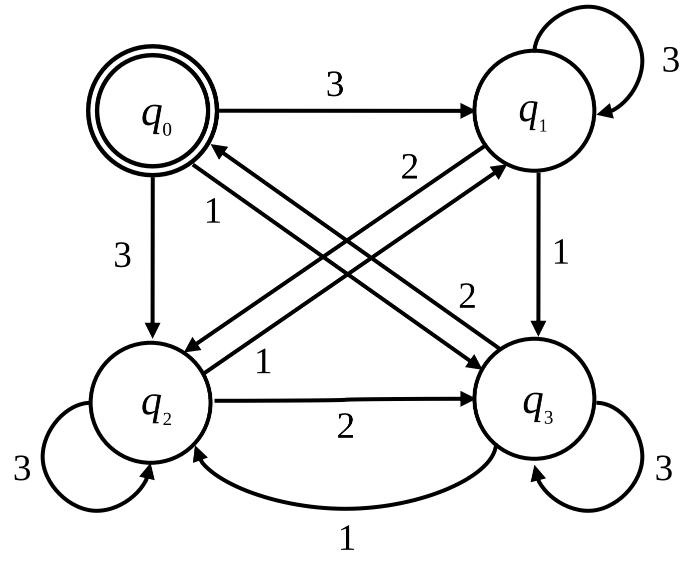


Рис. 2.8.6. Диаграмма автомата M к задаче 2.8.4

2.8.5. Используя алгоритм Глушкова, построить регулярное выражение по диаграмме конечного автомата на рис. 2.8.7.

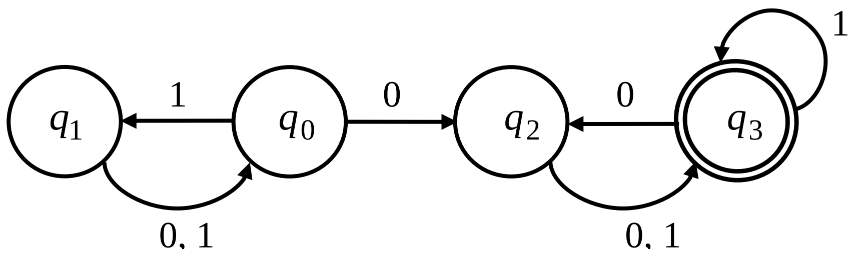


Рис. 2.8.7. Диаграмма автомата M к задаче 2.8.5

2.8.6. Пусть L − множество десятичных чисел над алфавитом {1,2,3}, кратных 6. В предположении, что пустое слово λ принадлежит L, построить регулярное выражение для языка L.

2.8.7. Пусть L − множество десятичных чисел над алфавитом {1,2,3}, кратных 6. В предположении, что пустое слово λ не принадлежит L, построить регулярное выражение для языка L.

2.8.8. Пусть L − множество десятичных чисел над алфавитом {0,1,2,3,4}, кратных 8. В предположении, что пустое слово λ принадлежит L, построить регулярное выражение для языка L.

2.8.9. Пусть L − множество десятичных чисел над алфавитом {1,2,3}, сумма цифр которых кратна 5. В предположении, что пустое слово λ принадлежит L, построить регулярное выражение для языка L.

2.8.10. Пусть L − множество десятичных чисел над алфавитом {1,2,3}, каждое из которых кратно 2 или кратно 3. В предположении, что пустое слово λ принадлежит L, построить регулярное выражение для языка L.

2.8.11. Пусть L − множество слов в алфавите $\{a,b\}$, длина которых кратна 6. Построить регулярное выражение для языка L.

2.8.12. Пусть L − множество слов в алфавите $\{a,b,c\}$, в которых число вхождений буквы a четно и число вхождений буквы b четно. Построить регулярное выражение для языка L.

2.9. Связь праволинейных грамматик и регулярных языков

2.9.1. Построение праволинейной грамматики по регулярному выражению

Пусть $G = \langle V_T, V_N, P, S \rangle$ − грамматика. Грамматика G называется ***праволинейной***, если каждое правило из P имеет вид $A \to xB$ или $A \to x$, где $A, B \in V_N$, $x \in V_T^{\,*}$.

Язык $L(G)$, порождаемый праволинейной грамматикой G, называется ***праволинейным***.

Например, грамматика $G = \langle V_T, V_N, P, S \rangle$, где $V_T = \{0,1\}$, $V_N = \{S\}$ и множество P содержит три правила: $S \to 0S \mid 1S \mid \lambda$, является праволинейной. Она порождает праволинейный язык $L = \{0,1\}^*$.

Лемма 2.9.1.1. *Пусть* V_T *−* *произвольный алфавит. Множества* $\varnothing$, $\{\lambda\}$, $\{a\}$ *для любого* $a \in V_T$ *являются праволинейными языками.*

1) $L(G) = \varnothing$ для грамматики $G = \langle V_T, \{S\}, \varnothing, S \rangle$.

2) $L(G) = \{\lambda\}$ для грамматики $G = \langle V_T, \{S\}, \{S \to \lambda\}, S \rangle$.

3) $L(G) = \{a\}$ для грамматики $G = \langle V_T, \{S\}, \{S \to a\}, S \rangle$.

Лемма 2.9.1.2. *Если* L_1 *и* L_2 *− праволинейные языки, то* $L_1 \cup L_2$, $L_1 L_2$ *и* L_1^* *- праволинейные языки.*

Пусть $G_1 = \left\langle V_T, V_N^1, P_1, S_1 \right\rangle$, $G_2 = \left\langle V_T, V_N^2, P_2, S_2 \right\rangle$, $V_N^1 \cap V_N^2 = \varnothing$.

1) Построим $G_3 = \left\langle V_T, V_N^1 \cup V_N^2 \cup \{S_3\}, P_1 \cup P_2 \cup \{S_3 \to S_1 \mid S_2\}, S_3 \right\rangle$, где $S_3 \notin V_N^1 \cup V_N^2$. $L(G_3) = L(G_1) \cup L(G_2)$.

2) Построим $G_4 = \left\langle V_T, V_N^1 \cup V_N^2, P_4, S_1 \right\rangle$, где P_4 определяется так:

а) если $A \to xB \in P_1$, то $A \to xB \in P_4$,

б) если $A \to x \in P_1$, то $A \to xS_2 \in P_4$,

в) $P_2 \subseteq P_4$.

$L(G_4) = L(G_1)L(G_2)$.

3) Построим $G_5 = \left\langle V_T, V_N^1 \cup \{S_5\}, P_5, S_5 \right\rangle$, $S_5 \notin V_N^1$, где P_5 строится так:

а) $P_1 \subseteq P_5$,

б) если $A \to x \in P_1$, то $A \to xS_5 \in P_5$,

в) $S_5 \to S_1 \mid \lambda \in P_5$.

$L(G_5) = \left(L(G_1)\right)^*$.

Следствием лемм 2.9.1.1 и 2.9.1.2 является

Теорема 2.9.1.1. *Любой регулярный язык является праволинейным.*

Для построения праволинейной грамматики по регулярному выражению достаточно определить порядок выполнения операций объединения, конкатенации и итерации в процессе построения регулярного выражения из базовых выражений вида $\varnothing, \lambda, a$.

Затем в соответствии с полученным порядком следует последовательно строить грамматики для базовых выражений и используемых операций, применяя правила построения грамматик, описанные в комментариях к леммам 2.9.1.1 и 2.9.1.2.

Пример 2.9.1.1. По регулярному выражению $R = b \cup ab^*$ построить праволинейную грамматику G.

Для построения грамматики G используем леммы 2.9.1.1 и 2.9.1.2 и пояснения к ним. Построение проведем в 5 этапов.

В качестве терминального алфавита на всех этапах возьмем $V_T = \{a, b\}$.

1) Построим грамматику G_1 для $R_1 = a$.

$G_1 = \langle V_T, \{S_1\}, \{S_1 \to a\}, S_1 \rangle$.

2) Построим грамматику G_2 для $R_2 = b$.

$G_2 = \langle V_T, \{S_2\}, \{S_2 \to b\}, S_2 \rangle$.

3) Построим грамматику G_3 для $R_3 = (R_2)^* = b^*$.

$G_3 = \langle V_T, \{S_3\}, P_3, S_3 \rangle$, где P_3 содержит правила $S_3 \to bS_3 \mid \lambda$.

Заметим, что по сравнению с общим правилом построения грамматики для итерации мы упростили грамматику за счет исключения лишних правил.

4) Построим грамматику G_4 для $R_4 = R_1 R_3 = ab^*$.

$G_4 = \langle V_T, \{S_1, S_3\}, P_4, S_1 \rangle$, где P_4 содержит правила $S_1 \to aS_3$, $S_3 \to bS_3 \mid \lambda$.

5) Построим грамматику G_5 для $R_5 = R_2 \cup R_4 = b \cup ab^*$.

$G_5 = \langle V_T, \{S_1, S_2, S_3, S_5\}, P_5, S_5 \rangle$, где P_5 содержит правила $S_2 \to b$, $S_1 \to aS_3$, $S_3 \to bS_3 \mid \lambda$, $S_5 \to S_2 \mid S_3$.

2.9.2. Построение праволинейной грамматики по конечному автомату

Существует простой способ построения грамматики по конечному автомату. Он состоит в следующем.

Пусть $M = \langle Q, A, q_0, \delta, F \rangle$ – конечный автомат.

Каждому состоянию $q_i \in Q$ ставится в соответствие свой нетерминал A_i. В качестве аксиомы грамматики выбирается A_0.

Для каждой дуги в диаграмме автомата строится свое правило. Если дуга ведет из состояния q_i в состояние q_j и помечена символом a, то в грамматике добавляется правило $A_i \to aA_j$. Для любого $q_i \in F$ вводится правило $A_i \to \lambda$.

Описанный способ может быть применен и для решения задачи перехода от регулярного выражения к грамматике. Для этого сначала следует решить задачу синтеза конечного автомата по заданному регулярному выражению, а затем по автомату построить грамматику.

Пример 2.9.2.1. По регулярному выражению $R = b \cup ab^*$ построить праволинейную грамматику G (см. пример 2.9.1.1).

Решая задачу синтеза конечного автомата, построим конечный автомат по выражению $R = b \cup ab^*$ (см. рис. 2.9.1).

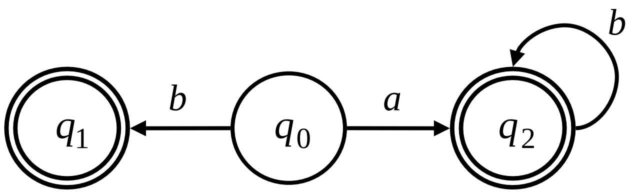


Рис. 2.9.1. Диаграмма автомата M для примера 2.9.2.1

По конечному автомату построим грамматику $G = \langle V_T, V_N, P, S \rangle$. Положим $V_T = \{a, b\}$, $V_N = \{A_0, A_1, A_2\}$, $S = A_0$. В диаграмме автомата есть три дуги, поэтому в P включаются три правила: $A_0 \to bA_1$, $A_0 \to aA_2$ и $A_2 \to bA_2$. Наконец, для заключительных состояний q_1 и q_2 добавляем правила $A_1 \to \lambda$ и $A_2 \to \lambda$.

2.9.3. Системы уравнений с регулярными коэффициентами

Рассмотрим уравнение

$$X = \alpha_0 \cup \alpha_1 X, \tag{2.9.3.1}$$

где коэффициенты α_0 и α_1 – регулярные выражения в алфавите A. Выделим два случая.

1) $\lambda \notin \alpha_1$. Тогда уравнение (2.9.3.1) имеет единственное решение $X = \alpha_1^* \alpha_0$.

2) $\lambda \in \alpha_1$. Можно проверить, что $X = \alpha_1^* (\alpha_0 \cup \gamma)$ является решением уравнения (2.9.3.1) при любом γ, не обязательно регулярном множестве. При этом $X = \alpha_1^* \alpha_0$ – минимальное решение, являющееся подмножеством любого другого решения.

Рассмотрим систему из двух уравнений с регулярными коэффициентами

$$\begin{cases} X_1 = \alpha_{10} \cup \alpha_{11} X_1 \cup \alpha_{12} X_2 \\ X_2 = \alpha_{20} \cup \alpha_{21} X_1 \cup \alpha_{22} X_2 \end{cases}. \tag{2.9.3.2}$$

Преобразуем первое уравнение из (2.9.3.2) к виду $X_1 = (\alpha_{10} \cup \alpha_{12} X_2) \cup \alpha_{11} X_1$ и найдем для X_1 минимальное решение, зависящее от X_2:

$$X_1 = \alpha_{11}^{*}(\alpha_{10} \cup \alpha_{12} X_2). \qquad (2.9.3.3)$$

Подставляя (2.9.3.3) во второе уравнение системы (2.9.3.2), получим:

$$X_2 = \alpha_{20} \cup \alpha_{21}\alpha_{11}^{*}(\alpha_{10} \cup \alpha_{12} X_2) \cup \alpha_{22} X_2 =$$

$$= \alpha_{20} \cup \alpha_{21}\alpha_{11}^{*}\alpha_{10} \cup \left(\alpha_{21}\alpha_{11}^{*}\alpha_{12} \cup \alpha_{22}\right) X_2 .$$

Найдем минимальное решение этого уравнения:

$$X_2 = \left(\alpha_{21}\alpha_{11}^{*}\alpha_{12} \cup \alpha_{22}\right)^{*} \left(\alpha_{20} \cup \alpha_{21}\alpha_{11}^{*}\alpha_{10}\right). \qquad (2.9.3.4)$$

Подставляя (2.9.3.4) в (2.9.3.3), получим:

$$X_1 = \alpha_{11}^{*}\left(\alpha_{10} \cup \alpha_{12}\left(\alpha_{21}\alpha_{11}^{*}\alpha_{12} \cup \alpha_{22}\right)^{*}\left(\alpha_{20} \cup \alpha_{21}\alpha_{11}^{*}\alpha_{10}\right)\right).$$

Минимальное решение системы (2.9.3.2) имеет вид

$$\begin{cases} X_1 = \alpha_{11}^{*}\left(\alpha_{10} \cup \alpha_{12}\left(\alpha_{21}\alpha_{11}^{*}\alpha_{12} \cup \alpha_{22}\right)^{*}\left(\alpha_{20} \cup \alpha_{21}\alpha_{11}^{*}\alpha_{10}\right)\right) \\ X_2 = \left(\alpha_{21}\alpha_{11}^{*}\alpha_{12} \cup \alpha_{22}\right)^{*}\left(\alpha_{20} \cup \alpha_{21}\alpha_{11}^{*}\alpha_{10}\right) \end{cases}.$$

Рассмотрим теперь систему из n уравнений:

$$X_i = \alpha_{i0} \cup \alpha_{i1} X_1 \cup \alpha_{i2} X_2 \cup \ldots \cup \alpha_{in} X_n, \text{ где } i = 1,2,\ldots,n \qquad (2.9.3.5)$$

и $\alpha_{i0}, \alpha_{i1}, \alpha_{i2}, \ldots \alpha_{in}$ – регулярные выражения в алфавите A.

Система с произвольным числом уравнений решается аналогично системе из двух уравнений методом последовательного исключения переменных. Приведем в общем виде алгоритм нахождения минимального решения.

Шаг 1. Положить $i = 1$.

Шаг 2. Если $i = n$, перейти к шагу 4. В противном случае с помощью тождеств записать уравнение для X_i в виде: $X_i = \alpha X_i \cup \beta$, где α – регулярное выражение в алфавите A и β – регулярное выражение в виде $\beta_0 \cup \beta_{i+1} X_{i+1} \cup \ldots \cup \beta_n X_n$. Затем в правых частях уравнений для $X_{i+1}, \ldots, X_n$ заменить X_i регулярным выражением $\alpha^* \beta$.

Шаг 3. Увеличить i на 1 и перейти к шагу 2.

Шаг 4. Записать уравнение для X_n в виде $X_n = \alpha X_n \cup \beta$, где $\alpha \cup \beta$ – регулярные выражения в алфавите A. Перейти к шагу 5.

Шаг 5. Уравнение для X_i имеет вид $X_i = \alpha X_i \cup \beta$, где $\alpha \cup \beta$ – регулярные выражения над A. Записать на выходе $X_i = \alpha^* \beta$, и в уравнения для $X_{i-1}, \ldots, X_1$ подставить $\alpha^* \beta$ вместо X_i.

Шаг 6. Если $i=1$, остановиться. В противном случае уменьшить i на 1 и перейти к шагу 5.

Пример 2.9.3.1. Решить систему уравнений

$$\begin{cases} X_1 = 1X_1 \cup 0X_2; \\ X_2 = \lambda \cup 0X_1 \cup 1X_2. \end{cases}$$

Из первого уравнения выразим X_1 через X_2:

$$X_1 = 1^*0X_2. \tag{2.9.3.6}$$

Подставив выражение для X_1 во второе уравнение, получим уравнение относительно X_2: $X_2 = \lambda \cup 01^*0X_2 \cup 1X_2$, или $X_2 = \lambda \cup \left(01^*0 \cup 1\right)X_2$. Решая это уравнение, найдем, что $X_2 = \left(01^*0 \cup 1\right)^* \lambda = \left(01^*0 \cup 1\right)^*$. Теперь подставим найденное выражение для X_2 в (2.9.3.6) и найдем X_1: $X_1 = 1^*0\left(01^*0 \cup 1\right)^*$. Таким образом, мы нашли следующее минимальное решение рассматриваемой системы уравнений:

$$\begin{cases} X_1 = 1^*0\left(01^*0 \cup 1\right)^*; \\ X_2 = \left(01^*0 \cup 1\right)^*. \end{cases}$$

2.9.4. Построение системы уравнений с регулярными коэффициентами по праволинейной грамматике

Пусть $G = \langle V_T, V_N, P, S \rangle$ – праволинейная грамматика, где $V_N = \{A_1, A_2, \ldots, A_n\}$. Будем полагать, что $S = A_1$.

Через G_i обозначим грамматику, полученную из исходной грамматики заменой аксиомы на нетерминал A_i, т.е. $G_i = \langle V_T, V_N, P, A_i \rangle$.

Положим $X_i = L(G_i)$. Через α_{i0} обозначим множество $\omega_1 \cup \omega_2 \cup \ldots \cup \omega_k$, где

$$A_i \to \omega_1 \mid \omega_2 \mid \ldots \mid \omega_k \tag{2.9.4.1}$$

– множество правил, в которых $\omega_i \in V_T{}^*$ (множество заключительных правил с левой частью A_i).

Положим $\alpha_{ij} = \omega_1 \cup \omega_2 \cup \ldots \cup \omega_m$, где

$$A_i \to \omega_1 A_j \mid \omega_2 A_j \mid \ldots \mid \omega_m A_j \tag{2.9.4.2}$$

– множество правил с левой частью A_i, правая часть которых содержит нетерминал A_j.

Построим систему уравнений

$$X_i = \alpha_{i0} \cup \alpha_{i1} X_1 \cup \ldots \cup \alpha_{in} X_n, \ i = 1, 2, \ldots, n. \qquad (2.9.4.3)$$

Нетрудно заметить, что α_{i0} – множество слов языка X_i, при выводе которых используется правило из (2.9.4.1), $\alpha_{ij} X_j$ – множество слов языка X_i, при выводе которых первым применено правило из (2.9.4.2).

Совокупность языков $(X_1, \ldots, X_n)$ является минимальным решением системы уравнений (2.9.4.3). Искомый язык $L(G) = X_1$. В минимальном решении X_i является регулярным языком для любого i, поэтому $L(G)$ – регулярный язык.

Теорема 2.9.4.1. *Любой праволинейный язык является регулярным.*

Из теорем 2.9.1.1 и 2.9.4.1 следует

Теорема 2.9.4.2. *Класс праволинейных языков совпадает с классом регулярных языков.*

Пример 2.9.4.1. Пусть $L \subseteq \{0,1,2\}^+$ - язык, в котором каждое слово, начинающееся с нуля, содержит не менее двух нулей. Язык L порождается грамматикой $G = \langle \{0,1,2\}, \{A, B, C\}, P, A \rangle$ с множеством правил P:

$$A \to 0B \mid 1C \mid 2C,$$
$$B \to 1B \mid 2B \mid 0C,$$
$$C \to 0C \mid 1C \mid 2C \mid \lambda.$$

Поставим в соответствие нетерминалу A переменную X_1, нетерминалу B – переменную X_2, нетерминалу C – переменную X_3.

Построим сначала уравнение для X_1. Так как для нетерминала A нет заключительных правил, полагаем $\alpha_{10} = \varnothing$. Для нетерминала A нет также правил вида $A \to \omega A,$ поэтому $\alpha_{11} = \varnothing$. В множестве правил для A есть правило $A \to 0B,$ поэтому $\alpha_{12} = 0$. Учитывая правила $A \to 1C \mid 2C$, полагаем $\alpha_{13} = (1 \cup 2)$. Таким образом, уравнение для X_1 имеет следующий вид: $X_1 = 0X_2 \cup (1 \cup 2)X_3$.

После построения уравнений для X_2 и X_3 получим следующую систему:

$$\begin{cases} X_1 = 0X_2 \cup (1 \cup 2)X_3; \\ X_2 = (1 \cup 2)X_2 \cup 0X_3; \\ X_3 = \lambda \cup (0 \cup 1 \cup 2)X_3. \end{cases}$$

Из третьего уравнения сразу находим $X_3 = (0 \cup 1 \cup 2)*$. Подставив значение X_3 в первые два уравнения, получим систему из двух уравнений относительно X_1 и X_2:

$$\begin{cases} X_1 = 0X_2 \cup (1 \cup 2)(0 \cup 1 \cup 2)^*; \\ X_2 = (1 \cup 2)X_2 \cup 0(0 \cup 1 \cup 2)^*. \end{cases}$$

Из второго уравнения найдем $X_2 = (1 \cup 2)^* 0(0 \cup 1 \cup 2)^*$.

Подставим значение X_2 в первое уравнение и получим $X_1 = 0(1 \cup 2)^* 0(0 \cup 1 \cup 2)^* \cup (1 \cup 2)(0 \cup 1 \cup 2)^*$. Это и есть регулярное выражение для языка $L(G)$.

Задачи и упражнения

2.9.1. Слово α принадлежит языку L тогда и только тогда, когда в этом слове буква a встречается четное число раз. Построить праволинейную грамматику, порождающую язык L.

2.9.2. Слово α принадлежит языку L тогда и только тогда, когда в этом слове буква a не встречается дважды подряд. Построить праволинейную грамматику, порождающую язык L.

2.9.3. Слово α принадлежит языку L тогда и только тогда, когда в нем содержится подслово bac. Построить праволинейную грамматику, порождающую язык L.

2.9.4. Слово α принадлежит языку L тогда и только тогда, когда в нем подслово ac встречается не более двух раз. Построить праволинейную грамматику, порождающую язык L.

2.9.5. Слово α принадлежит языку L тогда и только тогда, когда если оно содержит букву a, то содержит и букву b. Построить праволинейную грамматику, порождающую язык L.

2.9.6. Слово α принадлежит языку L тогда и только тогда, когда оно заканчивается на букву, ранее встречавшуюся в слове, причем $|\alpha| \geq 1$. Построить праволинейную грамматику, порождающую язык L.

2.9.7. Построить праволинейную грамматику, порождающую множество слов в алфавите $A=\{a,b\}$, длина которых кратна 3.

2.9.8. Пусть L − множество десятичных чисел над алфавитом {1,2,3}, каждое из которых кратно 2 или кратно 3. В предположении, что пустое слово λ принадлежит L, построить праволинейную грамматику для языка L.

2.9.9. По регулярному выражению построить праволинейную грамматику:

а) $R=(a\cup b\cup c)^*$.

б) $R=a^*b^*c$.

в) $R=aba\cup acba\cup bcba\cup ca$.

г) $R=c(ab)^*ba^*cb$.

д) $R=a\left(ab^*c\right)^*a$.

е) $R=a(ab\cup bc)^*aa$.

ж) $R=a^*(bc\cup ca)(ab\cup cb)^*$.

2.9.10. Решить уравнение $X=(a\cup b)X\cup a^*$.

2.9.11. Найти минимальное решение уравнения $X=(a\cup b)^*X\cup a$.

2.9.12. Найти минимальное решение уравнения $X=(a\cup b)^*X$. Существуют ли у этого уравнения другие решения, кроме минимального?

2.9.13. Решить систему уравнений $\begin{cases}X_1=bX_1\cup aX_2;\\ X_2=\lambda\cup bX_1.\end{cases}$

2.9.14. Решить систему уравнений $\begin{cases}X_1=bX_1\cup aX_2;\\ X_2=bX_1.\end{cases}$

2.9.15. Решить систему уравнений $\begin{cases}X_1=bX_1\cup aX_2;\\ X_2=a\cup bX_1\cup X_2.\end{cases}$

2.9.16. Решить систему уравнений $\begin{cases}X_1=\left(01^*\cup 1\right)X_1\cup X_2;\\ X_2=11\cup 1X_1\cup 00X_3;\\ X_3=\lambda\cup X_1\cup X_2.\end{cases}$

2.9.17. Решить систему уравнений $\begin{cases} X_1 = \lambda \cup 1X_1 \cup 0X_2; \\ X_2 = 1X_2 \cup 0X_3; \\ X_3 = 0X_1 \cup 1X_3. \end{cases}$

2.9.18. Пусть L − множество слов в алфавите $\{a,b,c\}$, каждое из которых содержит одну букву b и одну букву c. Построить праволинейную грамматику, порождающую язык L, по грамматике построить систему уравнений, и, решив ее, найти регулярное выражение для L.

2.9.19. Слово α принадлежит языку L тогда и только тогда, когда в нем буква a встречается нечетное число раз. Построить праволинейную грамматику, порождающую язык L, по грамматике построить систему уравнений, и, решив ее, найти регулярное выражение для L.

2.9.20. Слово α принадлежит языку L тогда и только тогда, когда в нем содержится подслово $abbc$. Построить праволинейную грамматику, порождающую язык L, по грамматике построить систему уравнений, и решив ее, найти регулярное выражение для L.

2.9.21. Слово α принадлежит языку L тогда и только тогда, когда в этом слове не содержится подслово cab. Построить праволинейную грамматику, порождающую язык L, по грамматике построить систему уравнений, и, решив ее, найти регулярное выражение для L.

2.9.22. Пусть L − множество слов в алфавите $A = \{a, b\}$, длина которых кратна 4. Построить праволинейную грамматику, порождающую язык L, по грамматике построить систему уравнений, и, решив ее, найти регулярное выражение для L.

Список литературы

1. Ахо У., Ульман Дж. ***Теория синтаксического анализа, перевода и компиляции.*** Т.1: Синтаксический анализ. М.: Мир, 1978. − 612 с.
2. Гилл А. ***Введение в теорию конечных автоматов***. М.: Радио и связь, 1987. − 272 с.
3. Гладкий А.В. ***Формальные грамматики и языки.*** М.: Наука, 1973. − 368 с.
4. Глушков В.М. ***Синтез цифровых автоматов***. М.: Физматгиз, 1962. − 478 с.
5. Коган Д.И., Бабкина Т.С. ***Основы теории конечных автоматов и регулярных языков***: Учебное пособие. − Нижний Новгород: Издательство Нижегородского университета, 2002. − 97 с.
6. Пентус А.Е., Пентус М.Р. ***Теория формальных языков***: Учебное пособие. − М.: Изд-во ЦПИ при механико-математическом факультете МГУ, 2004. − 80 с.
7. Хопкрофт Дж.Э., Мотвани Р., Ульман Дж.Д. ***Введение в теорию автоматов, языков и вычислений.*** 2-е изд. М.: Вильямс, 2002. − 528 с.

Оглавление