

# КОНЕЧНЫЕ АВТОМАТЫ

## (Лекции по дискретной математике)

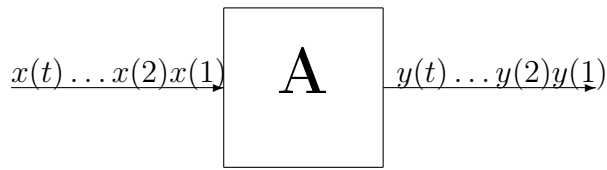
М.И. Дехтярь

### Содержание

<b>1</b>	<b>Переработка информации с помощью конечных автоматов</b>	<b>2</b>
1.1	Задачи . . . . .	5
<b>2</b>	<b>Конечные автоматы – распознаватели</b>	<b>5</b>
2.1	Детерминированные конечные автоматы (ДКА) и автоматные языки .	5
2.2	Произведение автоматов . . . . .	9
2.3	Задачи . . . . .	10
<b>3</b>	<b>Недетерминированные конечные автоматы и их детерминизация</b>	<b>11</b>
3.1	Задачи . . . . .	17
<b>4</b>	<b>Регулярные выражения и языки</b>	<b>17</b>
4.1	Задачи . . . . .	20
<b>5</b>	<b>Регулярные языки и конечные автоматы</b>	<b>21</b>
5.1	Автоматы для регулярных языков . . . . .	21
5.2	Свойства замкнутости класса автоматных языков . . . . .	25
5.3	Задачи . . . . .	30
<b>6</b>	<b>Теорема о разрастании автоматных языков. Неавтоматные языки</b>	<b>31</b>
6.1	Задачи . . . . .	34

## 1 Переработка информации с помощью конечных автоматов

Конечные автоматы являются математической моделью устройств, перерабатывающих дискретную входную информацию в режиме “реального времени”, т.е. в темпе ее поступления.



Автомат

На такие устройства в последовательные дискретные моменты времени  $1, 2, \dots, t, t+1, \dots$  поступают входные сигналы  $x(1), x(2), \dots, x(t), x(t+1), \dots$  и в ответ на них автомат **A** вырабатывает выходные сигналы  $y(1)y(2), \dots, y(t), y(t+1), \dots$ . Конечные автоматы характеризуются двумя особенностями.

1) *Отсутствие предвосхищения*: выходной сигнал  $y(t)$ , выдаваемый автоматом в момент  $t$ , зависит лишь от полученных к этому времени входов  $x(1), x(2), \dots, x(t)$ , т.е. автомат не может предвосхитить будущие входы и заранее на них отреагировать. Таким образом, имеется некоторая функция выходов  $\Psi(x(1), x(2), \dots, x(t)) = y(t)$ , определяющая очередной выход по предшествующему входу.

2) *Конечная память*: в каждый момент  $t$  информация в автомате о полученном к этому моменту входе  $x(1), x(2), \dots, x(t)$  конечна. Это свойство удобно интерпретировать следующим образом: автомат имеет конечное множество состояний  $Q$  и в каждый момент находится в одном из этих состояний. При получении очередного входа состояние может измениться. Таким образом, состояние  $q \in Q$ , в котором находится автомат после получения входной последовательности  $x(1), x(2), \dots, x(t)$ , и представляет информацию об этой последовательности, используемую в дальнейшей работе автомата при определении следующего состояния и выхода.

Наше обсуждение приводит к следующему определению конечного автомата с выходом.

**Определение 1.1. Конечный автомат - преобразователь** — это система вида

$$A = \langle \Sigma_X, \Sigma_Y, Q, q_0, \Phi, \Psi \rangle,$$

включающая следующие компоненты:

$\Sigma_X = \{a_1, \dots, a_m\}$  ( $m \geq 1$ ) — конечное множество - входной алфавит;

$\Sigma_Y = \{b_1, \dots, b_r\}$  ( $r \geq 1$ ) — конечное множество - выходной алфавит;

$Q = \{q_0, \dots, q_{n-1}\}$  ( $n \geq 1$ ) — конечное множество - алфавит внутренних состояний;

$q_0 \in Q$  — начальное состояние автомата;

$\Phi : Q \times \Sigma_X \rightarrow Q$  — функция переходов,  $\Phi(q, a)$  — это состояние, в которое переходит автомат из состояния  $q$ , когда получает на вход символ  $a$ ;

$\Psi : Q \times \Sigma_X \rightarrow \Sigma_Y$  — функция выхода,  $\Psi(q, a)$  — это символ из  $\Sigma_Y$ , который выдает на выход автомат в состоянии  $q$ , когда получает на вход символ  $a$ .

Иногда пару функций  $\Phi, \Psi$  называют *программой* автомата **A** и задают как спи-

сок из  $mn$  команд вида  $q_i a_j \rightarrow \Phi(q_i, a_j)/\Psi(q_i, a_j)$ .

Другой удобный способ задания функций  $\Phi$  и  $\Psi$  — табличный. Каждая из них определяется таблицей (матрицей) размера  $n \times m$ , строки которой соответствуют состояниям из  $Q$ , а столбцы — символам из входного алфавита  $\Sigma_X$ . В первой из них на пересечении строки  $q_i$  и столбца  $a_j$  стоит состояние  $\Phi(q_i, a_j)$ , а во второй — выходной символ  $\Psi(q_i, a_j)$ .

Еще один способ представления конечного автомата основан на использовании ориентированных размеченных графов.

**Определение 1.2.** Диаграмма автомата  $A = \langle \Sigma_X, \Sigma_Y, Q, q_0, \Phi, \Psi \rangle$  — это ориентированный (мульти) граф  $D_A = (Q, E)$  с помеченными ребрами, в котором выделена вершина-начальное состояние  $q_0$  и из каждой вершины  $q \in Q$  выходит  $|\Sigma_X|$  ребер, помеченных парами символов  $a/b$  ( $a \in \Sigma_X, b \in \Sigma_Y$ ). Таким образом, для каждой  $q \in Q$  и каждого символа  $a \in \Sigma_X$  имеется единственное ребро с меткой  $a/\Psi(q, a)$  из  $q$  в вершину  $q' = \Phi(q, a)$ .

Как автомат **A** перерабатывает входное слово  $x_1 x_2 \dots x_t$ ? Он начинает работу в состоянии  $q(0) = q_0$ . Затем, получив (прочитав) входной символ  $x_1$ , переходит в состояние  $q(1) = \Phi(q_0, x_1)$  и выдает символ  $y(1) = \Psi(q_0, x_1)$ . Далее, получив  $x_2$  **A** переходит в состояние  $q(2) = \Phi(q(1), x_2)$  и выдает символ  $y(2) = \Psi(q(1), x_2)$  и т.д. Таким образом, работа автомата, характеризуется последовательностью проходимых им состояний  $q(0), q(1), \dots, q(t), \dots$  и последовательностью выходных символов  $y(1), \dots, y(t), \dots$ . Они определяются следующими рекуррентными соотношениями:

$$\begin{aligned} q(0) &= q_0 \\ q(1) &= \Phi(q(0), x_1) \\ y(1) &= \Psi(q_0, x_1) \\ \dots \\ q(t+1) &= \Phi(q(t), x_{t+1}) \\ y(t+1) &= \Psi(q(t), x_{t+1}) \\ \dots \end{aligned}$$

Рассмотрим несколько примеров автоматов-преобразователей.

**Пример 1.1.** Сумматор последовательного действия

Мы уже строили схему из функциональных элементов  $SUM_n$ , реализующую для фиксированного  $n$  суммирование двух  $n$ -разрядных двоичных чисел. Построим теперь конечный автомат  $SUM$ , который сможет складывать два двоичных числа произвольной разрядности. На вход этого автомата будут последовательно подаваться пары  $x(i) = (x_1(i), x_2(i))$  соответствующих  $i$ -ых ( $1 \leq i \leq r$ ) разрядов двух двоичных

чисел  $x_1 = x_1(r) \dots x_1(2)x_1(1)$  и  $x_2 = x_2(r) \dots x_2(2)x_2(1)$ , а признаком завершения чисел будет служить символ  $x(r+1) = *$  (если одно из слагаемых короче другого, то будем считать, что недостающие разряды — нули). Выходом автомата должна быть последовательность  $(r+1)$  двоичных разрядов суммы  $y = x_1 + x_2$ :

$$\begin{array}{rcccc} & x_1(r) & \dots & x_1(2) & x_1(1) \\ + & x_2(r) & \dots & x_2(2) & x_2(1) \\ \hline y(r+1) & y(r) & \dots & y(2) & y(1) \end{array}$$

Таким образом, входной алфавит автомата:  $\Sigma_X = \{(00), (01), (10), (11), *\}$ , а выходной алфавит:  $\Sigma_Y = \{0, 1\}$ . Что нужно знать автомату  $SUM$ , о первых  $i$  разрядах  $x_1$  и  $x_2$ , чтобы получив их  $(i+1)$ -ые разряды  $(x_1(i+1), x_2(i+1))$ , верно определить выход  $y(i+1)$ ? Ясно, что для этого достаточно знать был ли перенос в  $i$ -ый разряд. Поэтому можно зафиксировать множество состояний  $Q = \{q_0, q_1\}$ , в котором  $q_0$  означает, что переноса не было, а  $q_1$ , что перенос был.

Теперь легко построить таблицы, представляющие функции переходов и выходов автомата  $SUM$ .

$\Phi :$	$Q \setminus \Sigma_X$	(00)	(01)	(10)	(11)	*
	$q_0$	$q_0$	$q_0$	$q_0$	$q_1$	$q_0$
	$q_1$	$q_0$	$q_1$	$q_1$	$q_1$	$q_0$
$\Psi :$	$Q \setminus \Sigma_X$	(00)	(01)	(10)	(11)	*
	$q_0$	0	1	1	0	0
	$q_1$	1	0	0	1	1

Заметим, что после получения символа  $*$  автомат  $SUM$  переходит в начальное состояние  $q_0$  и готов выполнять сложение следующей пары чисел. На диаграмме автомата

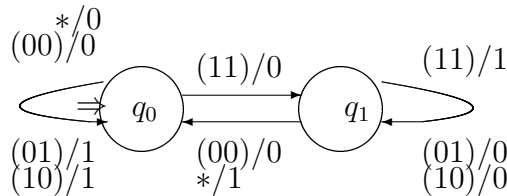


Рис. 1: Диаграмма автомата  $SUM$

у вершины  $q_0$  четыре петли, а у вершины  $q_1$  — три, объединены в одну с четырьмя и тремя метками, соответственно. Точно так же слиты два ребра из  $q_1$  в  $q_0$ . Стрелкой указано начальное состояние.

## 1.1 Задачи

**Задача 1.1.** Постройте конечный автомат, реализующий вычитание двоичных чисел.

**Задача 1.2.** Автомат по продаже кофе имеет щель для получения монет, кнопку, нажатие которой после уплаты достаточной суммы, приводит к получению кофе, и накопитель, через который он выдает сдачу покупателю. Автомат принимает монеты достоинством в 1, 2 и 5 рублей. Чашка кофе стоит 8 руб. Пока полученная сумма недостаточна горит красная лампочка. Если сумма, полученная автоматом,  $\geq 8$ , то загорается зеленая лампочка и после нажатия кнопки автомат наливает кофе и, если требуется, дает сдачу. Если автомат получает монету, когда горит зеленая лампочка, то он немедленно ее возвращает. Определите входной и выходной алфавиты конечного автомата, управляющего продажей кофе, и постройте его функции переходов и выходов.

**Задача 1.3.** Электронные часы имеют табло с указанием часов, минут и секунд и две управляющие кнопки. Одна кнопка переводит часы из нормального режима в режим настройки времени — вначале в настройку часов, затем — минут, затем — секунд, а затем возвращает в нормальный режим. Другая кнопка в нормальном режиме ничего не меняет, а в режиме настройки нажатие на нее увеличивает на единицу число настраиваемых часов, минут или секунд. Постройте автомат, который принимает на вход сигналы нажатия от двух кнопок, а на выходе выдает сигналы изменения режима и увеличения соответствующего числа.

## 2 Конечные автоматы — распознаватели

### 2.1 Детерминированные конечные автоматы (ДКА) и автоматные языки

Конечные автоматы часто используются для определения тех или иных свойств слов, т.е. для распознавания языков: автомат, распознающий некоторый язык  $L$  должен по произвольному слову  $w$  ответить на вопрос “ $w \in L$ ?”. Для решения такой задачи функция выходов может быть заменена на проверку того, в какое состояние переходит автомат после получения входного слова  $w$  — “принимющее” или “отвергающее”.

**Определение 2.1.** Детерминированный конечный автомат (ДКА) - распознаватель — это система вида

$$A = \langle \Sigma, Q, q_0, F, \Phi, \rangle,$$

включающая следующие компоненты:

$\Sigma = \{a_1, \dots, a_m\}$  ( $m \geq 1$ ) — конечное множество - входной алфавит;

$Q = \{q_0, \dots, q_{n-1}\}$  ( $n \geq 1$ ) — конечное множество - алфавит внутренних состояний;

$q_0 \in Q$  — начальное состояние автомата;

$F \subseteq Q$  — множество принимающих (допускающих, заключительных) состояний;

$\Phi : Q \times \Sigma \rightarrow Q$  — функция переходов,  $\Phi(q, a)$  — это состояние, в которое переходит автомат из состояния  $q$ , когда получает на вход символ  $a$ .

Функцию  $\Phi$  называют программой автомата  $A$  и задают как список из  $mn$  команд вида  $q_i a_j \rightarrow \Phi(q_i, a_j)$ .

Удобно также задавать функцию  $\Phi$  с помощью описанной выше таблицы. размера  $n \times m$ , строки которой соответствуют состояниям из  $Q$ , а столбцы — символам из входного алфавита  $\Sigma$ , в которой на пересечении строки  $q_i$  и столбца  $a_j$  стоит состояние  $\Phi(q_i, a_j)$ .

Как и автоматы-преобразователи, автоматы-распознаватели можно представлять с помощью размеченных ориентированных графов, называемых диаграммами.

**Определение 2.2.** Диаграмма ДКА  $A = \langle \Sigma, Q, q_0, \Phi \rangle$  — это ориентированный (мульти)граф  $D_A = (Q, E)$  с помеченными ребрами, в котором выделена вершина-начальное состояние  $q_0$  из каждой вершины  $q \in Q$  выходит  $|\Sigma_X|$  ребер, помеченных символами  $a \in \Sigma$  так, что для каждой  $q \in Q$  и каждого символа  $a \in \Sigma$  имеется единственное ребро из  $q$  в вершину  $q' = \Phi(q, a)$  с меткой  $a$ .

Скажем, что представленный последовательностью ребер путь  $p = e_1 e_2 \dots e_t$  в диаграмме несет слово  $w = w_1 w_2 \dots w_t$ , если  $w_i$  — это метка ребра  $e_i$  ( $1 \leq i \leq t$ ). Если  $q$  — начальная вершина (состояние) этого пути, а  $q'$  — его заключительная вершина, то будем говорить, что слово  $w$  переводит  $q$  в  $q'$ .

Работа конечного автомата-распознавателя состоит в чтении входного слова и изменению состояний в зависимости от его символов.

**Определение 2.3.** Назовем конфигурацией ДКА  $A = \langle \Sigma, Q, q_0, F, \Phi \rangle$  произвольную пару вида  $(q, w)$ , в которой  $q \in Q$  и  $w \in \Sigma^*$  (напомним, что через  $\Sigma^*$  обозначено множество всех слов в алфавите  $\Sigma$ , включающее и пустое слово  $\varepsilon$ ).

На множестве конфигураций введем отношение перехода за один шаг  $\vdash_A$ :

$$(q, w) \vdash_A (q', w') \Leftrightarrow w = aw', \Phi(q, a) = q'.$$

Если  $w = \varepsilon$ , то положим для каждого  $q \in Q$ :  $(q, \varepsilon) \vdash_A (q, \varepsilon)$ .

Через  $\vdash_A^*$  обозначим рефлексивное и транзитивное замыкание  $\vdash_A$ .

Содержательно,  $(q, w) \vdash_A^* (q', w')$  означает, что автомат  $A$ , начав работу в состоянии  $q$  на слове  $w = w_1 \dots w_l$  через некоторое конечное число шагов  $0 \leq k \leq l$  прочтет первые  $k$  символов слова  $w$  и перейдет в состояние  $q'$ , а  $w' = w_{k+1} \dots w_l$  — это непрочтенный остаток слова  $w$ .

**Определение 2.4.** ДКА  $A$  распознает (допускает, принимает) слово  $w$ , если для некоторого  $q \in F$

$(q_0, w) \vdash_A^* (q, \varepsilon)$ , т.е. после обработки слова  $w$  автомат переходит в принимающее состояние.

Язык  $L_A$ , распознаваемый (допускаемый, принимаемый) автоматом  $A$ , состоит из всех слов, распознаваемых этим автоматом:

$$L_A = \{w \mid A \text{ распознает } w\}.$$

Язык называется **конечно автоматным**, если он распознается некоторым ДКА.

Из этого определения, в частности, следует, что  $\varepsilon \in L_A \Leftrightarrow q_0 \in F$ .

Один и тот же язык может распознаваться разными автоматами.

**Определение 2.5.** Автоматы  $A$  и  $B$  называются эквивалентными, если совпадают распознаваемые ими языки, т.е.  $L_A = L_B$ .

Определение распознавания слова и языка можно легко перевести на язык диаграмм.

**Лемма 2.1.** Автомат  $A$  распознает (допускает, принимает) слово  $w$ , если для некоторого  $q \in F$  в диаграмме  $D_A$  имеется путь из  $q_0$  в  $q$ , который несет слово  $w$ , т.е.  $w$  переводит  $q_0$  в заключительное состояние  $q$ .

*Доказательство* можно провести индукцией по длине слова  $w$  (см. задачу 2.1 на стр. 10).

Таким образом, язык  $L_A$ , распознаваемый автоматом  $A$ , состоит из всех слов, которые переводят в его диаграмме  $D_A$  начальное состояние  $q_0$  в заключительные состояния из  $F$ .

Наша цель теперь состоит в изучении класса конечно автоматных языков.

Во многих случаях удастся доказать, что язык  $L$  конечно автоматный, непосредственно построив распознающий его автомат. Для этого нужно постараться разбить множество всех входных слов на конечное число классов “однородных”, “эквивалентных” слов, т.е. слов, получение которых на входе одинаково влияет на возможность их продолжения до слов распознаваемого языка. Затем для каждого такого класса создать состояние автомата и определить переходы между этими состояниями. Часто полезно бывает выделить одно состояние для представления “ошибочных” слов, для которых ни они сами, ни любые их продолжения не входят в язык.

**Пример 2.1.** Рассмотрим язык  $L$ , состоящий из всех слов в алфавите  $\Sigma = \{a, b\}$ , которые начинаются на  $aa$  и содержат нечетное число символов  $b$ .

Для выделения слов, начинающихся на  $aa$  создадим начальное состояние  $q_0$ , которое первый символ  $a$  будет переводить в состояние  $q_1$ , а второй символ  $a$  будет переводить  $q_1$  в состояние  $q_2$ . Ясно, что все слова, которые начинаются на  $ab, ba, bb$  сами не входят в язык  $L$  и все их продолжения также ошибочны. Заведем для них “ошибочное” состояние  $q_1$ . Остальные слова естественно разбиваются на два класса: те, в которых четное число символов  $b$ , и те, в которых число таких символов

нечетно (они и принадлежат  $L$ ). Так как после получения  $aa$  число  $b$  четно, то для представления слов первого класса будем использовать состояние  $q_2$ , а для представления слов второго — создадим состояние  $q_3$ , которое и будет заключительным. В результате получаем автомат, диаграмма которого представлена на следующем рисунке. (Мы будем далее на рисунках отмечать начальное состояние стрелкой  $\Rightarrow$ , а заключительные состояния — двумя окружностями).

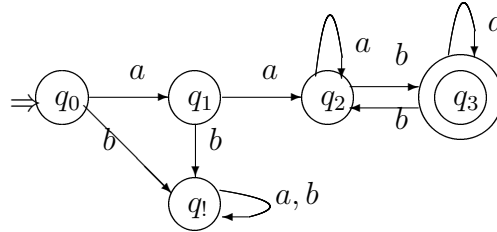


Рис. 2: Диаграмма автомата  $A$

Проверим работу этого автомата, например, на входном слове  $w = aaababa$ . При его чтении порождается следующая последовательность конфигураций:

$(q_0, aaababa) \vdash_A (q_1, aababa) \vdash_A (q_2, ababa) \vdash_A (q_2, baba) \vdash_A (q_3, aba) \vdash_A (q_3, ba) \vdash_A (q_2, a) \vdash_A (q_2, \varepsilon)$ .

Заключительное состояние этого вычисления  $q_2$  не является заключительным. Следовательно,  $w \notin L_A$ . Если же мы рассмотрим в качестве входа слово  $w_1 = wb = aaababab$ , то, продолжив на один шаг приведенное выше вычисление, получим, что  $(q_0, w_1) \vdash_A^* (q_3, \varepsilon)$ . Следовательно,  $w_1 \in L_A$ .

Мы проверили, что на двух входах автомат  $A$  работает верно. Как установить, что он построен корректно, т.е. верно работает на всех входных словах и распознает  $L$ ? Типичная схема доказательства правильности конечного автомата такова:

- 1) определить (описать) для каждого состояния  $q \in Q$  язык  $L(q)$ , состоящий из слов, переводящих начальное состояние  $q_0$  в  $q$ ;
- 2) доказать, что это определение правильное, используя индукцию по длине входного слова;
- 3) показать, что  $L = \cup_{q \in F} L(q)$ .

Применим эту схему к доказательству правильности, построенного выше автомата  $A$ . Языки, связанные с состояниями этого автомата, фактически, уже были определены при его построении. Уточним их:

$$L(q_0) = \{\varepsilon\},$$

$$L(q_1) = \{a\},$$

$$L(q_2) = \{w \mid \text{слова, начинающиеся с } aa \text{ и содержащие четное число букв } b\},$$



$$L(q_3) = L,$$

$$L(q_1) = \{w \mid \text{слова, не начинающиеся с } aa\}.$$

Правильность определения языков  $L(q_0)$ ,  $L(q_1)$  и  $L(q_2)$  следует непосредственно из определения  $A$ . Самое короткое слово, переводящее  $q_0$  в  $q_2$  —  $aa$ , и оно принадлежит  $L(q_2)$ . Аналогично, самое короткое слово, переводящее  $q_0$  в  $q_3$  —  $aab$ , и оно принадлежит  $L(q_3)$ . Предположим теперь, что для каждого слова  $w$  длины  $\leq n$  выполнено условие (\*):

$$w \text{ переводит начальное состояние } q_0 \text{ в } q_i \ (i = 2, 3) \Leftrightarrow w \in L(q_i).$$

Покажем, что оно будет выполнено и для всех слов длины  $n + 1$ .

Пусть  $|w| = n + 1$ . Тогда  $w = w'\alpha$ , где  $\alpha \in \{a, b\}$ . Так как  $|w'| = n$ , то для  $w'$  выполнено условие (\*). Поэтому, если  $w'$  переводит  $q_0$  в  $q_2$ , то это слово начинается с  $aa$  и содержит четное число  $b$ . При  $\alpha = a$  слово  $w$  переводит  $q_0$  в  $q_2$  и также начинается с  $aa$  и содержит четное число  $b$ , а при  $\alpha = b$  слово  $w$  переводит  $q_0$  в  $q_3$ , начинается с  $aa$  и содержит нечетное число  $b$ , т.е. принадлежит  $L$ .

Аналогично, если  $w'$  переводит  $q_0$  в  $q_3$ , то это слово начинается с  $aa$  и содержит нечетное число  $b$ . При  $\alpha = a$  слово  $w$  также переводит  $q_0$  в  $q_3$  и также начинается с  $aa$  и содержит нечетное число  $b$ , а при  $\alpha = b$   $w$  переводит  $q_0$  в  $q_2$ , оно начинается с  $aa$  и содержит четное число  $b$ . Обратно, если  $\alpha = a$ , то слово  $w$  переводит  $q_0$  в  $q_i$  ( $i = 2, 3$ )  $\Leftrightarrow w'$  переводит  $q_0$  в  $q_i$  ( $i = 2, 3$ ) и условие (\*) выполнено, так как четность числа букв  $b$  в  $w$  и в  $w'$  одинакова. Если же  $\alpha = b$ , то из определения автомата  $A$  следует, что слово  $w$  переводит  $q_0$  в  $q_2 \Leftrightarrow w'$  переводит  $q_0$  в  $q_3$  и  $w$  переводит  $q_0$  в  $q_3 \Leftrightarrow w'$  переводит  $q_0$  в  $q_2$ . Так как четность числа букв  $b$  в  $w$  и в  $w'$  разная, то и в этом случае условие (\*) выполнено. Для завершения доказательства осталось заметить, что единственным заключительным состоянием автомата  $A$  является  $q_3$  и поэтому  $L_A = L(q_3) = L$ .

## 2.2 Произведение автоматов

Рассмотрим одну важную конструкцию конечного автомата по двум другим, называемую произведением автоматов, которая позволит установить замкнутость класса конечно автоматных языков относительно теоретико множественных операций.

Пусть  $M_1 = \langle \Sigma, Q_1, q_0^1, F_1, \Phi_1 \rangle$  и  $M_2 = \langle \Sigma, Q_2, q_0^2, F_2, \Phi_2 \rangle$  — два конечных автомата с общим входным алфавитом  $\Sigma$ , распознающие языки  $L_1$  и  $L_2$ , соответственно. Определим по ним автомат  $M = \langle \Sigma, Q, q_0, F, \Phi \rangle$ , называемый *произведением*  $M_1$  и  $M_2$  ( $M = M_1 \times M_2$ ), следующим образом.  $Q = Q_1 \times Q_2 = \{(q, p) \mid q \in Q_1, p \in Q_2\}$ , т.е. состояния нового автомата — это пары, первый элемент которых — состояние первого автомата, а второй — состояние второго автомата. Для каждой такой пары  $(q, p)$  и входного символа  $a \in \Sigma$  определим функцию переходов:  $\Phi((q, p), a) = (\Phi_1(q, a), \Phi_2(p, a))$ . Начальным состоянием  $M$  является пара  $q_0 = (q_0^1, q_0^2)$ , состоящая из начальных состояний автоматов-множителей. Что касается множества заключи-

тельных состояний, то оно определяется в зависимости от операции над языками  $L_1$  и  $L_2$ , которую должен реализовать  $M$ .

**Теорема 2.1.**

а) При  $F_{\cup} = \{(q, p) \mid q \in F_1 \text{ или } p \in F_2\}$  автомат  $M = \langle \Sigma, Q, q_0, F_{\cup}, \Phi \rangle$  распознает язык  $L = L_1 \cup L_2$ .

б) При  $F_{\cap} = \{(q, p) \mid q \in F_1 \text{ и } p \in F_2\}$  автомат  $M = \langle \Sigma, Q, q_0, F_{\cap}, \Phi \rangle$  распознает язык  $L = L_1 \cap L_2$ .

в) При  $F_{\setminus} = \{(q, p) \mid q \in F_1 \text{ и } p \notin F_2\}$  автомат  $M = \langle \Sigma, Q, q_0, F_{\setminus}, \Phi \rangle$  распознает язык  $L = L_1 \setminus L_2$ .

Доказательство этой теоремы непосредственно выводится из следующего утверждения.

**Лемма 2.2.** Для любых двух состояний  $(q, p)$  и  $(q', p')$  автомата  $M$  и любого входного слова  $w$  слово  $w$  переводит  $(q, p)$  в  $(q', p')$  в автомате  $M$  тогда и только тогда, когда оно переводит  $q$  в  $q'$  в автомате  $M_1$  и  $p$  в  $p'$  в автомате  $M_2$ .

Лемма устанавливается индукцией по длине слова  $w$ .

**Следствие 2.1.1.** Класс конечно автоматных языков замкнут относительно теоретико-множественных операций объединения, пересечения и разности.

## 2.3 Задачи

**Задача 2.1.** Докажите лемму 2.1 на стр. 7 индукцией по длине входного слова.

**Задача 2.2.** Постройте детерминированный конечный автомат, который распознает следующие языки в алфавите  $\sigma = \{a, b\}$ :

- $L = \{w \mid \text{длина } w \text{ делится на } 5\}$ ;
- $L = \{w \mid w \text{ не содержит подслов 'aab' и 'bba'}\}$ ;
- $L = \{w \mid w \text{ содержит четное число букв } a \text{ и нечетное число букв } b\}$ ;
- $L = \{w \mid \text{число букв } a \text{ на } 3 \text{ больше, чем букв } b\}$ .

**Задача 2.3.** Выше в примере 1.1 на стр. 3 был построен автомат с выходом, выполняющий сложение двух двоичных чисел. Постройте автомат-распознаватель, который проверяет правильность сложения. На вход поступают последовательности троек нулей и единиц :

$(x_1(1), x_2(1), y(1)), (x_1(2), x_2(2), y(2)), \dots (x_1(n), x_2(n), y(n))$ .

Автомат должен допустить такую последовательность, если  $y = y(n) \dots y(2)y(1)$  — это первые  $n$  битов суммы двоичных чисел  $x_1 = x_1(n) \dots x_1(2)x_1(1)$  и  $x_2 = x_2(n) \dots x_2(2)x_2(1)$ .

**Задача 2.4.** Докажите лемму 2.2.

### 3 Недетерминированные конечные автоматы и их детерминизация

Недетерминированные конечные автоматы, рассматриваемые в этом параграфе, являются обобщениями детерминированных: они при чтении очередного символа на входе могут выбрать в качестве следующего одно из нескольких состояний, а кроме того, могут изменить состояние без чтения входа. Основной результат, который мы установим, утверждает, что это обобщение не существенно: недетерминированные и детерминированные конечные автоматы распознают одни и те же языки.

**Определение 3.1. Недетерминированный конечный автомат (НКА) - распознаватель** — это система вида

$$M = \langle \Sigma, Q, q_0, F, \Phi \rangle,$$

включающая следующие компоненты:

$\Sigma = \{a_1, \dots, a_m\}$  ( $m \geq 1$ ) — конечное множество - входной алфавит;

$Q = \{q_0, \dots, q_{n-1}\}$  ( $n \geq 1$ ) — конечное множество - алфавит внутренних состояний;

$q_0 \in Q$  — начальное состояние автомата;

$F \subseteq Q$  — множество принимающих (допускающих, заключительных) состояний;

$\Phi : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$  — функция переходов. Для  $a \in \Sigma$  значение  $\Phi(q, a)$  — это множество состояний в каждое из которых может перейти автомат из состояния  $q$ , когда получает на вход символ  $a$ .  $\Phi(q, \varepsilon)$  — это множество состояний в каждое из которых может перейти автомат из состояния  $q$  без чтения символа на входе.

Как и для детерминированных автоматов, функцию переходов можно представить с помощью набора команд-программы: для каждой пары  $q \in Q$  и  $a \in \Sigma$  и каждого состояния  $q' \in \Phi(q, a)$  в программу помещается команда  $qa \rightarrow q'$ , и для каждого состояния  $q' \in \Phi(q, \varepsilon)$  в программу помещается команда  $q \rightarrow q'$ . Отличие от детерминированного случая состоит в том, что для одной пары  $q \in Q$  и  $a \in \Sigma$  в программе может быть несколько команд вида  $qa \rightarrow q'$  или не быть ни одной такой команды. Кроме того, могут появиться  $\varepsilon$ -команды (пустые переходы) вида  $q \rightarrow q'$ , означающие возможность непосредственного перехода из  $q$  в  $q'$  без чтения символа на входе.

При табличном задании функции  $\Phi$  в таблице появляется  $(m + 1)$ -ый столбец, соответствующий пустому символу  $\varepsilon$  и на пересечении строки  $q$  и столбца  $a \in (\Sigma \cup \{\varepsilon\})$  стоит множество состояний  $\Phi(q, a)$ .

Для недетерминированного автомата  $M = \langle \Sigma, Q, q_0, \Phi \rangle$  в диаграмме  $D_M = (Q, E)$  с выделенной начальной вершиной  $q_0$  и множеством заключительных вершин  $F$  ребра взаимно-однозначно соответствуют командам: команде вида  $qa \rightarrow q'$  ( $a \in \Sigma$ )

соответствует ребро  $(q, q')$ , с меткой  $a$ , а команде вида  $q \rightarrow q'$  соответствует ребро  $(q, q')$ , с меткой  $\varepsilon$ .

Скажем, что заданный последовательностью ребер *путь*  $p = e_1 e_2 \dots e_t$  в *диаграмме*  $D_M$  *несет слово*  $w = w_1 w_2 \dots w_t$  ( $t \leq T$ ), если после удаления из него “пустых” ребер (т.е. ребер с метками  $\varepsilon$ ) остается последовательность из  $t$  ребер  $p' = e_{j_1} e_{j_2} \dots e_{j_t}$ , метки которых образуют слово  $w$ , т.е.  $w_i$  — это метка ребра  $e_{j_i}$  ( $1 \leq i \leq t$ ). Очевидно, это эквивалентно тому, что последовательность меток на ребрах пути  $p$  имеет вид  $\varepsilon^{k_1} w_1 \varepsilon^{k_2} w_2 \dots \varepsilon^{k_t} w_t \varepsilon^{k_{t+1}}$ , где  $k_j \geq 0$  ( $j = 1, 2, \dots, t+1$ ) и  $t + \sum_{j=1}^{t+1} k_j = T$ .

*Слово  $w$  переводит  $q$  в  $q'$  в диаграмме  $D_M$* , если в ней имеется путь из  $q$  в  $q'$  который несет  $w$ .

На недетерминированные автоматы естественным образом переносится определение конфигураций и отношения перехода между ними.

**Определение 3.2.** Назовем конфигурацией НКА  $M = \langle \Sigma, Q, q_0, F, \Phi, \rangle$  произвольную пару вида  $(q, w)$ , в которой  $q \in Q$  и  $w \in \Sigma^*$ . Определим отношение  $\vdash_M$  перехода из одной конфигурации в другую за один шаг :

$$(q, w) \vdash_M (q', w') \Leftrightarrow (w = aw' \text{ и } q' \in \Phi(q, a)) \text{ или } (w = w' \text{ и } q' \in \Phi(q, \varepsilon)).$$

Как и для ДКА, через  $\vdash_M^*$  обозначим рефлексивное и транзитивное замыкание отношения  $\vdash_M$ .

Внешне определение распознавания слов НКА совпадает с определением для ДКА.

### Определение 3.3.

НКА  $M$  *распознает (допускает, принимает) слово  $w$* , если для некоторого  $q \in F$   $(q_0, w) \vdash_M^* (q, \varepsilon)$ .

*Язык  $L_M$ , распознаваемый НКА  $M$ , состоит из всех слов, распознаваемых автоматом:*

$$L_M = \{w \mid M \text{ распознает } w\}.$$

Отличие состоит в том, что у НКА может быть несколько различных способов работы (путей вычисления) на одном и том же входном слове  $w$ . Считаем, что НКА распознает (допускает, принимает) это слово, если хотя бы один из этих способов приводит в заключительное состояние из  $F$ .

Из определения диаграммы  $D_M$  непосредственно следует, что НКА  $M$  распознает слово  $w$ , тогда и только тогда, когда существует такое заключительное состояние  $q \in F$ , что в диаграмме  $D_M$  слово  $w$  переводит  $q_0$  в  $q$ . Иными словами, в  $D_M$  имеется путь из  $q_0$  в  $q$ , на ребрах которого написано слово  $w$  (с точностью до меток  $\varepsilon$ ).

**Пример 3.1.** Рассмотрим НКА  $N_1 = \langle \{a, b\}, \{0, 1, 2, 3, 4\}, 0, \{3\}, \Phi \rangle$ , где

$\Phi :$

$Q \setminus \Sigma_X$	a	b	$\varepsilon$
0	$\{0, 1\}$	$\{0\}$	$\emptyset$
1	$\emptyset$	$\emptyset$	$\{4\}$
2	$\{3\}$	$\emptyset$	$\emptyset$
3	$\emptyset$	$\emptyset$	$\{1\}$
4	$\emptyset$	$\{2\}$	$\emptyset$

Его диаграмма  $D_{N_1}$  представлена на следующем рисунке.

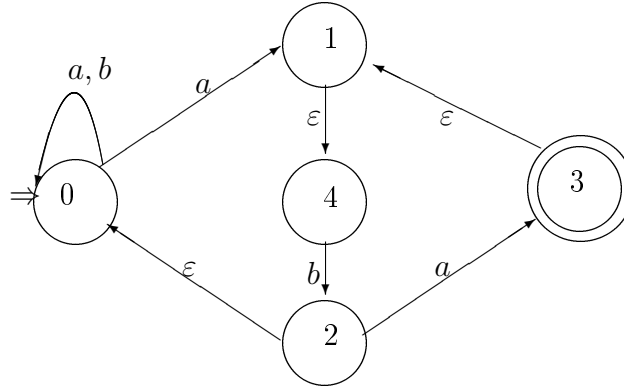


Рис. 3: Диаграмма автомата  $N_1$

Рассмотрим работу этого автомата на слове  $ababa$ :

$(0, ababa) \vdash_{N_1} (1, baba) \vdash_{N_1} (4, baba) \vdash_{N_1} (2, aba) \vdash_{N_1} (0, aba) \vdash_{N_1} (1, ba) \vdash_{N_1} (4, ba) \vdash_{N_1} (2, a) \vdash_{N_1} (3, \varepsilon)$ . Так как 3 — заключительное состояние, то  $ababa \in L_{N_1}$ . Заметим, что у автомата  $N_1$  имеются и другие способы работы на этом слове, не ведущие к заключительному состоянию. Например, он может после чтения каждого символа оставаться в состоянии 0. Но чтобы слово допускалось, достаточно существовать хотя бы *одному* “хорошему” способу.

Очевидно, что детерминированные конечные автоматы являются частными случаями недетерминированных. Естественно спросить, распознают ли недетерминированные конечные автоматы больший класс языков чем детерминированные? Следующая теорема показывает, что классы языков, распознаваемых НКА и ДКА совпадают.

### Теорема 3.1. (Детерминизация НКА)

Для каждого НКА  $M$  можно эффективно построить такой ДКА  $A$ , что  $L_A = L_M$ .

*Доказательство.* Пусть  $M = \langle \Sigma, Q, q_0, F, \Phi \rangle$  — НКА. Процедура построения по нему эквивалентного ДКА состоит из двух этапов: на первом по  $M$  строится эквивалентный ему НКА  $M_1$ , в программе которого отсутствуют переходы по  $\varepsilon$ , а на втором этапе по  $M_1$  строится эквивалентный ДКА  $A$ .

*Этап 1. Устранение пустых переходов.*

Рассмотрим поддиаграму автомата  $M$ , в которой оставлены лишь ребра, помеченные  $\varepsilon$ :  $D_\varepsilon = (Q, E_\varepsilon)$ , где  $E_\varepsilon = \{(q, q') \mid q' \in \Phi(q, \varepsilon)\}$ . Пусть  $D_{\varepsilon^*} = (Q, E_\varepsilon^*)$  — это граф достижимости (транзитивного замыкания) для  $D_\varepsilon$ . Тогда  $E_\varepsilon^* = \{(q, q') \mid q = q' \text{ или в } D_\varepsilon \text{ имеется путь из } q \text{ в } q'\}$ .

Определим НКА  $M_1 = \langle \Sigma, Q_1, q_0, F_1, \Phi_1 \rangle$  следующим образом:

$Q_1 = \{q_0\} \cup \{q \mid \text{существуют такие } q' \in Q \text{ и } a \in \Sigma, \text{ что } q \in \Phi(q', a)\}$ , т.е. кроме начального остаются лишь те состояния, в которые входят “непустые” ребра.

$F_1 = \{q \mid \text{существует такое } q' \in F, \text{ что } (q, q') \in E_\varepsilon^*\}$ , т.е. к заключительным состояниям  $M$  добавляются состояния, из которых можно было попасть в заключительные по путям из  $\varepsilon$ -ребер.

Для каждой пары  $q' \in Q, a \in \Sigma$  полагаем

$\Phi_1(q, a) = \{r \mid \text{существует такое } q' \in F, \text{ что } (q, q') \in E_\varepsilon^* \text{ и } r \in \Phi(q', a)\}$ , т.е. в  $D_{M_1}$  имеется  $a$ -ребро из  $q$  в  $r$ , если в  $D_M$  был (возможно пустой) путь из  $\varepsilon$ -ребер в некоторое состояние  $q'$ , из которого  $a$ -ребро шло в  $r$ .

Из этого определения непосредственно следует, что в НКА  $M_1$  нет пустых переходов по  $\varepsilon$ . Установим эквивалентность  $M$  и  $M_1$ .

**Лемма 3.1.**  $L_{M_1} = L_M$ .

*Доказательство.* Пусть  $w = w_1 w_2 \dots w_t$  — произвольное входное слово. Предположим, что  $w \in L_{M_1}$ . Это означает, что в диаграмме  $D_{M_1}$  имеется путь  $p = e_1 e_2 \dots e_t$  ( $e_1 = (q_0 = r_0, r_1), e_i = (r_{i-1}, r_i), i = 2, \dots, t$ ) из  $q_0$  в некоторое состояние  $r_t \in F_1$ , который несет слово  $w$ , т.е. ребро  $e_i$  помечено символом  $w_i$ . Из определения функции  $\Phi_1$  непосредственно следует, что для любого ребра  $e_i(r_{i-1}, r_i)$  этого пути в диаграмме  $D_M$  имеется путь из  $r_{i-1}$  в  $r_i$ , начало (возможно пустое) которого состоит из  $\varepsilon$ -ребер, а последнее ребро помечено символом  $w_i$ . Объединив эти пути, получим в диаграмме  $D_M$  путь из  $q_0$  в  $r_t$ , который несет слово  $w$ . Так как  $r_t \in F_1$ , то либо  $r_t \in F$ , либо в  $D_M$  имеется путь по  $\varepsilon$ -ребрам из  $r_t$  в некоторое состояние  $r' \in F$ . В обоих случаях в  $D_M$  имеется путь из  $q_0$  в заключительное состояние, который несет слово  $w$ , и следовательно,  $w \in L_M$ .

Обратно, пусть  $w \in L_M$ . Тогда в  $D_M$  имеется путь из  $q_0$  в некоторое заключительное состояние  $r$ , который несет слово  $w$ . Пусть  $r_0 = q_0$ , а  $r_i$  — это состояние этого пути, в которое приводит ребро с меткой  $w_i$  ( $i = 1, \dots, t$ ). Рассмотрим отрезок этого пути между вершинами  $r_{i-1}$  и  $r_i$ . Последнее ребро этого отрезка имеет метку  $w_i$ , а все предыдущие (если они имеются) помечены  $\varepsilon$ . Тогда по определению  $\Phi_1$  в диаграмме  $D_{M_1}$  между  $r_{i-1}$  и  $r_i$  имеется ребро с меткой  $w_i$ . Объединив эти ребра, получим в  $D_{M_1}$  путь из  $q_0$  в  $r_t$ . Так как либо  $r_t = r \in F$ , либо в  $D_M$  из  $r_t$  имеется путь из  $\varepsilon$ -ребер в  $r \in F$ , то из определения  $F_1$  следует, что  $r_t \in F_1$ . Таким образом,  $w \in L_{M_1}$ .  $\square$

*Этап 2. Детерминизация.*

Идея детерминизации состоит в том, что состояниями ДКА объявляются подмножества состояний НКА. Тогда для каждого такого подмножества  $T$  и входного символа  $a$  однозначно определено множество состояний  $T'$ , в которые НКА может попасть из состояний  $T$  при чтении  $a$ .

Определим по НКА  $M_1 = \langle \Sigma, Q_1, q_0, F_1, \Phi_1 \rangle$  ДКА  $A = \langle \Sigma, Q^A, q_0^A, F^A, \Phi^A \rangle$  следующим образом.

$$Q^A = \{Q' \mid Q' \subseteq Q_1\},$$

$$q_0^A = \{q_0\},$$

$$F^A = \{Q' \mid Q' \cap F_1 \neq \emptyset\},$$

$$\Phi^A(Q', a) = \{q \mid \text{существует такое } q' \in Q', \text{ что } q \in \Phi_1(q', a)\}.$$

Ясно, что  $A$  — детерминированный конечный автомат. Следующая лемма устанавливает связь между его вычислениями и вычислениями исходного НКА.

**Лемма 3.2.** *Для любой пары состояний  $Q', Q''$  из  $Q^A$  и любого слова  $w \in \Sigma^*$  имеем  $(Q', w) \vdash_A^* (Q'', \varepsilon) \iff Q'' = \{r \mid \text{существует такое } q' \in Q', \text{ что } (q, w) \vdash_{M_1}^* (r, \varepsilon)\}.$*

*Доказательство.* Применим индукцию по длине слова  $w$ .

*Базис.* Пусть  $|w| = 0$ , тогда  $w = \varepsilon$ ,  $Q' = Q''$  и утверждение выполнено. Пусть теперь  $|w| = 1$  и  $w = a \in \Sigma$ . Тогда утверждение леммы следует непосредственно из определения  $\Phi^A(Q', a)$ .

*Шаг индукции.* Предположим, что лемма справедлива для всех слов длины  $\leq k$ , и пусть  $|w| = k + 1$ . Выделим в  $w$  первый символ:  $w = aw'$ . Пусть  $\tilde{Q}$  — это такое состояние, что  $(Q', a) \vdash_A (\tilde{Q}, \varepsilon)$ . Тогда  $(\tilde{Q}, w') \vdash_A^* (Q'', \varepsilon)$ . Так как  $|w'| = k$ , то по индукционному предположению это эквивалентно тому, что  $Q'' = \{r \mid \text{существует такое } q' \in \tilde{Q}, \text{ что } (q, w') \vdash_{M_1}^* (r, \varepsilon)\}$ . Но из определения  $\Phi^A$  следует, что  $\tilde{Q} = \{q \mid \text{существует такое } q' \in Q', \text{ что } q \in \Phi_1(q', a)\}$ . Объединив, эти два равенства получаем, что  $Q'' = \{r \mid \text{существует такое } q' \in Q', \text{ что } (q, w) \vdash_{M_1}^* (r, \varepsilon)\}$ .

□

Для завершения доказательства теоремы покажем с помощью леммы 3.2, что  $L_A = L_{M_1}$ .

Действительно, если слово  $w$  переводит состояние  $q_0$  в некоторое  $q \in F_1$  в автомате  $M_1$ , то, положив в лемме  $Q' = \{q_0\}$ , получим, что  $q \in Q''$  для состояния  $Q''$ , такого что  $(Q', w) \vdash_A^* (Q'', \varepsilon)$ . Но тогда  $Q'' \in F^A$  и  $w \in L_A$ .

Обратно, если  $w \in L_A$ , то для некоторого  $Q'' \in F^A$  имеем  $(\{q_0\}, w) \vdash_A^* (Q'', \varepsilon)$ . Тогда в  $Q''$  имеется некоторое состояние  $q \in F_1$  и по лемме 3.2 в автомате  $M_1$   $(q_0, w) \vdash_{M_1}^* (q, \varepsilon)$ , т.е.  $w \in L_{M_1}$ .

□

**Пример 3.2.** *Применим процедуру из теоремы о детерминизации к НКА  $N_1$  из примера 3.1 на стр. 13.*

На первом этапе получаем  $E_\varepsilon^* = \{(2, 0), (1, 4), (3, 1), (4, 1)\}$  и НКА  $M_1$  без пустых переходов, представленный на следующей диаграмме.

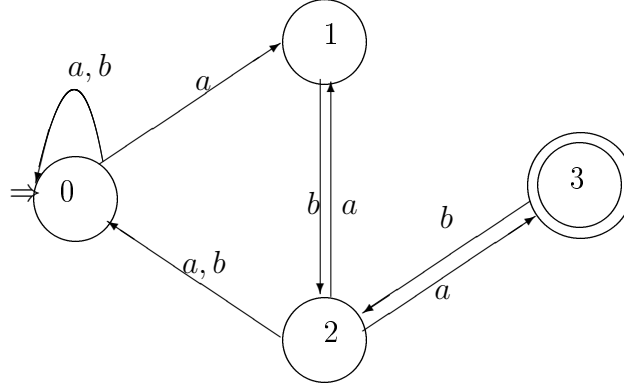


Рис. 4: Диаграмма автомата  $M_1$

Заметим, что состояние 4 исчезло, так как в автомате  $N_1$  в него можно было попасть только по  $\varepsilon$ -переходу.

На втором этапе детерминируем  $M_1$ . ДКА  $A$  будет иметь 16 состояний:  
 $Q^A = \{\emptyset, \{0\}, \{1\}, \{2\}, \{3\}, \{0, 1\}, \{0, 2\}, \{0, 3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{0, 1, 2\}, \{0, 1, 3\}, \{0, 2, 3\}, \{1, 2, 3\}, \{0, 1, 2, 3\}\}.$

Во множество заключительных состояний войдут состояния, содержащие заключительное состояние 3 автомата  $M_1$ :

$$F^A = \{\{3\}, \{0, 3\}, \{1, 3\}, \{0, 1, 3\}, \{0, 2, 3\}, \{1, 2, 3\}, \{0, 1, 2, 3\}\}.$$

Функция переходов  $\Phi^A$  определена в следующей таблице

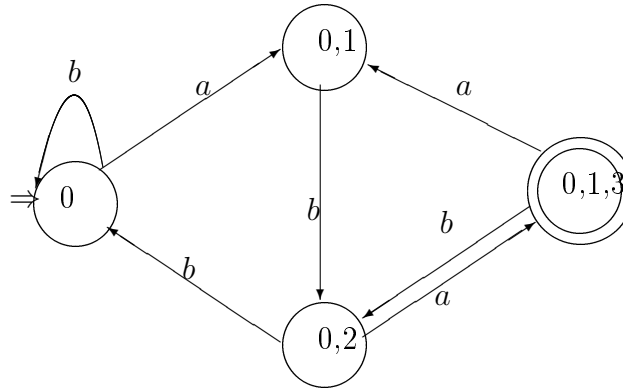
$Q^A \setminus \Sigma$	a	b
$\emptyset$	$\emptyset$	$\emptyset$
$\{0\}$	$\{0, 1\}$	$\{0\}$
$\{1\}$	$\emptyset$	$\{2\}$
$\{2\}$	$\{0, 1, 3\}$	$\{0\}$
$\{3\}$	$\emptyset$	$\{2\}$
$\{0, 1\}$	$\{0, 1\}$	$\{0, 2\}$
$\{0, 2\}$	$\{0, 1, 3\}$	$\{0\}$
$\{0, 3\}$	$\{0, 1\}$	$\{0, 3\}$

$Q^A \setminus \Sigma$	a	b
$\{1, 2\}$	$\{0, 1, 3\}$	$\{0, 2\}$
$\{1, 3\}$	$\emptyset$	$\{2\}$
$\{2, 3\}$	$\{0, 1, 3\}$	$\{0, 2\}$
$\{0, 1, 2\}$	$\{0, 1, 3\}$	$\{0, 2\}$
$\{0, 1, 3\}$	$\{0, 1\}$	$\{0, 2\}$
$\{0, 2, 3\}$	$\{0, 1, 3\}$	$\{0, 2\}$
$\{1, 2, 3\}$	$\{0, 1, 3\}$	$\{0, 2\}$
$\{0, 1, 2, 3\}$	$\{0, 1, 3\}$	$\{0, 2\}$

На самом деле нас интересуют лишь те состояния, в которые можно попасть из начального состояния  $\{0\}$ . Несложный анализ показывает, что их только три:  $\{0, 1\}$ ,  $\{0, 2\}$  и  $\{0, 1, 3\}$ . Остальные состояния не достижимы из  $\{0\}$  и, следовательно, не влияют на работу автомата  $A$ . Их можно отбросить. Таким образом, в диаграмме автомата остаются 4 состояния, показанные на рис. 5.

*Замечание.* В рассмотренном примере у построенного ДКА  $A$  оказалось не больше состояний, чем у исходного НКА  $N_1$ . К сожалению, это не всегда так. Существуют



Рис. 5: Диаграмма автомата  $A$ 

примеры НКА с  $n$  состояниями, для которых эквивалентные ДКА содержат не менее  $2^n$  состояний.

### 3.1 Задачи

**Задача 3.1.** Докажите, что приведенный на рис. 5 автомат  $A$  распознает язык, состоящий из всех слов, заканчивающихся на 'aba'.

**Задача 3.2.** Используя процедуру детерминизации недетерминированных автоматов из теоремы 3.1, постройте ДКА, эквивалентный заданному НКА  $M$ .

а)  $M = \langle \{a, b\}, \{0, 1, 2\}, 0, \{2\}, \Phi \rangle$  с программой  $\Phi: 0a \rightarrow 1, 0 \rightarrow 1, 1b \rightarrow 2, 1 \rightarrow 2, 2b \rightarrow 2$ .

б)  $M = \langle \{a, b\}, \{0, 1, 2\}, 0, \{2\}, \Phi \rangle$  с программой  $\Phi: 0a \rightarrow 1, 0a \rightarrow 2, 1b \rightarrow 2, 1 \rightarrow 2, 2b \rightarrow 0$ .

в)  $M = \langle \{a, b\}, \{0, 1, 2, 3\}, 0, \{2, 3\}, \Phi \rangle$  с программой  $\Phi: 0a \rightarrow 1, 0a \rightarrow 2, 0 \rightarrow 1, 1b \rightarrow 1, 1 \rightarrow 2, 2a \rightarrow 3, 3b \rightarrow 1, 3 \rightarrow 0$ .

## 4 Регулярные выражения и языки

Регулярные выражения являются достаточно удобным средством для построения “алгебраических” описаний языков. Они строятся из элементарных выражений  $\emptyset, \varepsilon, a \in \Sigma$  с помощью операций объединения (+), конкатенации ( $\circ$ ) и итерации (\*). Каждому такому выражению  $r$  соответствует, представляемый им язык  $L_r$ . Смысл операции объединения языков мы знаем. Определим операции конкатенации и итерации (иногда ее называют замыканием Клини).

Пусть  $L_1$  и  $L_2$  — языки в алфавите  $\Sigma$ .

Тогда  $L = L_1 \circ L_2 = \{w \mid (\exists w_1 \in L_1)(\exists w_2 \in L_2)(w = w_1w_2)\}$ , т.е. конкатенация языков

состоит из конкатенаций всех слов первого языка со всеми словами второго языка. В частности, если  $\varepsilon \in L_1$ , то  $L_2 \subseteq L$ , а если  $\varepsilon \in L_2$ , то  $L_1 \subseteq L$ .

Введем обозначения для “степеней” языка  $L$ :

$$L^0 = \{\varepsilon\},$$

$$L^1 = L,$$

$$L^{i+1} = L \circ L^i \quad (i = 1, 2, \dots).$$

Таким образом в  $L^i$  входят все слова, которые можно разбить на  $i$  подряд идущих слов из  $L$ .

Итерацию  $(L)^*$  языка  $L$  образуют все слова которые можно разбить на несколько подряд идущих слов из  $L$ :

$$(L)^* = \{\varepsilon\} \cup \{w \mid (\exists k \geq 1)(w = w_1 w_2 \dots w_k) \text{ и все } w_i \in L\}.$$

Ее можно представить с помощью степеней:

$$(L)^* = \bigcup_{i=0}^{\infty} L^i.$$

Часто удобно рассматривать “усеченную” итерацию языка, которая не содержит пустое слово, если его нет в языке:  $L^+ = \bigcup_{i=1}^{\infty} L^i$ . Это не новая операция, а просто удобное сокращение для выражения  $L \circ L^*$ .

Отметим также, что если рассматривать алфавит  $\Sigma = \{a_1, \dots, a_m\}$  как конечный язык, состоящий из однобуквенных слов, то введенное ранее обозначение  $\Sigma^*$  для множества всех слов, включая и пустое, в алфавите  $\Sigma$  соответствует определению итерации  $\Sigma^*$  этого языка.

В следующей таблице приведено формальное индуктивное определение регулярных выражений над алфавитом  $\Sigma$  и представляемых ими языков.

Выражение $r$	Язык $L_r$
$\emptyset$	$L_{\emptyset} = \emptyset$
$\varepsilon$	$L_{\varepsilon} = \{\varepsilon\}$
$a \in \Sigma$	$L_a = \{a\}$
Пусть $r_1$ и $r_2$ — это регулярные выражения. Тогда следующие выражения являются регулярными	$L_{r_1}$ и $L_{r_2}$ — представляемые ими языки.
$r = (r_1 + r_2)$	и представляют языки:
$r = (r_1 \circ r_2)$	$L_r = L_{r_1} \cup L_{r_2}$
$r = (r_1)^*$	$L_r = L_{r_1} \circ L_{r_2}$
	$L_r = L_{r_1}^*$

При записи регулярных выражений будем опускать знак конкатенации  $\circ$  и будем считать, что операция  $*$  имеет больший приоритет, чем конкатенация и  $+$ , а конкатенация — больший приоритет, чем  $+$ . Это позволит опустить многие скобки. Например,  $((1 \circ 0) \circ ((1)^* + 0))$  можно записать как  $10(1^* + 0)$ .

**Определение 4.1.** Два регулярных выражения  $r$  и  $p$  называются эквивалентными, если совпадают представляемые ими языки, т.е.  $L_r = L_p$ . В этом случае пишем

$r = p$ .

Нетрудно проверить, например, такие свойства операций:

$r + p = p + r$  (коммутативность объединения),  
 $(r + p) + q = r + (p + q)$  (ассоциативность объединения),  
 $(rp)q = r(pq)$  (ассоциативность конкатенации),  
 $(r^*)^* = r^*$  (идемпотентность итерации),  
 $(r + p)q = rq + pq$  (дистрибутивность).

**Пример 4.1.** Докажем в качестве примера не столь очевидное равенство:  $(r+p)^* = (r^*p^*)^*$ .

Пусть  $L_1$  — язык, представляемый его левой частью, а  $L_2$  — правой. Пустое слово  $\varepsilon$  принадлежит обоим языкам. Если непустое слово  $w \in L_1$ , то по определению итерации оно представимо как конкатенация подслов, принадлежащих языку  $L_r \cup L_p$ . Но этот язык является подмножеством языка  $L' = L_r^*L_p^*$  (почему?). Поэтому  $w \in L_2 = (L')^*$ . Обратно, если слово  $w \in L_2$ , то оно представимо как конкатенация подслов, принадлежащих языку  $L'$ . Каждое из таких подслов  $v$  представимо в виде  $v = v_1^1 \dots v_k^1 v_1^2 \dots v_l^2$ , где для всех  $i = 1, \dots, k$  подслово  $v_i^1 \in L_r$  и для всех  $j = 1, \dots, l$  подслово  $v_j^2 \in L_p$  (возможно, что  $k$  или  $l$  равно 0). Но это значит, что  $w$  является конкатенацией подслов, каждое из которых принадлежит  $L_r \cup L_p$  и, следовательно,  $w \in L_1$ .

Рассмотрим несколько примеров регулярных выражений и представляемых ими языков.

**Пример 4.2.** Регулярное выражение  $(0 + 1)^*$  представляет множество всех слов в алфавите  $\{0, 1\}$ .

**Пример 4.3.** Регулярное выражение  $11(0 + 1)^*001$  представляет язык, состоящий из всех слов в алфавите  $\{0, 1\}$ , которые начинаются на '11', а заканчиваются на '001'.

**Пример 4.4.** Регулярное выражение  $(1 + 01 + 001)^*(\varepsilon + 0 + 00)$  представляет язык, состоящий из всех слов в алфавите  $\{0, 1\}$ , которые не содержат подслово '000' (см. задачу 4.1 на стр. 20).

**Пример 4.5.** Регулярное выражение  $1^*(01^*01^*)^*$  представляет язык  $L_{0\text{ч}}$ , состоящий из всех слов в алфавите  $\{0, 1\}$ , в которых четное число нулей.

Действительно, каждое слово из  $L_{0\text{ч}}$  либо вообще не содержит нулей, т.е. входит в язык, представляющий  $1^*$ , либо может быть разбито на блоки вида  $01^i01^j$ ,  $i, j \geq 0$ , которым, быть может, предшествует блок единиц. Выражение  $(01^*01^*)^*$ , очевидно задает один такой блок, а его итерация — произвольную последовательность таких блоков.

**Пример 4.6.** Построим теперь регулярное выражение, которое представляет язык  $L_{0\text{ч}1\text{ч}}$ , состоящий из всех слов в алфавите  $\{0, 1\}$ , в которых четное число нулей и четное число единиц.

Пусть  $w = w_1w_2 \dots w_n$  — произвольное слово из  $L_{0\text{ч}1\text{ч}}$ . Тогда, разумеется,  $n$  — четно, пусть  $n = 2k$ . Разобьем  $w$  на пары соседних букв  $p_i = w_{2i-1}w_{2i}$ ,  $i = 1, 2, \dots, k$ . Возможны 4 вида таких пар: 00, 11, 01 и 10. Пар вида 00 и 11 может быть сколько угодно, а пар вида 01 и 10 обязательно четное число. Поэтому  $w$  разбивается на блоки, каждый из которых начинается одной из пар 01 или 10 и содержит еще одну такую пару. Каждый такой блок описывается выражением  $(01 + 10)(00 + 11)^*(01 + 10)(00 + 11)^*$ . При этом перед первым блоком может быть префикс, состоящий из пар 00 и 11. Множество слов состоящих из пар 00 и 11 задается выражением  $(00 + 11)^*$ . Отсюда получаем выражение  $R_{0\text{ч}1\text{ч}}$ , задающее язык  $L_{0\text{ч}1\text{ч}}$ :

$$R_{0\text{ч}1\text{ч}} = (00 + 11)^*((01 + 10)(00 + 11)^*(01 + 10)(00 + 11)^*)^*.$$

## 4.1 Задачи

**Задача 4.1.** Докажите правильность регулярного выражения в примере 4.4.

**Задача 4.2.** Докажите следующие эквивалентности для регулярных выражений.

- а)  $p^*(p + q)^* = (p + qp^*)^* = (p + q)^*$ ;
- б)  $p(qp)^* = (pq)^*p$ ;
- в)  $(p^*q^*)^* = (q^*p^*)^*$ ;
- г)  $(pq)^+(q^*p^* + q^*) = (pq)^*pq^+p^*$ .

**Задача 4.3.** Постройте регулярное выражение, задающее язык  $L$  в алфавите  $\Sigma = \{0, 1\}$ .

- а)  $L = \{w \mid w \text{ содержит нечетное число букв } 0 \text{ и четное число букв } 1\}$ ;
- б)  $L = \{w \mid w \text{ содержит подслово } 001 \text{ или подслово } 110\}$ ;
- в)  $L = \{w \mid w \text{ не содержит подслов } 011 \text{ и } 010\}$ .

**Задача 4.4.** Определите, какой язык представляется следующими регулярными выражениями.

- а)  $(0^*1^*)0$ ;
- б)  $(01^*)0$ ;
- в)  $(00 + 11 + (01 + 10)(00 + 11)^+(01 + 10))^*$ .

**Задача 4.5.** Упростите следующие регулярные выражения.

- а)  $(00^*)0 + (00)^*$ ;
- б)  $(0 + 1)(\varepsilon + 00)^+ + (0 + 1)$ ;
- в)  $(0 + \varepsilon)0^*1$ .

**Задача 4.6.** Выше в задаче 2.3 на стр. 10 предлагалось построить автомат-распознаватель, который проверяет правильность сложения. Постройте регулярное выражение, задающее распознаваемый этим автоматом язык  $S$ , т.е. следующее множество слов в алфавите  $\{0, 1\}$ <sup>3</sup>

$S = \{(x_1(1), x_2(1), y(1))(x_1(2), x_2(2), y(2)) \dots (x_1(n), x_2(n), y(n)) \mid y = y(n) \dots y(2)y(1)$  — это первые  $n$  битов суммы двоичных чисел  $x_1 = x_1(n) \dots x_1(2)x_1(1)$  и  $x_2 = x_2(n) \dots x_2(2)x_2(1)\}$ .

## 5 Регулярные языки и конечные автоматы

### 5.1 Автоматы для регулярных языков

Покажем, что каждый регулярный язык можно распознать конечным автоматом.

**Теорема 5.1.** Для каждого регулярного выражения  $r$  можно эффективно построить такой недетерминированный конечный автомат  $M$ , который распознает язык, задаваемый  $r$ , т.е.  $L_M = L_r$ .

*Доказательство.* Построение автомата  $M$  по выражению  $r$  проведем индукцией по длине  $r$ , т.е. по общему количеству символов алфавита  $\Sigma$ , символов  $\emptyset$  и  $\varepsilon$ , знаков операций  $+$ ,  $\circ$ ,  $*$  и скобок в записи  $r$ .

*Базис.* Автоматы для выражений длины 1:  $\emptyset$ ,  $\varepsilon$  и  $a \in \Sigma$  показаны на следующем рисунке.

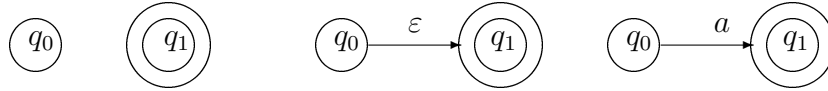


Рис. 6: Диаграммы автоматов для языков  $\emptyset$ ,  $\{\varepsilon\}$  и  $\{a\}$

Заметим, что у каждого из этих трех автоматов множество заключительных состояний состоит из одного состояния.

*Индукционный шаг.* Предположим теперь, что для каждого регулярного выражения длины  $\leq k$  построен соответствующий НКА, причем у него единственное заключительное состояние. Рассмотрим произвольное регулярное выражение  $r$  длины  $k+1$ . В зависимости от последней операции оно может иметь один из трех видов:  $(r_1 + r_2)$ ,  $(r_1 r_2)$  или  $(r_1)^*$ . Пусть  $M_1 = \langle \Sigma, Q_1, q_0^1, \{q_f^1\}, \Phi_1 \rangle$  и  $M_2 = \langle \Sigma, Q_2, q_0^2, \{q_f^2\}, \Phi_2 \rangle$  —

это НКА, распознающие языки  $L_{r_1}$  и  $L_{r_2}$ , соответственно. Не ограничивая общности, мы будем предполагать, что у них разные состояния:  $Q_1 \cap Q_2 = \emptyset$ .

Тогда НКА  $M = \langle \Sigma, Q, q_0, \{q_f\}, \Phi \rangle$ , диаграмма которого представлена на следующем рисунке, распознает язык  $L_r = L_{r_1+r_2} = L_{r_1} \cup L_{r_2}$ .

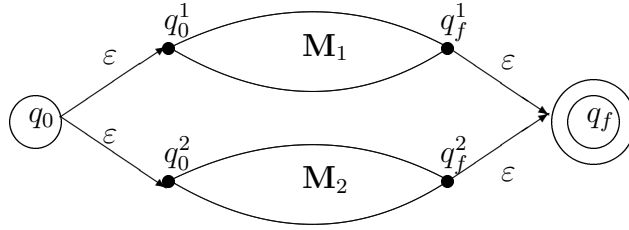


Рис. 7: Диаграмма автомата  $M$ , распознающего язык  $L_{r_1+r_2}$

У этого автомата множество состояний  $Q = Q_1 \cup Q_2 \cup \{q_0, q_f\}$ , где  $q_0$  — это новое начальное состояние,  $q_f$  — новое (единственное !) заключительное состояние, а программа включает программы автоматов  $M_1$  и  $M_2$  и четыре новых команды  $\varepsilon$ -переходов:  $\Phi = \Phi_1 \cup \Phi_2 \cup \{q_0 \rightarrow q_0^1, q_0 \rightarrow q_0^2, q_f^1 \rightarrow q_f, q_f^2 \rightarrow q_f\}$ . Очевидно, что язык, распознаваемый НКА  $M$ , включает все слова из  $L_{M_1}$  и из  $L_{M_2}$ . С другой стороны, каждое слово  $w \in L_M$  переводит  $q_0$  в  $q_f$  и после первого шага, несущий его путь проходит через  $q_0^1$  или  $q_0^2$ . Так как состояния  $M_1$  и  $M_2$  не пересекаются, то в первом случае этот путь может попасть в  $q_f$  только по  $\varepsilon$ -переходу из  $q_f^1$  и тогда  $w \in L_{M_1}$ . Аналогично, во втором случае  $w \in L_{M_2}$ .

Для выражения  $r = r_1 \circ r_2$  диаграмма НКА  $M = \langle \Sigma, Q, q_0, \{q_f\}, \Phi \rangle$ , распознающего язык  $L_r$ , представлена на следующем рисунке.

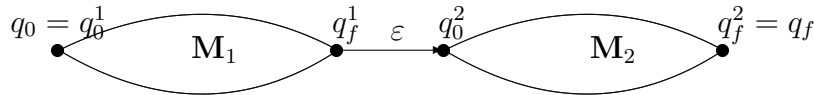


Рис. 8: Диаграмма автомата  $M$ , распознающего язык  $L_{r_1 \circ r_2}$

У этого автомата множество состояний  $Q = Q_1 \cup Q_2$ , начальное состояние  $q_0 = q_0^1$ , заключительное состояние  $q_f = q_f^2$ , а программа включает программы автоматов  $M_1$  и  $M_2$  и одну новую команду —  $\varepsilon$ -переход из заключительного состояния  $M_1$  в начальное состояние  $M_2$ , т.е.  $\Phi = \Phi_1 \cup \Phi_2 \cup \{q_f^1 \rightarrow q_0^2\}$ . Здесь также очевидно, что всякий путь из  $q_0 = q_0^1$  в  $q_f = q_f^2$  проходит через  $\varepsilon$ -переход из  $q_f^1$  в  $q_0^2$ . Поэтому всякое слово, допускаемое  $M$ , представляет конкатенацию некоторого слова из  $L_{M_1}$  с некоторым словом из  $L_{M_2}$ , и любая конкатенация таких слов допускается. Следовательно, НКА  $M$  распознает язык  $L_r = L_{r_1 \circ r_2} = L_{r_1} L_{r_2}$ .

Пусть  $r = r_1^*$ . Диаграмма НКА  $M = \langle \Sigma, Q, q_0, \{q_f\}, \Phi \rangle$ , распознающего язык  $L_r = L_{r_1^*} = L_{M_1}^*$  представлена на рисунке 5.1 на стр. 23.

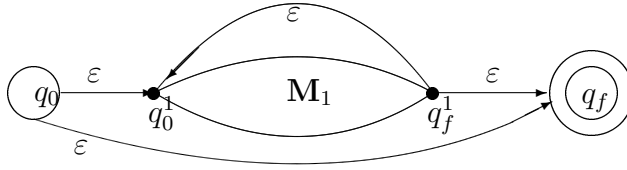


Рис. 9: Диаграмма автомата  $M$ , распознающего язык  $L_{r_1^*}$

У этого автомата множество состояний  $Q = Q_1 \cup \{q_0, q_f\}$ , где  $q_0$  — это новое начальное состояние,  $q_f$  — новое (единственное !) заключительное состояние, а программа включает программу автомата  $M_1$  и четыре новых команды  $\varepsilon$ -переходов:  $\Phi = \Phi_1 \cup \{q_0 \rightarrow q_0^1, q_0 \rightarrow q_f^1, q_f^1 \rightarrow q_0^1, q_f^1 \rightarrow q_f\}$ . Очевидно,  $\varepsilon \in L_M$ . Для непустого слова  $w$  по определению итерации  $w \in L_{r_1^*} \Leftrightarrow$  для некоторого  $k \geq 1$  слово  $w$  можно разбить на  $k$  подслов:  $w = w_1 w_2 \dots w_k$  и все  $w_i \in L_{M_1}$ . Для каждого  $i = 1, \dots, k$  слово  $w_i$  переводит  $q_0^1$  в  $q_f^1$ . Тогда для слова  $w$  в диаграмме  $M$  имеется путь  $q_0 \xrightarrow{\varepsilon} q_0^1 \xrightarrow{w_1} q_f^1 \xrightarrow{\varepsilon} q_0^1 \xrightarrow{w_2} q_f^1 \dots q_0^1 \xrightarrow{w_k} q_f^1 \xrightarrow{\varepsilon} q_f$ . Следовательно,  $w \in L_M$ . Обратно, если некоторое слово переводит  $q_0$  в  $q_f$ , то либо оно есть  $\varepsilon$ , либо его несет путь, который, перейдя из  $q_0$  в  $q_0^1$  и затем пройдя несколько раз по пути из  $q_0^1$  в  $q_f^1$  и вернувшись из  $q_f^1$  в  $q_0^1$  по  $\varepsilon$ -переходу, в конце концов из  $q_f^1$  по  $\varepsilon$ -переходу завершается в  $q_f$ . Поэтому такое слово  $w \in L_{M_1}^*$ .

□

Из теорем 3.1 и 5.1 непосредственно получаем

**Следствие 5.1.1.** Для каждого регулярного выражения можно эффективно построить детерминированный конечный автомат, который распознает язык, представляемый этим выражением.

Это утверждение — один из примеров *теорем синтеза*: по описанию задания (языка как регулярного выражения) эффективно строится программа (ДКА), его выполняющая. Справедливо и обратное утверждение — *теорема анализа*.

**Теорема 5.2.** По каждому детерминированному (или недетерминированному) конечному автомату можно построить регулярное выражение, которое представляет язык, распознаваемый этим автоматом.

Доказательство этой теоремы достаточно техническое и выходит за рамки нашего курса.

Таким образом, можно сделать вывод, что класс конечно автоматных языков совпадает с классом регулярных языков. Далее мы будем называть его просто *классом автоматных языков*.

Автомат  $M_r$ , который строится в доказательстве теоремы 5.1 по регулярному выражению  $r$ , не всегда является самым простым.

Например, для реализации выражения-слова  $a_1a_2 \dots a_n$ , где  $a_i \in \Sigma$  ( $i = 1, 2, \dots, n$ ), можно просто использовать автомат с  $(n + 1)$  состоянием  $q_i$  ( $i = 0, 1, 2, \dots, n$ ) и командами  $q_{i-1}a_i \rightarrow q_i$ , в котором нет пустых  $\varepsilon$ -переходов, участвующих в общей конструкции для конкатенации. Также при построении автомата для объединения  $M_1$  и  $M_2$  можно сливать их начальные состояния в одно, если в них нет переходов из других состояний (тогда не потребуется новое начальное состояние). Можно также объединить их заключительные состояния, если из них нет переходов в другие состояния и алфавиты  $M_1$  и  $M_2$  совпадают. Если из заключительного состояния  $M_1$  нет переходов в другие состояния, то при конкатенации его можно объединить с начальным состоянием  $M_2$ . Вместе с тем, утверждения задачи 5.1 на стр. 30 показывают, что наша общая конструкция достаточно экономна.

**Пример 5.1.** Применим теорему 5.1 к регулярному выражению  $r = (1 + 01 + 001)^*(\varepsilon + 0 + 00)$ , которое, как мы заметили в примере 4.4 на стр. 19, представляет язык, состоящий из всех слов, которые не содержат подслово '000'.

На рисунке 10 представлены диаграммы автоматов  $M_1$  и  $M_2$ , построенных по выражениям  $r_1 = (1 + 01 + 001)$  и  $r_2 = (\varepsilon + 0 + 00)$ , соответственно, с помощью конструкций для конкатенации и объединения. Как мы отмечали выше, автомат  $M_1$  можно было бы еще упростить, склеив начальные состояния  $q_2, p_1$  и  $s_1$ , а также заключительные состояния  $q_3, p_3$  и  $s_4$ .

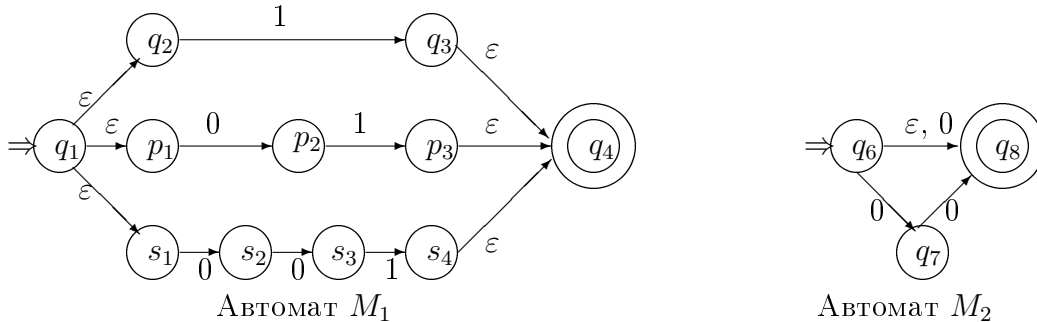
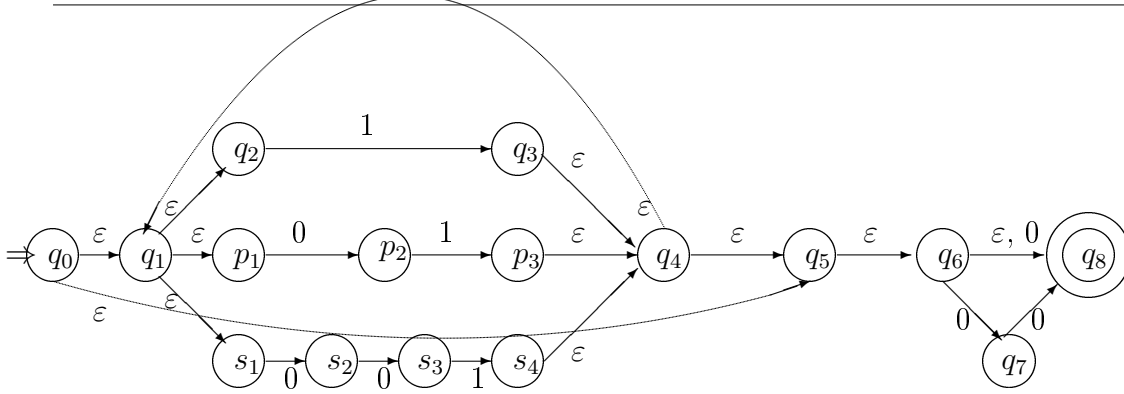


Рис. 10: Диаграммы автоматов для выражений  $r_1 = (1 + 01 + 001)$  и  $r_2 = (\varepsilon + 0 + 00)$

Автомат  $M_3$  для выражения  $r_1^* = (1 + 01 + 001)^*$  получается из  $M_1$  добавлением нового начального состояния  $q_0$  и заключительного состояния  $q_5$  и  $\varepsilon$ -переходов из  $q_0$  в  $q_1$  и  $q_5$ , из  $q_4$  в  $q_5$  и из  $q_5$  в  $q_1$ . Затем результирующий автомат для исходного выражения  $r$  получается последовательным соединением  $M_3$  и  $M_2$ . Он представлен на следующем рисунке.



Рис. 11: Диаграмма автомата  $M$ , распознающего язык  $L_r$ 

## 5.2 Свойства замкнутости класса автоматных языков

Обратимся снова к свойствам замкнутости класса автоматных языков. Как мы уже установили с помощью конструкции произведения автоматов, этот класс замкнут относительно объединения, пересечения и разности (см. следствие 2.1.1 на стр. 10). Из теоремы 5.1 на стр. 21 непосредственно следует, что класс автоматных языков замкнут относительно операций конкатенации и итерации. Можно легко установить, что он также замкнут относительно дополнения.

**Предложение 5.1.** Пусть  $L$  — автоматный язык в алфавите  $\Sigma$ . Тогда его дополнение — язык  $\bar{L} = \{w \mid w \in \Sigma^* \text{ и } w \notin L\}$  также является автоматным.

Действительно, достаточно заметить, что язык  $\Sigma^*$ , включающий все слова в алфавите  $\Sigma$  является автоматным и что  $\bar{L} = \Sigma^* \setminus L$ .

Определенная ниже операция гоморфизма формализует идею посимвольного перевода слов одного алфавита в слова другого.

**Определение 5.1.** Пусть  $\Sigma$  и  $\Delta$  — два алфавита. Отображение  $\phi: \Sigma^* \rightarrow \Delta^*$  слов первого из них в слова второго называется **гоморфизмом**, если

- 1)  $\phi(\varepsilon) = \varepsilon$ ;
- 2) для любых двух слов  $w_1$  и  $w_2$  в алфавите  $\Sigma$  имеет место равенство  $\phi(w_1 w_2) = \phi(w_1) \phi(w_2)$ .

Из этого определения непосредственно следует, что гоморфизм однозначно определяется своими значениями на символах алфавита  $\Sigma$ . Если  $w = w_1 w_2 \dots w_n$ ,  $w_i \in \Sigma$  ( $1 \leq i \leq n$ ), то  $\phi(w) = \phi(w_1) \phi(w_2) \dots \phi(w_n)$ .

**Пример 5.2.** Пусть  $\Sigma = \{a, b, c\}$ ,  $\Delta = \{0, 1\}$ , а гоморфизм  $\phi$  определен на символах  $\Sigma$  следующим образом:  $\phi(a) = 00$ ,  $\phi(b) = \varepsilon$ ,  $\phi(c) = 101$ .

Тогда  $\phi(aca) = 0010100$ ,  $\phi(abcb) = 00101$ ,  $\phi(bbb) = \varepsilon$ .

**Определение 5.2.** Пусть  $\phi : \Sigma^* \rightarrow \Delta^*$  — произвольный гомоморфизм и  $L$  — язык в алфавите  $\Sigma$ . Образом  $\phi(L)$  языка  $L$  при гомоморфизме  $\phi$  называется язык  $\phi(L) = \{\phi(w) \mid w \in L\}$ , состоящий из образов всех слов языка  $L$ .

Пусть  $L$  — язык в алфавите  $\Delta$ . Прообразом этого языка при гомоморфизме  $\phi$  называется язык  $\phi^{-1}(L) = \{w \in \Sigma^* \mid \phi(w) \in L\}$ , состоящий из всех таких слов в алфавите  $\Sigma$ , чьи образы при гомоморфизме  $\phi$  попадают в  $L$ .

Оказывается, что класс автоматных языков замкнут относительно операций гомоморфизма и обращения гомоморфизма (взятия прообраза) .

**Теорема 5.3.** Пусть  $\phi : \Sigma^* \rightarrow \Delta^*$  — произвольный гомоморфизм и  $L$  — автоматный язык в алфавите  $\Sigma$ . Тогда и язык  $\phi(L)$  является автоматным.

*Доказательство.* Пусть  $A = \langle \Sigma, Q, q_0, F, \Phi \rangle$  — ДКА, распознающий язык  $L$ . Построим по нему НКА  $M = \langle \Delta, Q^M, q_0^M, F^M, \Phi^M \rangle$ , распознающий язык  $\phi(L)$ . Идея этого построения проста: нужно каждый переход из состояния  $q$  в  $q'$  по букве  $a \in \Sigma$  в автомате  $A$  превратить в переход из  $q$  в  $q'$  по слову  $\phi(a)$  в автомате  $M$ .

Пусть  $\Sigma = \{a_1, \dots, a_m\}$ ,  $Q = \{q_0, q_1, \dots, q_n\}$  и  $\phi(a_i) = d_1^i d_2^i \dots d_{k_i}^i$ ,  $d_l^i \in \Delta$  ( $1 \leq l \leq k_i$ ) (если  $\phi(a_i) \neq \varepsilon$ ). Для каждого  $a_i$  зафиксируем простой НКА  $M_i$ , распознающий язык  $\{d_1^i d_2^i \dots d_{k_i}^i\}$ , имеющий  $(k_i + 1)$  состояние  $p_0^i, p_1^i, \dots, p_{k_i}^i$  и команды  $p_{l-1}^i d_l^i \rightarrow p_l^i$  ( $1 \leq l \leq k_i$ ). (Если  $\phi(a_i) = \varepsilon$ , то у  $M_i$  будут два состояния, соединенные  $\varepsilon$ -переходом). Теперь для каждой команды  $q_j a_i \rightarrow q_r$  поместим в  $M$  между  $q_j$  и  $q_r$  автомат  $M_i$  (цепочку состояний  $p_0^i, p_1^i, \dots, p_{k_i}^i$ ). Чтобы состояния различных цепочек не склеивались придадим им верхний индекс  $j$ , т.е. у каждого  $q_j$  будет своя копия каждого из автоматов  $M_i$ . Для этого положим  $Q^M = Q \cup \{p_l^{ji} \mid 0 \leq j \leq n, 1 \leq i \leq m, 0 \leq l \leq k_i\}$ . Таким образом,  $p_l^{ji}$  — это  $l$ -ое состояние на пути из  $q_j$  по “старой” букве  $a_i$ . Программа  $\Phi^M$  автомата  $M$  строится по программе  $A$  следующим образом. Для каждой команды вида  $q_j a_i \rightarrow q_r$  из  $\Phi$  поместим в  $\Phi^M$  следующие команды:

$$\begin{aligned} q_j &\rightarrow p_0^{ji}, \\ p_0^{ji} d_1^i &\rightarrow p_1^{ji}, \\ &\bullet \bullet \bullet \\ p_{k_i-1}^{ji} d_{k_i}^i &\rightarrow p_{k_i}^{ji}, \\ p_{k_i}^{ji} &\rightarrow q_r. \end{aligned}$$

Таким образом, из  $q_j$  автомат  $M$  по пустому переходу попадает в начальное состояние  $p_0^{ji}$   $j$ -ой копии автомата  $M_i$ , затем проходит по слову  $\phi(a_i)$  и снова по пустому переходу попадает в  $q_r$ .

Для завершения определения  $M$  положим  $q_0^M = q_0$  и  $F^M = F$ .

Докажем теперь, что наше построение корректно, т.е., что  $\phi(L) = \phi(L_A) = L_M$ .

1)  $\phi(L) \subseteq L_M$ . Заметим вначале, что если  $\varepsilon \in L$ , то  $q_0 \in F$  и по определению  $q_0 \in F^M$ , следовательно  $\phi(\varepsilon) = \varepsilon \in L_M$ .

Пусть  $w = w_1 w_2 \dots w_k \in L, w_s \in \Sigma$ . Тогда в диаграмме  $A$  имеется путь из  $q_0$  в некоторое заключительное состояние  $q' \in F$ , который несет слово  $w$ . Пусть это путь  $q_0 = q_{j_0}, q_{j_1}, \dots, q_{j_k} = q'$ . Тогда для каждого  $1 \leq x \leq k$  в  $\Phi$  имеется команда  $q_{j_{x-1}} w_x \rightarrow q_{j_x}$ . Но из определения  $\Phi^M$  следует, что тогда в автомате  $M$  имеется путь из  $q_{j_{x-1}}$  в  $q_{j_x}$ , несущий слово  $\phi(w_x)$ . Объединив все такие пути, получим путь из  $q_0$  в  $q' \in F^M$ , несущий слово  $\phi(w)$ . Следовательно,  $\phi(w) \in L_M$ .

2)  $L_M \subseteq \phi(L)$ . Пусть слово  $u \in \Delta^*$  принадлежит  $L_M$ . Покажем, что тогда для некоторого  $w \in L$   $u = \phi(w)$ . Рассмотрим для этого путь в диаграмме  $M$  из  $q_0$  в  $q' \in F^M$ , несущий слово  $u$ . Выделим на этом пути все состояния из  $Q$ . Пусть это будут по порядку состояния  $q_0 = q_{j_0}, q_{j_1}, \dots, q_{j_k} = q'$ . Тогда слово  $u$  разбивается на  $k$  подслов:  $u = u_1 u_2 \dots u_k$  таких, что  $u_x$  переводит в  $M$  состояние  $q_{j_{x-1}}$  в  $q_{j_x}$  ( $1 \leq x \leq k$ ). Покажем, что для каждого такого  $u_x$  существует символ  $w_x \in \Sigma$  такой, что  $u_x = \phi(w_x)$  и в  $\Phi$  имеется команда  $q_{j_{x-1}} w_x \rightarrow q_{j_x}$ . Действительно, любой путь из  $q_{j_{x-1}}$  в  $M$  начинается  $\varepsilon$ -переходом в некоторое состояние вида  $p_0^{j_{x-1}i}$ . Пусть это будет состояние на пути, который несет  $u_x$  в  $q_{j_x}$ . Далее этот путь обязательно будет проходить по состояниям вида  $p_l^{j_{x-1}i}$  ( $l = 1, \dots, k_i$ ) и завершится  $\varepsilon$ -переходом из  $p_{k_i}^{j_{x-1}i}$  в состояние  $q_{j_x}$ . Тогда из определения  $M$  следует, что  $u_x = \phi(a_i)$  и в  $\Phi$  имеется команда  $q_{j_{x-1}} w_x \rightarrow q_{j_x}$ . Положив  $w_x = a_i$ , получим, что  $u_x = \phi(w_x)$  и  $u = \phi(w_1) \phi(w_2) \dots \phi(w_k) = \phi(w)$ , для слова  $w = w_1 w_2 \dots w_k \in \Sigma^*$ . При этом каждый символ  $w_x$  этого слова переводит в автомате  $A$  состояние  $q_{j_{x-1}}$  в  $q_{j_x}$ . Поэтому в  $A$  существует путь из  $q_0$  в  $q' \in F$ , несущий слово  $w$  и, следовательно  $w \in L$ .

□

**Пример 5.3.** Пусть алфавиты  $\Sigma, \Delta$  и гомоморфизм  $\phi$  определены как выше в примере 5.2 на стр. 25. Рассмотрим язык  $L = \{w \mid \text{число букв } a \text{ в слове } w \text{ нечетно}\}$ .

На следующем рисунке показана диаграмма ДКА  $A$ , распознающего язык  $L$  и диаграммы автоматов  $M_a$  для  $\phi(a) = 00$ ,  $M_b$  для  $\phi(b) = \varepsilon$  и  $M_c$  для  $\phi(c) = 101$ .

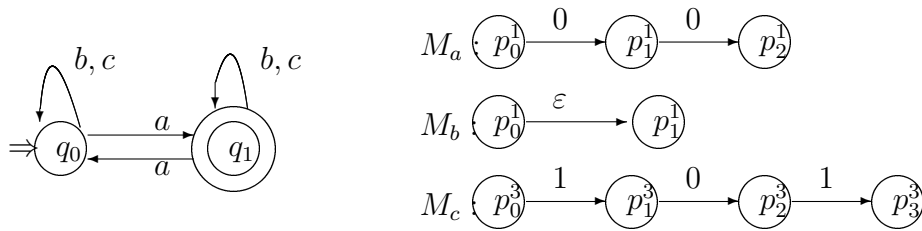


Рис. 12: Диаграмма автомата  $A$ , распознающего язык  $L$ , и автоматов  $M_a, M_b$  и  $M_c$

Подставив в  $A$  вместо  $a$ -переходов автомат  $M_a$ , вместо  $b$ -переходов автомат  $M_b$  и вместо  $c$ -переходов автомат  $M_c$ , получим представленный на рис. 13 на стр. 28 недетерминированный автомат  $M$ , распознающий язык  $\phi(L)$ . На этом рисунке каждая из  $\varepsilon$ -петель в состояниях  $q_0$  и  $q_1$  заменяет по три  $\varepsilon$ -перехода, связанных с  $M_b$ .

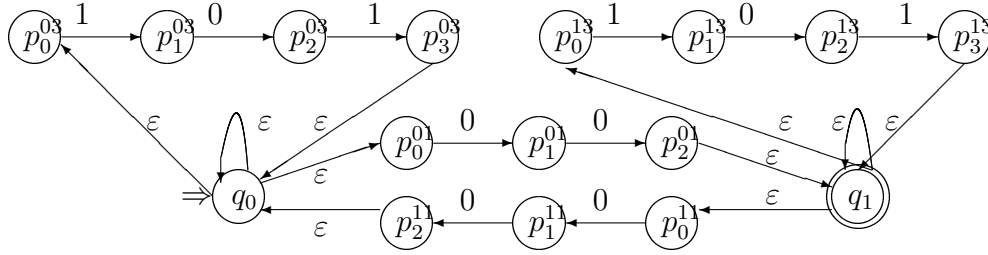


Рис. 13: Диаграмма автомата  $M$ , распознающего язык  $\phi(L)$

Отметим, что конструкция автомата  $M$  в теореме 5.3 добра для доказательства, но несколько избыточна. Без труда можно сократить в ней все  $\epsilon$ -переходы, склеив начальные и заключительные состояния автоматов  $M_i$  с соответствующими состояниями автомата  $A$ . Например, в автомате на рис. 13 на стр. 28 можно объединить начальные состояния  $p_0^0$  и  $p_0^{01}$  с  $q_0$ , заключительные состояния  $p_2^{01}$  и  $p_3^{13}$  с  $q_1$  и т.п.

Одним из интересных частных случаев гомоморфизма является проекция.

**Определение 5.3.** Пусть  $\Delta \subset \Sigma$ . Проекцией языка  $L$  в алфавите  $\Sigma$  на подалфавит  $\Delta$  называется язык

$PROJ_{\Delta}(L) = \{w \mid w \text{ получено из некоторого слова } v \in L \text{ вычеркиванием всех символов, не принадлежащих алфавиту } \Delta\}$ .

Определим гомоморфизм  $\pi : \Sigma^* \rightarrow \Delta^*$  следующим образом:  $\pi(a) = a$ , если  $a \in \Delta$  и  $\pi(a) = \epsilon$ , если  $a \notin \Delta$ . Тогда для всякого языка  $L$  в алфавите  $\Sigma$  имеет место равенство  $PROJ_{\Delta}(L) = \pi(L)$ . Отсюда и из предыдущей теоремы 5.3 получаем замкнутость класса автоматных языков относительно проекции.

**Предложение 5.2.** Для любых алфавитов  $\Delta$  и  $\Sigma$  таких, что  $\Delta \subset \Sigma$ , и любого автоматного языка  $L$  в алфавите  $\Sigma$  проекция  $PROJ_{\Delta}(L)$  также является автоматным языком.

Отметим, что для проекции конструкция автомата  $M$  для  $PROJ_{\Delta}(L)$  по ДКА  $A$  для  $L$  существенно упрощается: достаточно заменить в  $A$  все переходы по символам из  $\Sigma \setminus \Delta$  заменить на  $\epsilon$ -переходы.

Следующая теорема устанавливает замкнутость класса автоматных языков относительно обращения гомоморфизмов.

**Теорема 5.4.** Пусть  $\phi : \Sigma^* \rightarrow \Delta^*$  — произвольный гомоморфизм и  $L$  — автоматный язык в алфавите  $\Delta$ . Тогда и язык  $\phi^{-1}(L)$  является автоматным.

*Доказательство.* Пусть  $A = \langle \Delta, Q, q_0, F, \Phi \rangle$  — ДКА, распознающий язык  $L$ . Пусть  $\Sigma = \{a_1, \dots, a_m\}$ ,  $Q = \{q_0, q_1, \dots, q_n\}$  и  $\phi(a_i) = d_1^i d_2^i \dots d_{k_i}^i$ ,  $d_l^i \in \Delta$  ( $1 \leq l \leq k_i$ ) (если  $\phi(a_i) \neq \varepsilon$ ).

Перестроим его в ДКА  $M = \langle \Sigma, Q, q_0, F, \Phi^M \rangle$  с тем же множеством состояний, начальным и заключительными состояниями, который распознает язык  $\phi^{-1}(L)$ .

Идея этого построения состоит в том, чтобы переходить из состояния  $q$  в  $q'$  по букве  $a \in \Sigma$  в автомате  $M$ , если в автомате  $A$  слово  $\phi(a) \neq \varepsilon$  переводит  $q$  в  $q'$ . Если же для  $a \in \Sigma$  образ пуст, т.е.  $\phi(a) = \varepsilon$ , то в автомате  $M$  слово  $a$  переводит каждое состояние в себя, так как символы  $a$  могут встречаться в каждом слове из  $\phi^{-1}(L)$  в любом месте и в любом количестве.

Таким образом, положим для каждой пары  $q_j \in Q$  и  $a_i \in \Sigma$   $\Phi^M(q_j, a_i) = q_r$ , если  $\phi(a_i) \neq \varepsilon$  и в автомате  $A$   $(q, \phi(a)) \vdash_A^* q_r$ . Если же  $\phi(a) = \varepsilon$ , то полагаем  $\Phi^M(q_j, a_i) = q_j$ .

Так как  $A$  — детерминированный автомат, то функция переходов  $\Phi^M$  определена однозначно и для всех пар  $q_j \in Q$  и  $a_i \in \Sigma$ . Следовательно,  $M$  детерминированный.

Нетрудно показать, что  $L_M = \phi^{-1}(L)$ . Действительно, если слово  $w = a_{i_1} a_{i_2} \dots a_{i_k} \in L_M$ , то в  $M$  путь  $q_0, q_{i_1} q_{i_2} \dots q_{i_k}$ , несущий это слово ведет в заключительное состояние  $q_{i_k} \in F$ . Из определения  $\Phi^M$  следует, что тогда в  $A$  существует соответствующий путь из  $q_0$  в  $q_{i_k} \in F$ , который несет слово  $\phi(a_{i_1}) \phi(a_{i_2}) \dots \phi(a_{i_k}) = \phi(w)$ . Следовательно,  $w \in \phi^{-1}(L)$ .

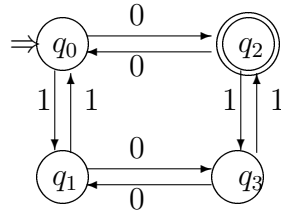
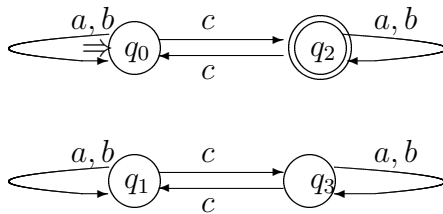
Обратно, пусть  $w = a_{i_1} a_{i_2} \dots a_{i_k} \in \phi^{-1}(L)$ . Тогда слово  $u = \phi(w) = \phi(a_{i_1}) \phi(a_{i_2}) \dots \phi(a_{i_k}) \in L$  и в автомате  $A$  имеется путь, несущий  $u$ , который переводит  $q_0$  в некоторое заключительное состояние  $q' \in F$ . Зафиксируем на этом пути состояния  $q_{i_j}$ , в которые он попадает после прочтения префиксов  $\phi(a_{i_1}) \phi(a_{i_2}) \dots \phi(a_{i_j})$  слова  $u$  ( $j = 1, 2, \dots, k$ ). Тогда  $q_{i_k} = q'$  и для всех  $j = 1, 2, \dots, k$  имеет место  $(q_{i_{j-1}}, \phi(a_{i_j})) \vdash_A^* q_{i_j}$ . Отсюда и из определения  $\Phi^M$  получаем, что в  $M$  для всех  $j = 1, 2, \dots, k$  имеет место переход на один шаг  $(q_{i_{j-1}}, a_{i_j}) \vdash_M q_{i_j}$ . Следовательно, в  $M$  путь  $q_0, q_{i_1} q_{i_2} \dots q_{i_k}$  несет слово  $w$  и завершается в заключительном состоянии  $q_{i_k} = q'$ . А это означает, что  $w \in L_M$ .  $\square$

**Пример 5.4.** Пусть алфавиты  $\Sigma$ ,  $\Delta$  и гомоморфизм  $\phi$  определены как выше в примере 5.2:  $\phi(a) = 00$ ,  $\phi(b) = \varepsilon$ ,  $\phi(c) = 101$ . Рассмотрим язык  $L = \{w \mid \text{число букв } 0 \text{ в слове } w \text{ нечетно, а число букв } 1 \text{ — четно}\}$ .

На следующем рисунке показана диаграмма ДКА  $A$ , распознающего язык  $L$ .

Применив к этому автомату конструкцию из теоремы 5.4, обнаружим, что  $a$  и  $b$  оставляют все состояния на месте, а  $c$  переводит каждое состояние в соседнее состояние “по горизонтали”. В результате получаем автомат  $M$ , показанный ниже на рис. 15.

Легко заметить, что в нем состояния  $q_2$  и  $q_3$  недостижимы из начального состояния  $q_0$  и что этот автомат  $M$  распознает язык  $L_M = \phi^{-1}(L) = \{u \mid \text{слово } u \text{ содержит нечетное число букв } 0 \text{ и четное число букв } 1\}$ .

Рис. 14: Автомат  $A$ :  $L(A) = L$ Рис. 15: Диаграмма автомата  $M$ , распознающего язык  $\phi^{-1}(L)$ 

Имеется еще много операций, относительно которых замкнут класс автоматных языков. некоторые из них приведены в следующих задачах.

### 5.3 Задачи

**Задача 5.1.** Пусть  $M_r$  — это автомат, который строится в доказательстве теоремы 5.1 по регулярному выражению  $r$ . Докажите, что

- а) у  $M_r$  нет переходов из единственного заключительного состояния  $q_f$ ;
- б) в диаграмме  $M_r$  из каждой вершины выходит не более двух ребер;
- в) число состояний  $M_r$  не более чем вдвое превосходит длину выражения  $r$ , т.е.  $|Q| \leq 2|r|$ .

**Задача 5.2.** Примените процедуру детерминизации из теоремы 3.1 и постройте ДКА, эквивалентный НКА  $M$  из примера 5.1.

**Задача 5.3.** Примените процедуру детерминизации из теоремы 3.1 на стр. 13 и постройте ДКА, эквивалентный построенному выше в примере 5.3 на стр. 27 НКА  $M$ .

**Задача 5.4.** Цилиндрификация — это операция, которая обратна проекции. Для любых алфавитов  $\Delta$  и  $\Sigma$  таких, что  $\Delta \subset \Sigma$ , и любого языка  $L$  в алфавите  $\Delta$  определим его цилиндрификацию как язык  $CYL_{\Sigma}(L) = \{w \in \Sigma^* \mid \text{при вычеркивании из } w \text{ всех букв, не в } \Delta \text{ получается слово } u \in L\}$ .

Показать, что для автоматного языка  $L$  язык  $CYL_{\Sigma}(L)$  также является автоматным языком. Предложите процедуру перестройки автомата, распознающего  $L$ , в автомат, распознающий  $CYL_{\Sigma}(L)$ .

**Задача 5.5.** Обращением слова  $w = w_1w_2\dots w_k$  ( $w_i \in \Sigma, i = 1, \dots, k$ ) называется слово  $w^{-1} = w_k\dots w_2w_1$ . Показать, что для автоматного языка  $L$  его обращение — язык  $L^{-1} = \{w^{-1} \mid w \in L\}$  также является автоматным языком.

**Задача 5.6.** Пусть  $L$  — автоматный язык в алфавите  $\Sigma$ . Доказать, что автоматными являются и следующие языки:

- 1)  $\text{ПРЕФ}(L) = \{w \mid \text{существует такое слово } x \in \Sigma^*, \text{ что } wx \in L\}$ .
- 2)  $\text{СУФ}(L) = \{w \mid \text{существует такое слово } x \in \Sigma^*, \text{ что } xw \in L\}$ .
- 3)  $\text{КОР}(L) = \{w \mid \text{существуют такие слова } x, y \in \Sigma^*, \text{ что } xwy \in L\}$ .

**Задача 5.7.** Пусть  $L$  — автоматный язык в алфавите  $\Sigma = \{a_1, \dots, a_m\}$ , а  $L_1, \dots, L_m$  — это автоматные языки в алфавите  $\Delta$ . Доказать, что автоматным является и язык  $\text{ЗАМ}(L)$ , полученный из слов  $L$  заменой каждой буквы  $a_i$  на некоторое слово из  $L_i$ , т.е.

$\text{ЗАМ}(L) = \{w \mid \text{существует такое слово } u = a_{i_1}a_{i_2}\dots a_{i_n} \in L \text{ и такие слова } w_1, w_2, \dots, w_n \in \Delta^*, \text{ что } w = w_1w_2\dots w_n \text{ и } w_j \in L_{i_j} \text{ для всех } j = 1, 2, \dots, n\}$ .

**Задача 5.8.** Пусть  $L$  — автоматный язык в алфавите  $\Sigma$ ,  $k$  — целое положительное число и  $\phi$  — отображение  $\Sigma^k$  в  $\Sigma$ . Доказать, что автоматным является язык  $L_1 = \{\phi(a_1a_2\dots a_k)\dots\phi(a_{(n-1)k+1}a_{(n-1)k+2}\dots a_{nk}) \mid a_1a_2\dots a_{nk} \in L\}$ .

## 6 Теорема о разрастании автоматных языков. Неавтоматные языки

До сих пор мы встречались лишь с автоматными языками и накопили достаточно много средств для доказательства того, что некоторый язык является автоматным. Для этого, например, достаточно построить для него регулярное выражение или получить его с помощью различных рассмотренных выше операций из заведомо автоматных языков. В этом разделе мы установим некоторое необходимое условие, которому удовлетворяют все автоматные языки. После этого, проверив, что некоторый язык этому условию не удовлетворяет, можно заключить, что он не автоматный.

**Теорема 6.1.** (о разрастании автоматных языков)

Пусть  $L$  — бесконечный автоматный язык. Тогда существует такая константа  $n$ , что любое слово  $w \in L$  длины  $|w| > n$  можно разбить на три части  $x, y$  и  $z$  так, что  $w = xyz$  и

- 1)  $|xy| \leq n$ ;
  - 2)  $|y| > 0$ ;
  - 3) для любого  $m \geq 0$  слово  $w_m = xy^mz$  принадлежит языку  $L$ .
- (Здесь  $y^0 = \varepsilon$ ,  $y^1 = y$ ,  $y^{i+1} = y^iy$ ).

*Доказательство.* Так как язык  $L$  автоматный, то существует ДКА  $A = \langle \Sigma, Q, q_0, F, \Phi \rangle$ , распознающий  $L$ . Пусть  $|Q| = n$  и слово  $w = w_1 w_2 \dots w_k \in L$  имеет длину  $k > n$ . Рассмотрим путь  $p = (q_0 = q_{i_0}, q_{i_1}, \dots, q_{i_k})$  в диаграмме  $A$ , который несет слово  $w$ . Очевидно, что среди первых  $(n + 1)$  состояний этого пути хотя бы одно встречается дважды. Выберем первое из таких состояний  $q \in Q$ . Тогда для некоторой пары чисел  $l < j \leq n$  имеем  $q_{i_l} = q_{i_j} = q$ . Пусть  $x = w_1 w_2 \dots w_l$  — это префикс  $w$ , который переводит  $q_0$  в  $q_{i_l} = q$ ,  $y = w_{i_l+1} \dots w_{i_j}$  — это подслово  $w$ , которое переводит  $q_{i_l} = q$  в  $q_{i_j} = q$ , и  $z = w_{i_j+1} \dots w_k$  — это суффикс  $w$ , который переводит  $q_{i_j} = q$  в  $q_{i_k} \in F$ .  $x$  и  $z$  могут быть пусты, но  $|y| = j - l \geq 1$ . Длина  $|xy| = j \leq n$ . Таким образом, условия (1) и (2) теоремы выполнены. Нетрудно убедиться и в выполнении условия (3). Действительно, выбросив из пути  $p$  цикл  $q_{i_l} = q, \dots, q_{i_j}$ , получим путь  $p_0$  из  $q_0$  в  $q_{i_k} \in F$ , который несет слово  $xz$ , а повторив этот цикл  $m$  раз, получим путь  $p_0$  из  $q_0$  в  $q_{i_k} \in F$ , который несет слово  $xy^m z$ . Следовательно, для любого  $m \geq 0$   $w_m = xy^m z \in L$ .  $\square$

Содержательно, эта теорема утверждает, что у всякого достаточно длинного слова из автоматного языка имеется непустое подслово, которое можно вырезать или повторить сколько угодно раз, оставаясь внутри языка. Как, используя теорему 6.1 на стр. 31, доказать, что некоторый язык  $L$  не является автоматным? Это можно сделать, используя схему доказательства “от противного”:

- 1) Предположим, что  $L$  автоматный язык. Тогда для него имеется константа  $n$  из утверждения теоремы 6.1 на стр. 31.
- 2) Определим по  $n$  некоторое “специальное” слово  $w$  из  $L$  длины  $> n$  и докажем, что для любого разбиения  $w = xyz$ , удовлетворяющего условиям (1) и (2) теоремы, найдется такое  $k \geq 0$ , что слово  $w_k = xy^k z$  не принадлежит  $L$ .
- 3) На основании полученного противоречия делаем вывод, что  $L$  не автоматный язык.

Разумеется, в этой схеме самым сложным является выбор “специального” слова  $w$  в пункте (2). Что касается, подбора такого  $k \geq 0$ , для которого  $w_k \notin L$ , то, как правило, достаточно рассмотреть  $k = 0$  или  $k = 2$ .

Рассмотрим несколько примеров применения теоремы о разрастании.

**Пример 6.1.** Покажем, что язык  $L_1 = \{w = 0^i 1^i \mid i \geq 1\}$  не является автоматным.

Предположим, что он автоматный. Тогда для него имеется  $n$  из утверждения теоремы 6.1. Рассмотрим следующее (“специальное” !) слово  $w = 0^n 1^n$ . Очевидно, что  $w \in L_1$ . Предположим, что существует разбиение  $w = xyz$ , удовлетворяющего условиям (1) и (2) теоремы. Так как по условию (2)  $|xy| \leq n$ , то  $y = 0^i$  для некоторого  $i > 0$ . Но тогда слово  $w_0 = xz = 0^{n-i} 1^n \notin L_1$ , что противоречит условию (3) теоремы. Следовательно язык  $L_1$  не автоматный.

**Пример 6.2.** Покажем, что язык СКОБ правильных скобочных последовательностей в алфавите  $\{(, )\}$  не является автоматным.



Схема доказательства та же. В качестве специального слова выберем слово  $w = \binom{n}{n}^n$ , оно, очевидно, принадлежит СКОБ. Тогда для всякого разбиения  $w = xyz$  такого, что  $|xy| \leq n$  слово  $y = 0^i$  для некоторого  $i > 0$ . И, как и в предыдущем примере, слово  $w_0 = xz = \binom{n-i}{n}^n \notin \text{СКОБ}$ , что противоречит условию (3) теоремы. Следовательно, язык СКОБ не автоматный.

**Пример 6.3.** Покажем, что язык  $L_2 = \{w = 0^i 1^j \mid i \leq 2j + 1\}$  не является автоматным.

Здесь, предположив, что  $L_2$  автоматный язык и зафиксировав константу  $n$  из теоремы 6.1, рассмотрим слово  $w = 0^{2n+1} 1^n \in L_2$ . Для всякого разбиения  $w = xyz$  такого, что  $|xy| \leq n$  слово  $y = 0^i$  для некоторого  $i > 0$ . Рассмотрим слово  $w_2 = xy^2 z = 0^{2n+1+i} 1^n$ . Но  $2n + 1 + i \geq 2n + 2 \not\leq 2n + 1$ . Следовательно,  $w_2 \notin L_2$  и язык  $L_2$  не является автоматным.

**Пример 6.4.** Рассмотрим язык “квадратов” в унарном алфавите  $\{|\}$ :  
 $L_3 = \{|^{i^2} \mid i \geq 0\}$ .

Здесь, предположив, что  $L_3$  автоматный язык и зафиксировав константу  $n$  из теоремы 6.1, рассмотрим слово  $w = |^{n^2}$ . Для всякого разбиения  $w = xyz$  такого, что  $|xy| \leq n$  слово  $y = |^i$  для некоторого  $0 < i \leq n$ . Тогда  $w_0 = xz = |^{n^2-i}$ . Но  $n^2 - i \geq n^2 - n > n^2 - 2n + 1 = (n - 1)^2$ . Следовательно,  $n^2 - i$  не является полным квадратом и  $w_0 \notin L_3$ , т.е. язык “квадратов”  $L_3$  не является автоматным.

**Пример 6.5.** Рассмотрим язык “простых чисел” в унарном алфавите  $\{|\}$ :  
 $L_{pr} = \{|^p \mid p - \text{простое число}\}$ .

Предположим, что  $L_{pr}$  — автоматный язык и зафиксируем для него константу  $n$  из теоремы 6.1. Выберем простое число  $p > n$  и рассмотрим слово  $w = |^p$ . Пусть  $w = xyz$  — произвольное разбиение  $w$  такое, что  $|xy| \leq n$ . Тогда для некоторого  $0 < i \leq n$  слово  $y = |^i$  и  $xz = |^{p-i}$ . Положим  $k = p - i$  и рассмотрим слово  $w_k = xy^k z$ . Его длина  $p'$  равна  $|x| + k|y| + |z| = (p - i)(i + 1)$ . Так как  $1 \leq i < n + 1 \leq p$ , то  $p'$  — составное число и  $w_k \notin L_{pr}$ . Следовательно,  $L_{pr}$  — не автоматный язык. Заметим, что в этом примере  $k$  выбирается для каждого  $n$  по-своему.

Еще один прием доказательства неавтоматности языка  $L$  состоит в том, чтобы вместо  $L$  рассмотреть некоторый язык  $L' = op(L, L_1, \dots, L_k)$ , полученный из  $L$  и автоматных языков  $L_1, \dots, L_k$  с помощью операций  $op$ , сохраняющих автоматность. Если доказать, что  $L'$  не является автоматным, то и исходный язык  $L$  не автоматен.

**Пример 6.6.** Рассмотрим язык  $L_4 = \{0^i 1^j \mid i \neq j\}$ .

Пусть  $L_5 = \{0^i 1^j \mid i \geq 1, j \geq 1\}$ . Очевидно, что язык  $L_5$  автоматный. Нетрудно заметить, что его пересечение с дополнением  $L_4$  совпадает с языком  $L_1$  из примера 6.1, т.е.  $L_1 = L_5 \cap \overline{L_4}$ . Так как мы установили, что  $L_1$  не автоматный, то и  $L_4$  не является автоматным.

Являются ли условия теоремы 6.1 достаточными для того, чтобы язык оказался автоматным? Следующий пример показывает, что ответ на этот вопрос отрицателен.

**Пример 6.7.** Пусть  $L_6 = \{c^r a^i b^i \mid r \geq 1, i \geq 0\}$ ,  $L_7 = \{a^i b^j \mid i \geq 0, j \geq 0\}$ . Рассмотрим язык  $L_8 = L_6 \cup L_7$ .

Для этого языка можно в качестве  $n$  выбрать 1. Каждое слово  $w$  из  $L_8$  принадлежит  $L_6$  или  $L_7$ . Если слово  $w = c^r a^i b^i \in L_6$ , то оно представимо в виде  $xyz$ , где  $x = \varepsilon$ ,  $y = c$ ,  $z = c^{r-1} a^i b^i$  ( $r \geq 1, i \geq 0$ ). Тогда  $w_0 = z = c^{r-1} a^i b^i$  ( $r \geq 1, i \geq 1$ ) и при  $r = 1$  слово  $w_0 = a^i b^i \in L_7$ , а при  $r > 1$ , очевидно,  $w_0 \in L_6$ . При  $k \geq 1$  имеем  $w_k = c^{r+k-1} a^i b^i \in L_6$ . Если слово  $w = a^i b^j \in L_7$  и  $i \geq 1$ , то его можно представить как в виде  $xyz$ , где  $x = \varepsilon$ ,  $y = a$ ,  $z = a^{i-1} b^j$  ( $i \geq 1, j \geq 0$ ) и для каждого  $k \geq 0$   $w_k = a^k a^{i-1} b^j \in L_7$ . Если же  $i = 0$ , то  $w = b^j$  ( $j \geq 1$ ) и его можно разбить на части  $x = \varepsilon$ ,  $y = b$ ,  $z = b^{j-1}$  ( $j \geq 1$ ). И в этом случае для каждого  $k \geq 0$   $w_k = b^k b^{j-1} \in L_7$ . Во всех случаях  $w_k \in L_8$  и, следовательно язык  $L_8$  удовлетворяет условиям теоремы 6.1. Но этот язык не автоматный. Действительно, пусть  $\phi : \{a, b, c\}^* \rightarrow \{0, 1\}^*$  — это гомоморфизм, заданный как  $\phi(a) = 0, \phi(b) = 1, \phi(c) = \varepsilon$ . Тогда  $\phi(L_8 \setminus L_7) = \phi(L_6) = L_1$  из примера 6.1. Так как язык  $L_7$  является автоматным, а  $L_1$  — нет, то и язык  $L_8$  не является автоматным.

## 6.1 Задачи

**Задача 6.1.** Докажите, что теорема 6.1 о разрастании остается справедливой и при замене условия 1)  $|xy| \leq n$  на условие 1')  $|yz| \leq n$ , т.е. повторяющееся подслово  $y$  имеется и в суффиксе  $w$  длины  $\leq n$ .

**Задача 6.2.** Доказать, что следующие языки в алфавите  $\Sigma = \{a, b, c\}$  не являются автоматными.

- а) Множество всех слов, в которых букв  $a$  на 3 больше, чем букв  $b$ .
- б)  $L = \{a^n c b^m \mid m > 3n\}$ .
- в)  $L = \{w c w^{-1} \mid w = a^2 b^n a \text{ для некоторого } n > 0\}$ .
- г)  $L = \{w \mid |w| = 2^n \text{ для некоторого целого числа } n\}$ .
- д)  $L = \{w c^{|w|} \mid w \in \{a, b\}^*, |w| — \text{длина слова } w\}$ .

**Задача 6.3.**  $\lambda$ -выражение — это либо переменная  $x$ , или символ  $\lambda$ , за которым следует переменная, а далее либо  $\lambda$ -выражение, либо левая скобка,  $\lambda$ -выражение, еще одно  $\lambda$ -выражение и правая скобка. Например,  $\lambda x x, \lambda x (x x), \lambda x \lambda x (\lambda x (x x) \lambda x (x x))$  — это правильные  $\lambda$ -выражения, а  $(x x), \lambda x (\lambda x)$  и  $\lambda x ((x x))$  — неправильные.

Докажите, что язык  $\lambda$ -выражений в алфавите  $\{x, \lambda, (, )\}$  не является автоматным.

**Задача 6.4.** Выше в задаче 2.3 на стр. 10 строился автомат-распознаватель, который проверял правильность сложения двоичных чисел. Докажите, что для операции умножения двоичных чисел такого автомата не существует, т.е. что язык в алфавите троек битов

$U = \{(x_1(1), x_2(1), y(1))(x_1(2), x_2(2), y(2)) \dots (x_1(n), x_2(n), y(n)) \mid y = y(n) \dots y(2)y(1) \\ \text{— это первые } n \text{ битов произведения двоичных чисел } x_1 = x_1(n) \dots x_1(2)x_1(1) \text{ и } x_2 = x_2(n) \dots x_2(2)x_2(1)\}$

не является автоматным.

**Задача 6.5.** Доказать, что язык  $L = \{w \mid \text{число букв } a \text{ в } w \neq \text{число букв } b \text{ в } w\}$  в алфавите  $\Sigma = \{a, b\}$  не является автоматным.