Министерство образования Российской Федерации

Нижегородский государственный университет

им. Н.И. Лобачевского

Факультет вычислительной математики и кибернетики

Учебно-методическая разработка для самостоятельной работы студентов по курсу «Теория алгоритмов и математическая логика» при изучении темы «Концепции конечного автомата и регулярного языка. Операции над регулярными языками»

Нижний Новгород 2000 г.

УДК 518.5

Методическая разработка предназначена для самостоятельной работы студентов специальности «Прикладная информатика» над материалом темы «Концепции конечного автомата и регулярного языка. Операции над регулярными языками», входящей в состав учебного курса «Теория алгоритмов и математическая логика». Вводятся понятие формального языка и действия над формальными языками, включая основные теоретико-множественные операции. Излагается концепция конечного автомата (в детерминированном и недетерминированном вариантах); регулярные языки представляют собой класс языков, распознаваемых конечными автоматами. Показывается, что операции, объединения, пересечения, дополнения, конкатенации и итерации не выводят из класса регулярных языков. Приводятся соответствующие алгоритмы синтеза конечных автоматов.

Составители:
преподаватели кафедры информатики и автоматизации научных исследований факультета ВМК ННГУ доцент, д.т.н. Коган Д.И. и ассистент Бабкина Т.С.

1. Понятие языка, примеры языков, операции над языками.

Алфавитом именуем произвольное непустое конечное множество символов; символы произвольного алфавита называем его ***буквами***. В качестве примеров укажем русский алфавит (с включением или невключением в него знаков препинания), латинский алфавит, объединение указанных алфавитов, множество цифр десятичной или двоичной систем счисления. В общем виде алфавит определяем как совокупность $A=\{a_1, a_2, ..., a_n\}$; в числе букв алфавита A может быть и специальный символ пробела (пустой буквы), этот символ обычно обозначается e (сокращение английского «empty» – пустой).

Словом в алфавите A называем произвольную конечную последовательность его букв; одна и та же буква в этой последовательности может встречаться многократно. ***Длиной*** слова α (обозначение $l(\alpha)$) называем количество букв в этом слове. Специальным символом λ будем обозначать единственное слово, имеющее нулевую длину (***пустое слово***); следует различать пустое слово и слово e, состоящее из одной пустой буквы; длина слова e (пробела) равна единице. В алфавите $A=\{a_1, a_2, ..., a_n\}$ можно записать n^l различных слов длины l, где l=0, 1, 2, ... Множество всех слов алфавита A будем обозначать A^*. Специально отметим, что в совокупность A^* входит и пустое слово. Мощность множества всех слов алфавита A счетна.

Если α и β – два произвольных слова в алфавите A, то $\alpha\beta$ – результат приписывания справа слова β к слову α. Для любых слов α и β считаем, что $\alpha\lambda=\lambda\alpha=\alpha$, $\alpha\lambda\beta=\alpha\beta$.

Языком в алфавите A называем произвольное подмножество слов L из A^*. Язык L именуем ***конечным***, если в его составе конечное множество слов; язык L ***бесконечен***, если в его составе бесконечное множество слов. Совокупности всех слов русского, всех слов английского языка представляют собой примеры конечных языков. Множество записей всех простых чисел в десятичной или двоичной системе счисления является бесконечным языком. Множество всех языков алфавита A имеет континуальную мощность.

Язык L в алфавите A называем ***универсальным***, если $L=A^*$. Язык L называем ***пустым***, если множество L пусто ($L=\varnothing$).

Пусть L_1 и L_2 – языки в алфавите A. Через $L_1\cup L_2$ и $L_1\cap L_2$ будем обозначать ***объединение*** и ***пересечение*** этих языков соответственно. Слово α принадлежит объединению двух языков, если оно принадлежит по меньшей мере одному из них; слово α принадлежит пересечению двух языков, если оно принадлежит как одному, так и другому языку. Пусть L – язык в алфавите A; через L^c будем обозначать ***дополнение*** этого языка. Язык L^c образуют слова алфавита A, не входящие в состав языка L: $L^c=A^*\backslash L$. Операции объединения, пересечения и дополнения – основные теоретико-множественные операции.

Пусть L_1 и L_2 – языки в алфавите A. Через $L_1\backslash L_2$ будем обозначать ***разность*** языков L_1 и L_2 . Слово α из A^* считается принадлежащим $L_1\backslash L_2$ тогда и только тогда, когда оно принадлежит языку L_1, но не принадлежит языку L_2. Очевидно, что операция разности представима через основные теоретико-множественные операции: $L_1\backslash L_2=L_1\cap(L_2)^c$.

Дополнительно введем несколько специальных операций над языками. Пусть L_1 и L_2 – языки в алфавите A. Через $L_1\bigcirc L_2$ обозначим язык, определяемый следующим образом: слово α принадлежит языку $L_1\bigcirc L_2$ тогда и только тогда, когда это слово можно разбить на две последовательные части (т.е. представить в виде $\alpha=\alpha_1\alpha_2$) так, что первая часть является словом языка L_1, а вторая часть – словом языка L_2. Операция $\bigcirc$ называется ***конкатенацией*** (сцепкой) языков. Знак $\bigcirc$ нередко будет опускаться, тогда конкатенация языков L_1 и L_2 записывается L_1L_2. Язык L_1L_2 получается приписыванием к словам языка L_1 слов языка L_2 в качестве окончаний. Отметим, что если к произвольному слову γ приписать в качестве окончания пустое слово, то результатом будет слово γ. Если языки L_1 и L_2 конечны, причем в составе первого языка m слов, а в составе второго n слов, то язык L_1L_2 состоит максимум из mn слов. Если один из языков L_1, L_2 пуст, то L_1L_2 – также пустой язык.

Введем операцию возведения языка в степень. Полагаем:

$$L^0=\{\lambda\};$$

$$L^1=L;$$

$$L^2=LL;$$

$$L^{n+1}=L^nL,\ n=2, 3, \ldots;$$

таким образом, понятие степени L^i языка L определено для любого i=0, 1, 2, 3, ...

Далее через L^* обозначим объединение всех степеней языка L:

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

Введенную операцию именуем ***итерацией***. Отметим, что пустое слово принадлежит итерации любого языка. Непустое слово α принадлежит итерации языка L тогда и только тогда, когда это слово можно разбить на некоторое количество последовательных частей (подслов) так, что каждая часть принадлежит языку L. Если язык L состоит из всех однобуквенных слов алфавита A, то итерацией этого языка является универсальный язык A^*. Отметим, что для любого языка L имеет место $(L^*)^*=L^*$.

На множестве языков иногда рассматривают и следующую одноместную операцию $(\ )^+$:

$$(L)^+ = \bigcup_{i=1}^{\infty} L^i.$$

Языки L^* и L^+ отличаются один от другого разве что пустым словом. Слово λ всегда принадлежит языку L^*. Языку L^+ оно принадлежит тогда и только тогда, когда принадлежит языку L.

2. Концепции конечного автомата и регулярного языка.

В кибернетике автоматами называют технически или программно реализованные модули, предназначенные для переработки поступающей информации. ***Конечный автомат*** – это модуль, имеющий конечное число возможных состояний и функционирующий в дискретном времени. В данном пособии конечные автоматы изучаются как абстрактные алгоритмические устройства, предназначенные для обработки слов фиксированного входного алфавита. Считаем, что обработку произвольного слова α во входном алфавите A автомат начинает из специально выделенного начального состояния. В каждый такт дискретного времени на вход автомата поступает очередная буква обрабатываемого слова, под ее воздействием автомат меняет свое состояние; состояние, в которое автомат перейдет, определяется предыдущим его состоянием и прочитанной буквой входного слова. Над словом длины l автомат работает l тактов. Результат работы автомата определяется состоянием, в котором он оказывается по ее завершении.

Формально конечный автомат K определяем как совокупность

$$K=\{Q, A, q_0, g, F\},$$

где $Q=\{q_0, q_1, q_2, ..., q_m\}$ – множество состояний автомата; $A=\{a_1, a_2, ..., a_n\}$ – входной алфавит; q_0 – специально выделенное состояние автомата, именуемое начальным состоянием; g – функция переходов конечного автомата, представляющая собой отображение типа Qx$A\to Q$ (если $g(q_i,a_j)=q_k$, то автомат из состояния q_i под воздействием буквы a_j должен перейти в состояние q_k); F – специально выделенное подмножество состояний автомата, которые мы условно будем именовать «хорошими», $F\subseteq Q$.

Пусть $\alpha = a_{i_1}a_{i_2}....a_{i_p}$ – входное слово, $l(\alpha)=p$. Через $q_\alpha(t)$ обозначим состояние, в котором оказывается автомат K через t тактов работы над этим словом (здесь t=0, 1, 2, ..., p):

$$q_\alpha(0)=q_0;$$

$$q_\alpha(1) = g(q_\alpha(0), a_{i_1})$$

$$q_\alpha(2) = g(q_\alpha(1), a_{i_2})$$

. . .

$$q_\alpha(p) = g(q_\alpha(p-1), a_{i_p})$$

Будем говорить, что слово α ***принимается*** автоматом *К*, если $q_\alpha(p) \in F$. Введем в рассмотрение язык $L(К)$: слово α принадлежит языку $L(К)$, если данное слово принимается автоматом *К*. Язык $L(К)$ именуем языком, распознаваемым данным конечным автоматом. Язык L назовем ***регулярным***, если для него можно построить распознающий конечный автомат.

Конечный автомат удобно задавать ***диаграммой*** его ***переходов***. Диаграмма представляет собой ориентированный граф, вершины которого одноименны состояниям автомата; дуга из вершины q_i, в вершину q_k с надписанной над ней буквой a_j проводится тогда и только тогда, когда $g(q_i,a_j)=q_k$. В случае, когда переход из q_i в q_k осуществляется под воздействием любой из букв некоторого подмножества S, $S \subseteq A$, все буквы этого подмножества подписываются над соответствующей дугой (вместо перечня всех букв допускается сокращенная запись «$x \in S$» или просто «S»). Если произвольное состояние q_i входит в F, то данный факт на диаграмме отмечается жирным кружком, выделяющим вершину q_i.

Очевидно, что любой конечный автомат полностью определяется своей диаграммой переходов. Далее под задачей построения конечного автомата, обладающего теми или иными свойствами, будет пониматься задача построения диаграммы его переходов.

На рис. 2.1 показана диаграмма автомата K_1, работающего над словами алфавита $A=\{a,b,c\}$. Автомат имеет два состояния, q_0 и q_1, причем «хорошим» является только состояние q_1. Начав работу в состоянии q_0, автомат под воздействием букв *a, b* из этого состояния не выходит; под воздействием буквы *c* реализуется переход в состояние q_1; любая далее поступающая буква оставляет автомат в том же состоянии. Таким образом, автомат K_1 распознает язык L_1, состоящий из слов, имеющих в своем составе букву *c*. Данный язык является регулярным.

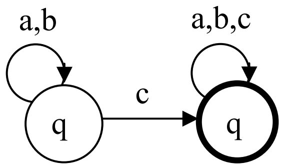


Рис. 2.1

Приведем еще несколько примеров регулярных языков в том же алфавите: L_2 - множество слов, в которых буква a встречается четное число раз; L_3 - множество слов, начинающихся и заканчивающихся одинаковой буквой; L_4 - множество слов, содержащих подслово $\alpha=abbc$; L_5 - множество слов, при чтении которых слева направо разность между числом встреченных букв a и b никогда не превосходит 2. Диаграммы конечных автоматов K_2 - K_5, распознающих языки L_2 – L_5 соответственно, представлены на рисунках 2.2 – 2.5. Информацию об обработанной части входного слова автомат “помнит” своим состоянием. Так, например, автомат K_5, обработав некоторую часть входного слова, оказывается в состоянии q_x, если в прочитанной им части входного слова число встреченных букв a превышает число встреченных букв b на x; здесь $x\in\{-2, -1, 0, 1, 2\}$; автомат оказывается в состоянии $q_{\text{плох}}$, если в некоторый момент работы автомата абсолютная величина разности между числом обработанных букв a и числом обработанных букв b превысила 2.

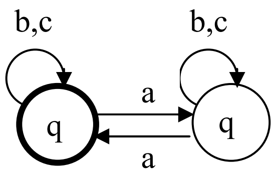


Рис. 2.2

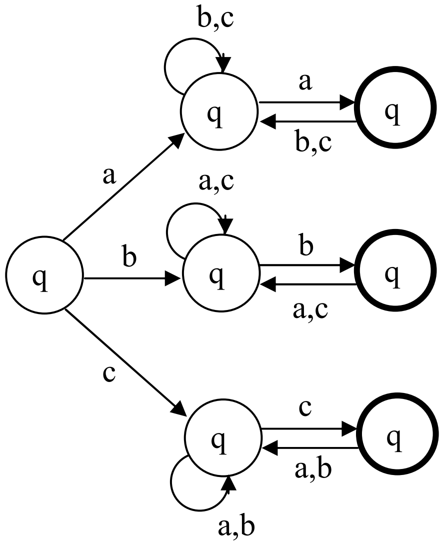


Рис. 2.3

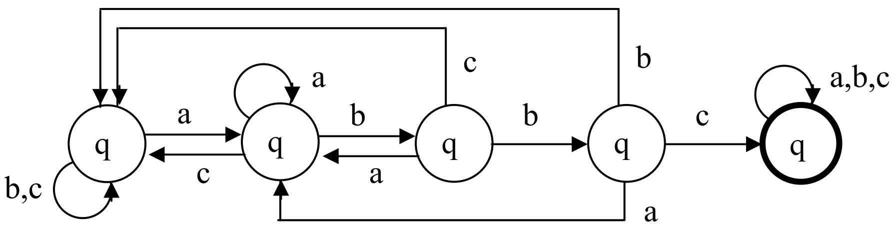


Рис. 2.4

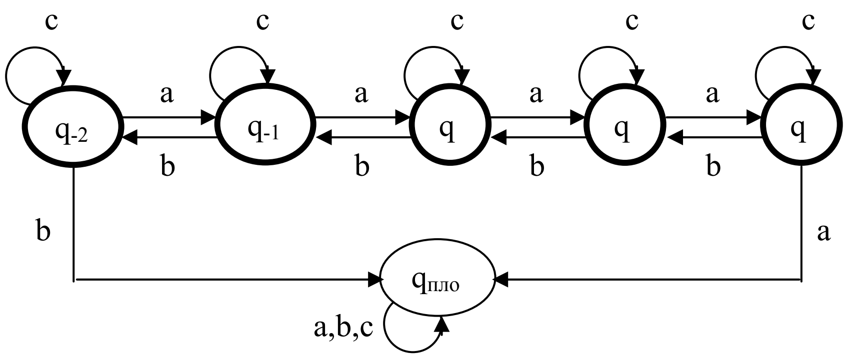


Рис. 2.5

Состояние конечного автомата назовем ***поглощающим***, если любая входная буква не выводит автомат из этого состояния. Поглощающее состояние назовем ***положительно поглощающим***, если оно принадлежит F, и ***отрицательно поглощающим*** в противном случае. В автоматах K_1 и K_4 полижительно поглощающими являются состояния q_1 и q_4 соответственно, в автомате K_5 отрицательно поглощающим является состояние $q_{\text{плох}}$.

Можно считать, что каждый автомат имеет не более одного положительно поглощающего и не более одного отрицательно поглощающего состояния (если положительно или отрицательно поглощающих состояний несколько, то они легко могут быть заменены одним). Обычно в диаграмме переходов отрицательно поглощающее состояние не изображается; если из некоторого состояния по некоторой букве переход не указан, то предполагается, что он ведет в отрицательно поглощающее состояние. Представленный на рисунке 2.6 конечный автомат распознает в алфавите A язык, состоящий из слов, в которые буква c входит ровно один раз. Этот автомат имеет 3 состояния, отрицательно поглощающее состояние $q_{\text{плох}}$ (в него автомат переходит из состояния q_1 по букве c) в диаграмме не указано.

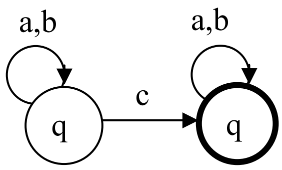


Рис. 2.6

Введем алфавит B={0, 1, 2, …, 9}, каждое слово в этом алфавите трактуем как целое неотрицательное число. Пусть $L^{(p)}$ обозначает множество слов – записей в десятичной системе счисления всех целых неотрицательных чисел, кратных натуральной константе p; будем считать, что языку $L^{(p)}$ дополнительно принадлежит также пустое слово λ. Покажем, что $L^{(p)}$ – регулярный язык. Распознающий $L^{(p)}$ конечный автомат $K^{(p)}$ можно построить следующим образом. Состояниями автомата считаем q_0, q_1, q_2, q_{p-1}; из произвольного состояния q_i по цифре x, $x\in\{0, 1, 2, ... ,9\}$, автомат переходит в состояние q_j, если остаток от деления числа ix (т.е. числа, получаемого приписыванием к числу i цифры x справа) на p равен j. Благодаря указанной структуре переходов автомат, обработавший, начиная от начального состояния q_0, записанное в десятичной системе счисления целое неотрицательное число α, оказывается в состоянии q_y тогда и только тогда, когда остаток от деления α на p равен y. Единственным «хорошим» состоянием автомата $K^{(p)}$ следует считать q_0. На рис. 2.7 и 2.8 представлены диаграммы конечных автоматов, распознающих числа, делящиеся на 2 и на 3 соответственно.

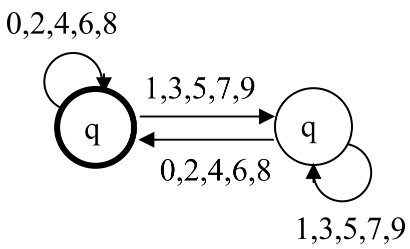


Рис. 2.7.

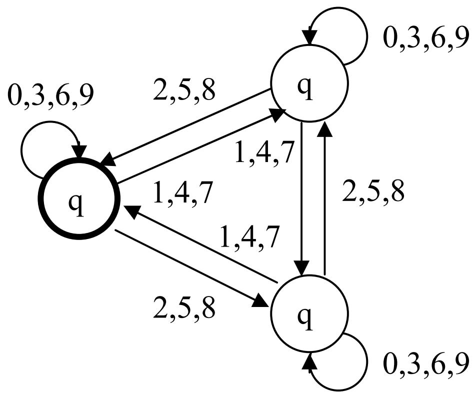


Рис. 2.8.

Отметим, что в ряде случаев, исходя из иных принципов, можно построить распознающие делимость на p автоматы с меньшим числом состояний. На рис. 2.9 представлен имеющий два состояния конечный автомат, распознающий числа, кратные пяти (используется тот факт, что кратное пяти число имеет своей последней цифрой 0 или 5).

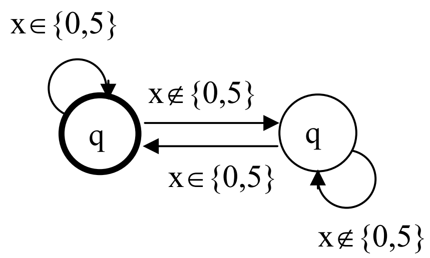


Рис. 2.9.

По конечному автомату $K^{(p)}$ легко строится автомат $K^{(p)}[\lambda]$, распознающий язык $L^{(p)}\backslash\{\lambda\}$. Для получения диаграммы переходов автомата $K^{(p)}[\lambda]$ в имеющейся диаграмме переходов автомата $K^{(p)}$ делаются следующие изменения: начальное состояние q_0 автомата $K^{(p)}$ переименовываем в q_0'; вводим новое начальное состояние q_0; по произвольной цифре x из q_0 автомат $K^{(p)}[\lambda]$ реализует переход в то же состояние, что и автомат $K^{(p)}$ из своего начального состояния (при этом вместо перехода в q_0 всегда реализуется переход в q_0'); единственным «хорошим» состоянием автомата $K^{(p)}[\lambda]$ следует считать q_0'. Согласно изложенному, начальное состояние автомата $K^{(p)}[\lambda]$ является невозвратным – в процессе обработки любого входного слова, выйдя из этого состояния, автомат в него никогда не вернется. На рис.2.10 представлен конечный автомат $K^{(3)}[\lambda]$, распознающий числа, кратные трем (пустое слово, в язык, распознаваемый данным автоматом, не входит).

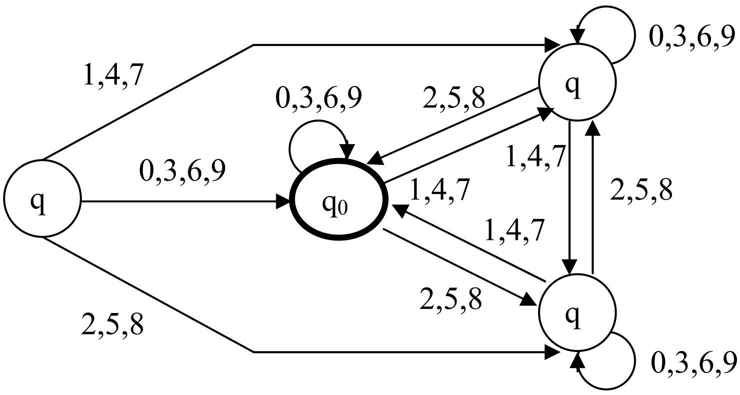


Рис. 2.10.

Отметим, что язык, распознаваемый произвольным конечным автоматом, содержит пустое слово тогда и только тогда, когда начальное состояние этого автомата принадлежит множеству F.

Теорема 2.1. Пустой язык, универсальный язык, любой конечный язык в произвольном фиксированном алфавите $A=\{a_1, a_2, \dots , a_n\}$ являются регулярными языками.

Любой конечный автомат с пустым множеством «хороших» состояний F распознает пустой язык. Любой конечный автомат, в котором каждое состояние «хорошее», распознает универсальный язык. Конечные автоматы, распознающие пустой и универсальный языки и имеющие только одно состояние, представлены на рис. 2.11 и 2.12 соответственно (x- произвольная буква алфавита A).

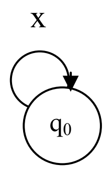


Рис. 2.11.

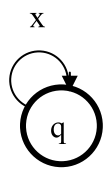


Рис. 2.12.

Сейчас предположим, что язык L конечен, пусть $L=\{\alpha_1,\ \alpha_2,\ \dots\ ,\ \alpha_p\}$. Диаграмму конечного автомата $K(L)$, распознающего слова языка L, строим в виде дерева, корнем которого является вершина q_0, образующая нулевой уровень. В вершине q_0 реализуем ветвление (далее эта операция будет выполняться и для других вершин): проводим из данной вершины n дуг, над первой надписываем a_1, над второй – a_2, ... , над n-ой – a_n, концами этих дуг являются n вершин следующего (в данном случае – первого) уровня. Выполнив ветвление в каждой из вершин первого уровня, получаем n^2 вершин второго уровня; выполнив ветвление в каждой из вершин второго уровня, получаем n^3 вершин третьего уровня, и т.д. Процесс продолжается до тех пор, пока не будут построены вершины r-го уровня, где r – число букв в самом длинном слове языка L. Операции ветвления в вершинах r-го уровня не выполняются; если на вход автомата, находящегося в некотором состоянии из r-го уровня, поступает еще одна буква, он переходит в неотображенное диаграммой отрицательно поглощающее состояние $q_{\text{плох}}$. Между вершинами-состояниями построенного r-уровневого дерева и имеющими длину не больше, чем r, словами алфавита A существует взаимно однозначное соответствие – каждому такому слову сопоставляется состояние, в котором оказывается автомат, обработавший, начиная от q_0, данное слово. Каждое отображенное в дереве состояние автомата $K(L)$ называем именем соответствующего ему слова (состояние q_0 получает при этом имя пустого слова). Множество F «хороших» состояний автомата $K(L)$ считаем следующим: α_1, α_2, ... α_p. Построение распознающего язык L конечного автомата закончено. Теорема доказана полностью.

В приведенном доказательстве изложен универсальный алгоритм синтеза автомата, распознающего конечное множество слов. Отметим, что в конкретных задачах автоматы, распознающие те или иные конечные языки, нередко строятся существенно проще. На рис. 2.13 представлен автомат, распознающий конечный язык A={0, 01, 10, 001, 101, 1100}.

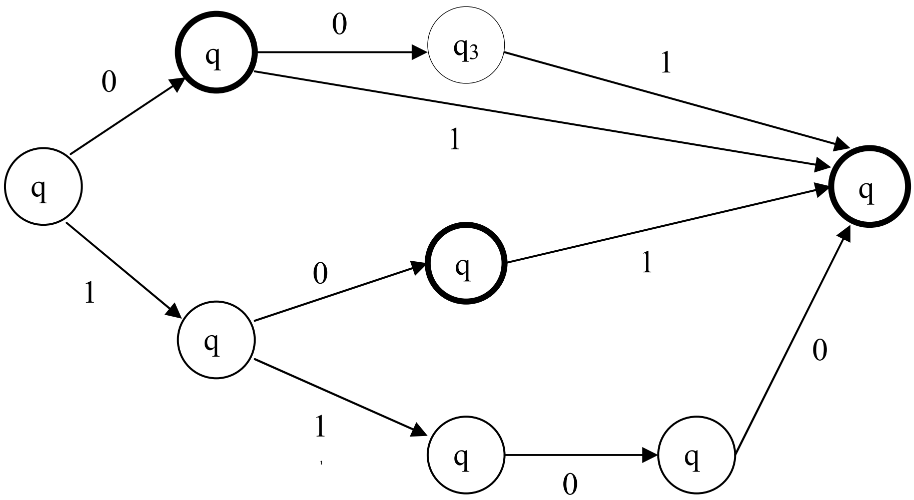


Рис. 2.13.

Теорема 2.2. Класс регулярных языков замкнут относительно основных теоретико - множественных операций – объединения, пересечения и дополнения.

Приводимое доказательство конструктивно – излагаются алгоритмы построения соответствующих автоматов. Пусть L_1 и L_2 – регулярные языки, распознаваемые конечными автоматами $K^1=\{Q^1, A, q^1_0, g^1, F^1\}$ и $K^2=\{Q^2, A, q^2_0, g^2, F^2\}$ соответственно; считаем, что $Q^1=\{q^1_0, q^1_1, q^1_2, \ldots, q^1_f\}$ и $Q^2=\{q^2_0, q^2_1, q^2_2, \ldots, q^2_h\}$. Автомат $K^{\cup}=\{Q, A, q_0, g, F^{\cup}\}$, распознающий язык $L_1 \cup L_2$, строим следующим образом. Полагаем $Q=Q^1$xQ^2; каждое состояние конструируемого автомата содержит две компоненты, левую и правую. Начальным состоянием нового автомата считаем (q^1_0, q^2_0), а функцию переходов g определяем следующим образом:

$$g((q^1_i, q^2_j), a_k) = (g^1(q^1_i, a_k), g^2(q^2_j, a_k))$$

Очевидно, по первой компоненте автомат $K^{\cup}$ моделирует действия автомата K^1, а по второй компоненте – действия автомата K^2. Входное слово α принадлежит объединению языков L_1 и L_2 тогда и только тогда, когда после его обработки автомат $K^{\cup}$ окажется в состоянии, у которого либо первая компонента принадлежит совокупности F^1, либо вторая компонента принадлежит совокупности F^2.Таким образом, следует положить:

$$F^{\cup}=(F^1 \text{x} Q^2)\cup(Q^1 \text{x} F^2).$$

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

Автомат $K^{\cap}=\{Q,\ A,\ q_0,\ g,\ F^{\cap}\}$, распознающий язык $L_1\cap L_2$, отличается от $K^{\cup}$ только последней своей компонентой. Следует положить

$$F^{\cap}=F^1 \text{x} F^2.$$

Пусть $K=\{Q,\ A,\ q_0,\ g,\ F\}$ – конечный автомат, распознающий язык L. Произвольное слово α из A^* принадлежит языку $L^c=A^*\backslash L$ тогда и только тогда, когда после его обработки автомат K оказывается в состоянии, не принадлежащем F. Поэтому автоматом, распознающим язык L^c, является конечный автомат $K^c=\{Q,\ A,\ q_0,\ g,\ Q\backslash F\}$. Образно говоря, для того, чтобы получить конечный автомат, распознающий дополнение регулярного языка, надо в автомате, распознающем исходный язык, поменять местами «хорошие» и «плохие» состояния.

Теорема доказана.

На рис. 2.14 представлены два конечных автомата, распознающих языки L_1 и L_2. На рис. 2.15 и 2.16 изображены автоматы, распознающие языки $L_1\cup L_2$ и $L_1\cap L_2$ соответственно.

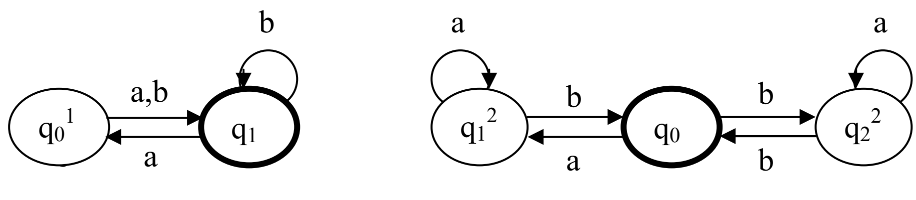


Рис. 2.14

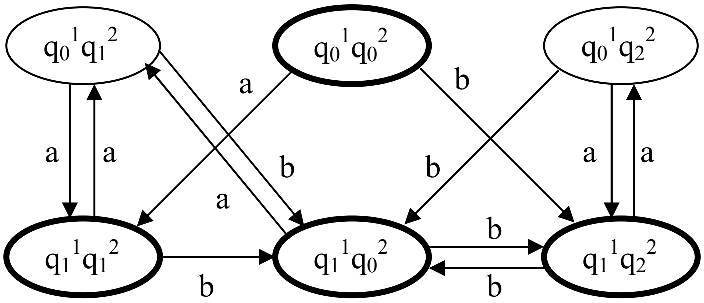


Рис. 2.15

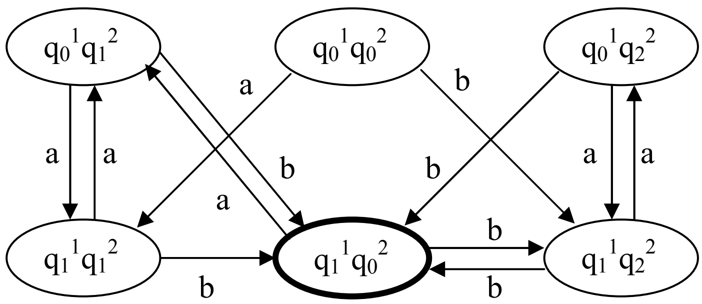


Рис. 2.16

На рис. 2.17 представлен автомат, распознающий язык L_3, представляющий собой дополнение к языку L_2. На рис. 2.18 изображен автомат, распознающий разность языков L_1 и L_2 (или пересечение языков L_1 и L_3).

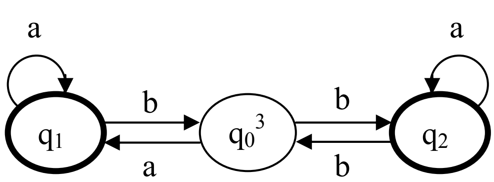


Рис. 2.17

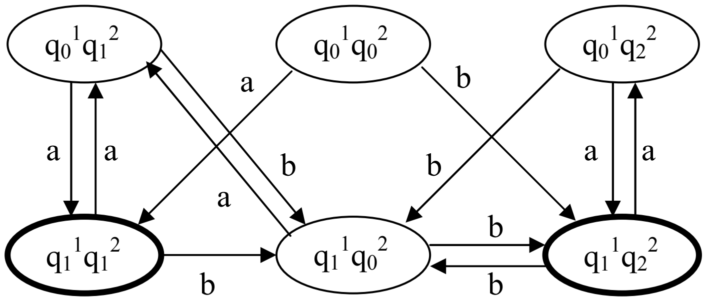


Рис. 2.18

Заметим, что часто автоматы, распознающие объединение и пересечение языков, строятся существенно проще. Это связано с тем, что некоторые состояния из множества состояний $Q=Q^1 \text{x} Q^2$ могут быть недостижимыми. Рассмотрим следующий пример. Пусть требуется построить автомат, распознающий объединение языков L_1 и L_2; автоматы K_1 и K_2, распознающие эти языки, изображены на рис. 2.19.

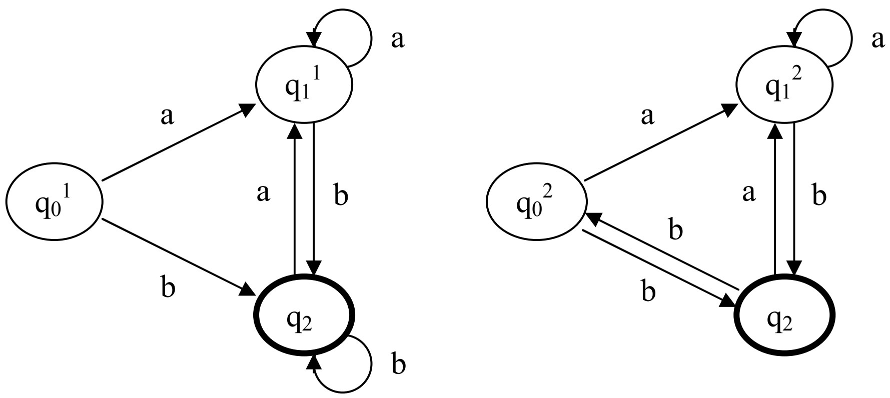


Рис. 2.19

Диаграмму переходов автомата $K^{\cup}$ (рис. 2.20), распознающего язык $L_1 \cup L_2$, строим следующим образом. Вводим вершину, соответствующую начальному состоянию (q_0^1, q_0^2). По букве a автомат K_1 из состояния q_0^1 переходит в состояние q_1^1, а автомат K_2 – из q_0^2 в q_1^2. Следовательно, автомат $K^{\cup}$

из состояния $(q_0{}^1,q_0{}^2)$ перейдет в состояние $(q_1{}^1,q_1{}^2)$. Добавим в диаграмму соответствующую вершину и ведущую в нее дугу. Т.к. по букве b автомат K_1 переходит из $q_0{}^1$ в $q_2{}^1$, а автомат K_2 – из $q_0{}^2$ в $q_2{}^2$, то автомат $K^{\cup}$ из состояния $(q_0{}^1,q_0{}^2)$ перейдет в состояние $(q_2{}^1,q_2{}^2)$. Соответствующая вершина и дуга добавляются в диаграмму. Рассмотрим вершину $(q_2{}^1,q_2{}^2)$. По букве a автомат K_1 из $q_2{}^1$ переходит в $q_1{}^1$, а автомат K_2 – из $q_2{}^2$ в $q_1{}^2$, следовательно; автомат $K^{\cup}$ из состояния $(q_2{}^1,q_2{}^2)$ перейдет в состояние $(q_1{}^1,q_1{}^2)$, которое уже присутствует на диаграмме. Поэтому в диаграмму добавляется только соответствующая дуга. По букве b автомат K_1 остается в состоянии $q_2{}^1$, а автомат K_2 переходит из $q_2{}^2$ в $q_0{}^2$, поэтому автомат $K^{\cup}$ по букве b из состояния $(q_2{}^1,q_2{}^2)$ переходит в состояние $(q_2{}^1,q_0{}^2)$. В диаграмму добавляются вершина $(q_2{}^1,q_0{}^2)$ и ведущая в нее дуга. Рассматривая далее вершины $(q_1{}^1,q_1{}^2)$ и $(q_2{}^1,q_0{}^2)$, мы обнаружим, что новых вершин не возникает, в диаграмму добавляются лишь дуги, ведущие в уже существующие вершины. «Хорошими» будут являться состояния $(q_2{}^1,q_0{}^2)$ и $(q_2{}^1,q_2{}^2)$.

Точно так же строится диаграмма конечного автомата $K^{\cap}$, распознающего язык $L_1 \cap L_2$ (рис. 2.21). Автомат $K^{\cap}$ отличается от автомата $K^{\cup}$ только множеством «хороших» состояний.

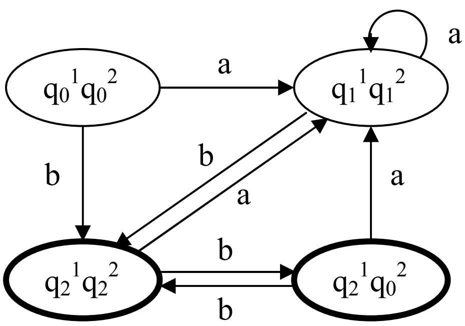


Рис. 2.20

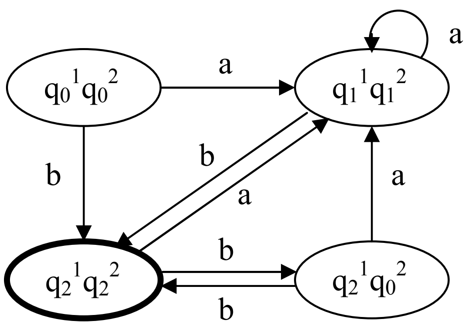


Рис. 2.21

Сейчас приведем пример языка, не являющегося регулярным. Пусть x^f, здесь x – произвольная буква рассматриваемого алфавита, обозначает слово, состоящее из буквы x, повторенной f раз, $f \in \{1, 2, 3,...\}$. Запись $x^f y^h$, где x и y – произвольные буквы алфавита, $f \in \{1, 2, 3,...\}$ и $h \in \{1, 2, 3,...\}$, обозначает результат приписывания справа к слову x^f слова y^h. Через $L^{a\text{-}b}$ обозначим бесконечный язык, каждое слово которого имеет вид $a^n b^n$, т.е. в слове, принадлежащем языку, сначала n раз повторяется буква a, затем такое же число раз повторяется буква b, где n=1, 2, 3,....

Теорема 2.3. Язык $L^{a\text{-}b}$ нерегулярен.

Доказательство проводим методом от противного. Пусть данный язык регулярен. Тогда существует распознающий $L^{a\text{-}b}$ конечный автомат $K^{a\text{-}b}$, число состояний этого автомата обозначим w. Далее через q^z условно обозначим состояние, в котором оказывается автомат $K^{a\text{-}b}$ после того, как он, начиная от своего начального состояния q_0, обработал z букв a подряд, z=1, 2, 3,.... Учитывая, что общее число состояний $K^{a\text{-}b}$ равно w, делаем вывод, что среди состояний, обозначаемых $q^1, q^2, \ldots, q^{w+1}$, имеется по меньшей мере одно с двумя обозначениями. Пусть q^i и q^j, здесь $i \neq j$, – два обозначения некоторого состояния q_k. Так как слова $a^i b^i$ и $a^j b^j$ принадлежат языку $L^{a\text{-}b}$, то как слово b^i, так и слово b^j переводит автомат $K^{a\text{-}b}$ из состояния q_k, в которое он попадает в результате обработки из начального состояния слова a^i или слова a^j, в состояние из множества F. Но тогда процесс работы данного автомата над не принадлежащим языку $L^{a\text{-}b}$ словом $\gamma = a^i b^j$ протекает следующим образом: обработав, начиная от состояния q_0 начальную часть a^i слова γ, автомат оказывается в состоянии q_k; далее, обработав, начиная от состояния q_k заключительную часть b^j входного слова γ, автомат оказывается в «хорошем» состоянии. Таким образом, слово γ принадлежит языку, распознаваемому автоматом $K^{a\text{-}b}$. Полученное противоречие означает справедливость сформулированной теоремы.

Как отмечалось выше, требуемая информация об обработанной части входного слова «помнится» состояниями автомата. Так как число состояний конечно, то память конечного автомата является ограниченной, не достаточной для запоминания сколь угодно большого числа букв a, прошедших перед появлением на входе первой буквы b.

Через $L^{\beta}_{\alpha,\gamma}$, где α, β, γ – фиксированные слова алфавита $A=\{a, b, c\}$, обозначим язык, состоящий из слов вида $\alpha\beta^i\gamma$, здесь β^i обозначает повторенное i раз слово β, i=0, 1, 2, ... (слово β^0 считаем по определению пустым). Рассмотрим в качестве примера язык $L^{abc}_{bca,cab}$ (здесь $\alpha=bca$, $\beta=abc$, $\gamma=cab$), данный язык регулярен, диаграмма соответствующего конечного автомата представлена на рис. 2.22. В результате обработки начальной части bca слова $\alpha\beta^i\gamma$ автомат из начального состояния q_0 переходит в состояние q_3; далее, обрабатывая подслово β, автомат реализует цикл q_3, q_4, q_5, q_3 (этот цикл выполняется i раз); заключительная часть γ обеспечивает переход автомата из состояния q_3 в «хорошее» состояние q_8. Отметим, что, следуя указанному принципу при построении диаграммы конечного автомата для языка $L^{abc}_{bca,acb}$, мы встретимся с осложнением: из состояния q_3 по букве a следует идти в q_4, если этой буквой начинается очередное подслово β, и в q_6, если этой буквой начинается заключительное подслово γ. Путь преодоления возникающей неоднозначности указан далее, в п.3.

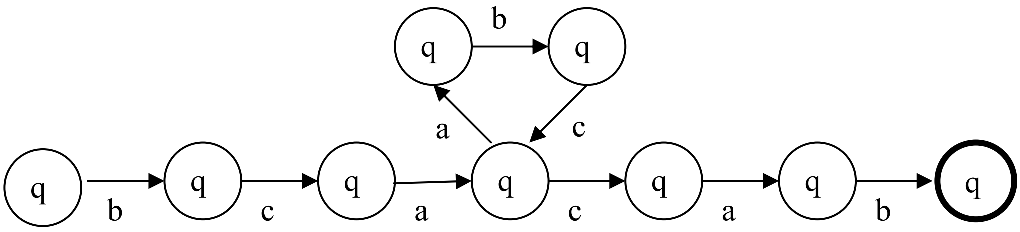


Рис. 2.22.

Примеры.

В нижеперечисленных примерах 2.1-2.19 требуется построить конечный автомат, распознающий язык L. В примерах 2.1-2.8 алфавит $A=\{a,b,c\}$. В примерах 2.9-2.19 алфавит $A=\{0,1,2,\ldots 9\}$

Пример 2.1. $\alpha\in L$ тогда и только тогда, когда в слове α встречается не более трех букв a подряд.

Пример 2.2. $\alpha\in L$ тогда и только тогда, когда в слове α сочетание ab встречается не более двух раз.

Пример 2.3. $\alpha\in L$ тогда и только тогда, когда в слове α содержится подслово $bbcc$.

Пример 2.4. $\alpha\in L$ тогда и только тогда, когда слово α имеет длину не более 8 и содержащит одинаковое число букв a и b.

Пример 2.5. $\alpha \in L$ тогда и только тогда, когда слово α содержит четное число букв.

Пример 2.6. $\alpha \in L$ тогда и только тогда, когда слово α содержит нечетное число букв a.

Пример 2.7. $\alpha \in L$ тогда и только тогда, когда при наличии в слове α буквы a там встречается также и буква b.

Пример 2.8. $\alpha \in L$ тогда и только тогда, когда каждая буква алфавита встречается в слове α не более двух раз.

Пример 2.9. $L=L^{(4)}$.

Пример 2.10. $L=L^{(6)}$.

Пример 2.11. $L=L^{(7)}$.

Пример 2.12. $L=L^{(8)}$.

Пример 2.13. $L=L^{(10)}$.

Пример 2.14. $L=L^{(15)}$.

Пример 2.15. $L=L^{(20)}$.

Пример 2.16. $L=L^{(25)}$.

Пример 2.17. $L=L^{(30)}$.

Пример 2.18. $L=L^{(50)}$.

Пример 2.19. $L=L^{(125)}$.

Пример 2.20.

Построить конечные автоматы, распознающие объединение, пересечение и разность языков, заданных конечными автоматами, диаграммы которых представлены на рис. 2.22.

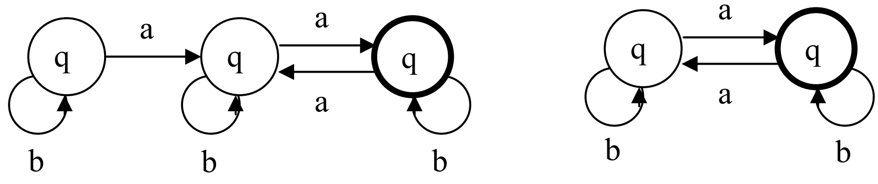


Рис. 2.22.

Пример 2.21.

Построить конечные автоматы, распознающие объединение, пересечение и разность языков, заданных конечными автоматами, диаграммы которых представлены на рис. 2.23.

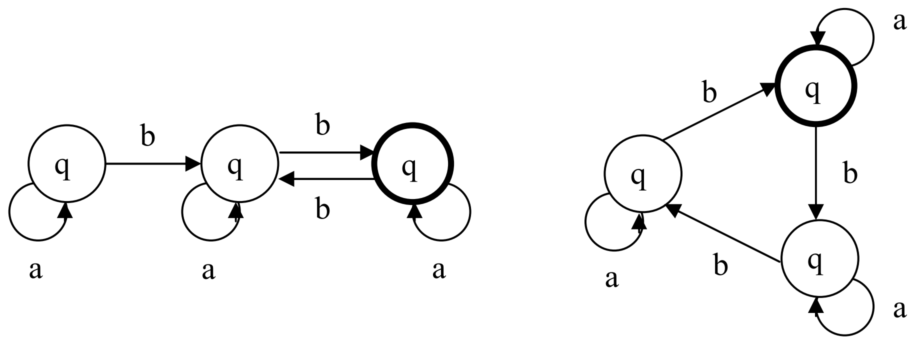


Рис. 2.23.

Пример 2.22.

Построить конечные автоматы, распознающие объединение, пересечение и разность языков, заданных конечными автоматами, диаграммы которых представлены на рис. 2.24.

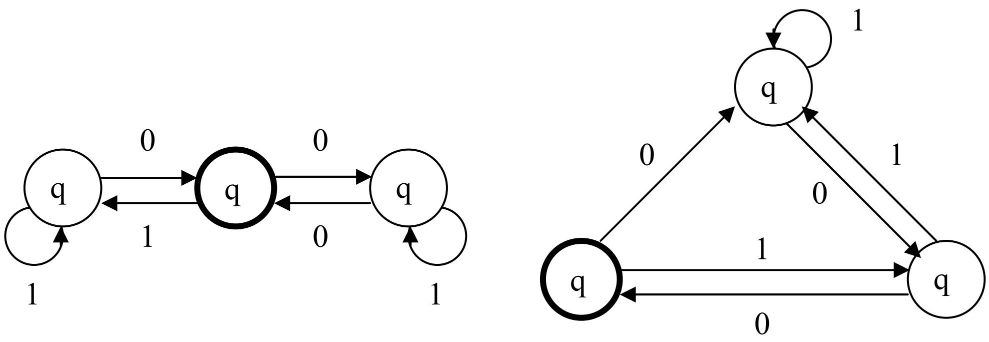


Рис. 2.24.

Пример 2.23.

Построить конечные автоматы, распознающие объединение, пересечение и разность языков, заданных конечными автоматами, диаграммы которых представлены на рис. 2.25.

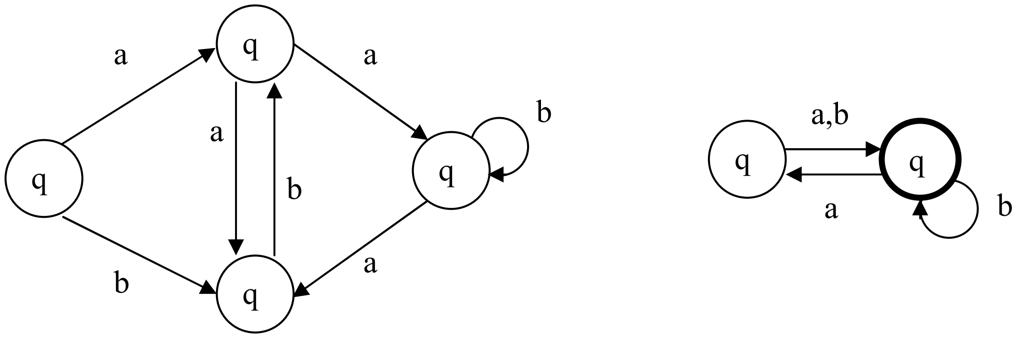


Рис. 2.25.

3. Недетерминированные конечные автоматы и определяемые ими языки.

Недетерминированный конечный автомат определяем как совокупность $K^*=\{Q, A, q_0, g^*, F\}$, где Q, A, q_0 и F имеют тот же смысл, что в п.2, а функция переходов g^* является отображением типа $Q\text{x}A\to[2^Q\backslash\varnothing]$. Напомним, что для любого множества M через 2^M обозначается множество всех подмножеств из M; символ $\varnothing$ обозначает пустое множество. Совокупность $g^*(q_i,a_j)$ – это всегда непустое подмножество состояний автомата, в любое из которых он может перейти из состояния q_i под воздействием буквы a_j, здесь $q_i\in Q$, $a_j\in A$.

В отличие от недетерминированных автоматов, конечные автоматы, введенные в предыдущем пункте, именуем ***детерминированными***. Детерминированный конечный автомат является частным случаем недетерминированного, он получается из последнего в предположении, что все множества $g^*(q_i,a_j)$ являются одноэлементными.

Будучи запущен в работу над произвольным словом α из своего начального состояния, недетерминированный конечный автомат может функционировать по-разному. Язык $L(K^*)$, распознаваемый недетерминированным конечным автоматом K^*, определяем следующим образом: слово $\alpha = a_{i_1}a_{i_2}\ldots a_{i_p}$ принадлежит языку $L(K^*)$ тогда и только тогда, когда имеется последовательность состояний автомата $q_{r_0}, q_{r_1}, q_{r_2}, \ldots q_{r_p}$ такая, что

$$q_{r_0} = q_0;$$

$$q_{r_1} \in g^*(q_{r_0}, a_{i_1});$$

$$q_{r_2} \in g^*(q_{r_1}, a_{i_2});$$

. . .

$$q_{r_p} \in g^*(q_{r_{p-1}}, a_{i_p});$$

при этом $q_{r_p} \in F$.

Иными словами, слово α принадлежит языку $L(K^*)$ тогда и только тогда, когда существует способ работы данного автомата над данным словом такой, что после завершения обработки α автомат оказывается в состоянии, принадлежащем множеству F.

На рис. 3.1 дан пример недетерминированного конечного автомата K_1* (причина недетерминированности заключается в том, что под воздействием буквы a автомат из состояния q_0 может либо перейти в состояние q_1, либо остаться в состоянии q_0). Легко видеть, что язык $L(K_1^*)$ совпадает с ранее введенным регулярным языком L_4.

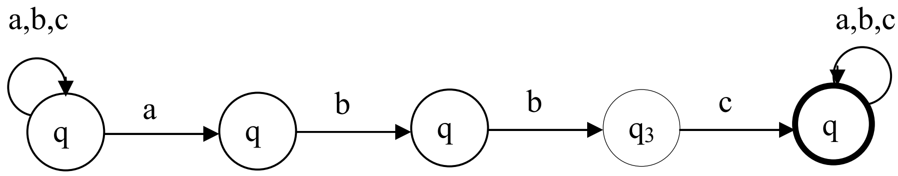


Рис. 3.1.

Теорема 3.1. Языки, определяемые недетерминированными конечными автоматами, являются регулярными языками.

По произвольному недетерминированному конечному автомату $K^*=\{Q, A, q_0, g^*, F\}$, распознающему язык $L(K^*)$, детерминированный автомат $K'=\{Q', A, q'_0, g', F'\}$ такой, что $L(K')=L(K^*)$, строим следующим образом. Полагаем $Q'=2^Q\backslash\varnothing$, т.е. состояниями автомата K' считаем непустые подмножества состояний автомата K^*, при этом определяем $q'_0=\{q_0\}$. Функцию переходов автомата K' строим таким образом, чтобы, обработав из q'_0 произвольное слово α, этот автомат приходил в состояние, представляющее собой подмножество состояний исходного автомата K^*, в каждое из которых K^* может перейти из своего начального состояния в результате обработки некоторым способом данного слова α. Достижение указанной цели обеспечивается следующим определением функции g':

$q_u \in g'(q'_i, a_j) \Leftrightarrow (\exists\, q_v \in q'_i)$ такое, что $\{q_u \in g^*(q_v, a_j)\}$.

Так как слово α принадлежит $L(K^*)$ тогда и только тогда, когда существует способ работы автомата K^* над данным словом такой, что после завершения обработки α автомат оказывается в одном из состояний множества F, совокупность F' определяем следующим образом: произвольное состояние $q'_i \in F'$ тогда и только тогда, когда $q'_i \cap F \neq \varnothing$.

Построенный изложенным способом детерминированный конечный автомат распознает тот же язык, что и исходный автомат K^*. Теорема доказана.

Далее в данном пособии недетерминированность рассматриваемых автоматов будет оговариваться дополнительно, под термином «конечный автомат» всегда понимается детерминированный конечный автомат.

Два автомата называем эквивалентными, если они распознают один и тот же язык. Согласно изложенному в доказательстве теоремы 3.1 алгоритму, для недетерминированного конечного автомата с N состояниями всегда можно построить эквивалентный ему детерминированный конечный автомат, имеющий 2^N-1 состояние. В действительности число требуемых состояний может оказаться меньшим. На рис. 3.2 представлен недетерминированный автомат K^*, имеющий 3 состояния. Диаграмму переходов эквивалентного ему детерминированный конечного автомата K' (см. рис. 3.3) строим следующим образом. Вводим вершину - состояние $\{q_0\}$. Из своего начального состояния q_0 автомат K^* по букве a либо переходит в q_1, либо остается в q_0; по букве b автомат переходит в q_2. Поэтому автомат K' по букве a из $\{q_0\}$ переходит в состояние $\{q_0,q_1\}$, а по букве b переходит в состояние $\{q_2\}$. Из состояний совокупности $\{q_0,q_1\}$ по букве a автомат K^* переходит в состояния той же совокупности, а по букве b как из состояния q_0, так и из состояния q_1 реализуется переход в q_2. Поэтому автомат K' по букве a из $\{q_0,q_1\}$ переходит в то же состояние $\{q_0,q_1\}$, а по букве b переходит в состояние $\{q_2\}$. Из состояния q_2 автомат K^* по букве a переходит в q_0, а по букве b – остается в q_2. Поэтому автомат K' по букве a из $\{q_2\}$ переходит в $\{q_0\}$, а по букве b – остается в $\{q_2\}$. Построение автомата K' закончено. Другие мыслимые состояния-подмножества оказываются излишними, они недостижимы; начав работу из своего начального состояния, автомат может оказываться только в трех введенных состояниях (включая начальное).

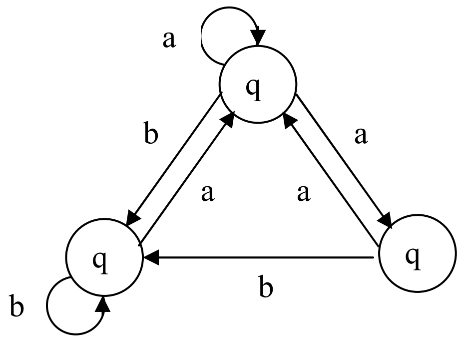


Рис. 3.2.

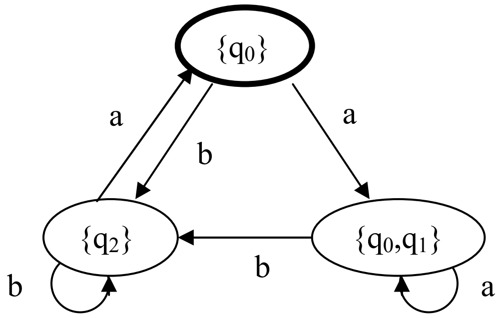


Рис. 3.3.

Концепция недетерминированного конечного автомата легко применяется для построения автомата, распознающего объединение двух регулярных языков. Пусть L_1 и L_2 – регулярные языки, распознаваемые конечными автоматами $K^1=\{Q^1,\ A,\ q^1{}_0,\ g^1,\ F^1\}$ и $K^2=\{Q^2,\ A,\ q^2{}_0,\ g^2,\ F^2\}$ соответственно. Пусть D_1 и D_2 – диаграммы переходов, определяющие эти конечные автоматы. Для построения диаграммы переходов D автомата, распознающего объединение языков L_1 и L_2, объединяем эти диаграммы следующим образом. Вводим новую вершину - состояние q_0. По каждой букве x входного алфавита A из q_0 проводим две дуги с надписанной буквой x; верхняя дуга имеет своим концом вершину $g^1(q^1{}_0,x)$, т.е. состояние, в которое переходит из своего начального состояния под воздействием буквы x первый автомат, а нижняя – вершину $g^2(q^2{}_0,x)$, т.е. состояние, в которое переходит из своего начального состояния под воздействием буквы x второй автомат. Начальным состоянием построенного автомата считаем q_0; множество F его «хороших» состояний определяем как объединение множеств F^1 и F^2. Специально отметим, что в случае, когда хотя бы в одном из автоматов K^1, K^2 начальное состояние является «хорошим», в F следует включить состояние q_0. На первом такте обработки любого непустого входного слова $\alpha=x\beta$ автомат имеет возможность перехода из q_0 либо по верхней, либо по нижней дуге с надписанной буквой x. Если реализован переход по верхней дуге, то далее фактически работает автомат K^1 и проверяется принадлежность слова α языку L_1; если реализован переход по нижней дуге, далее работает автомат K^2 и проверяется принадлежность слова α языку L_2; построенный автомат в результате обработки произвольного входного слова α может оказаться в состоянии, принадлежащем подмножеству F, тогда и только тогда, когда α принадлежит объединению языков L_1 и L_2. На рис. 3.4 представлена диаграмма переходов построенного по изложенной схеме недетерминированного конечного автомата, распознающего множество чисел, каждое из которых кратно 2 или 5.

В заключение отметим, что для любых слов α, β и γ в произвольном алфавите A определяемый ими язык $L^{\beta}_{\alpha,\gamma}$ (см. п.2) является регулярным. Диаграмма (вообще говоря, недетерминированного) конечного автомата, определяющего этот язык, строится аналогично представленной на рис. 2.13 диаграмме автомата, определяющего язык $L^{abc}_{bca,cab}$.

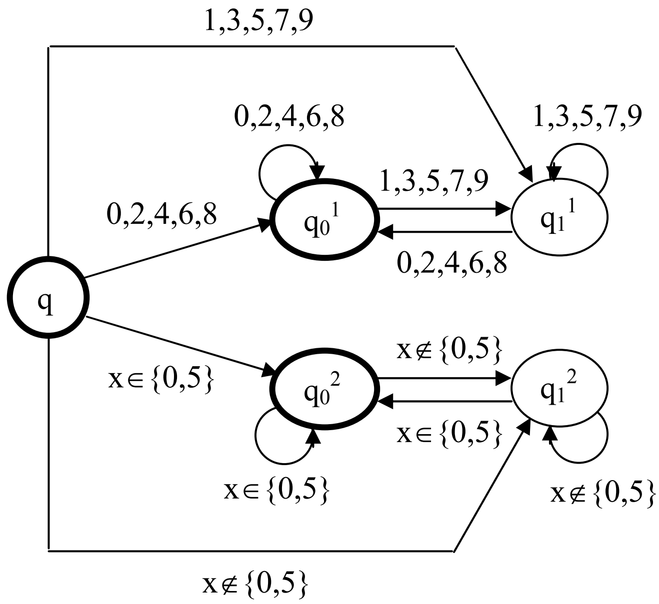


Рис. 3.4

Примеры.

Пример 3.1.

По недетерминированному конечному автомату, диаграмма которого представлена на рис. 3.5, построить детерминированный конечный автомат, распознающий тот же язык.

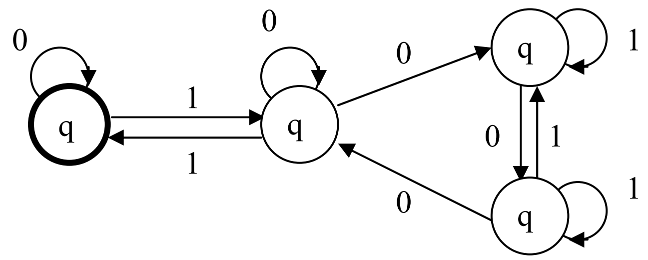


Рис. 3.5

Пример 3.2.

По недетерминированному конечному автомату, диаграмма которого представлена на рис. 3.6, построить детерминированный конечный автомат, распознающий тот же язык.

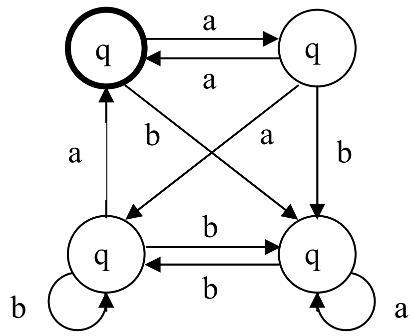


Рис. 3.6

Пример 3.3.

По недетерминированному конечному автомату, диаграмма которого представлена на рис. 3.7, построить детерминированный конечный автомат, распознающий тот же язык.

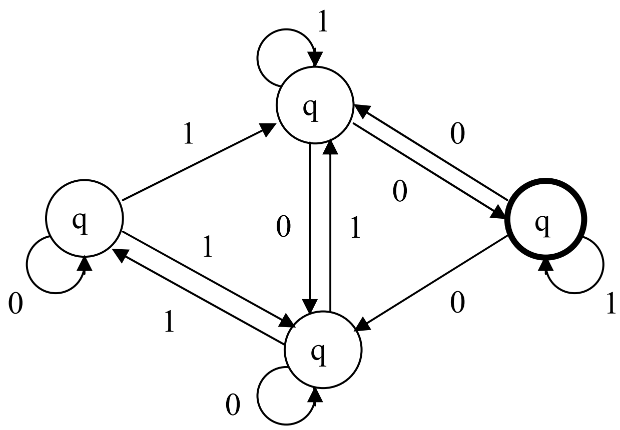


Рис. 3.7

Пример 3.4.

Построить конечный автомат, распознающий дополнение к языку, заданному недетерминированным конечным автоматом, диаграмма которого представлена на рис. 3.8

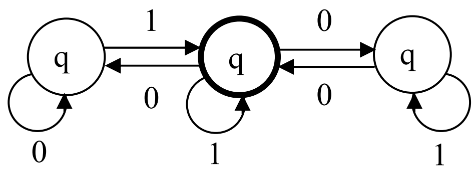
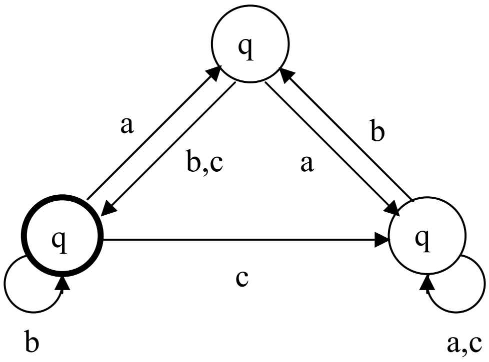


Рис. 3.8

Пример 3.5.

Построить недетерминированный конечный автомат, распознающий объединение языков, заданных конечными автоматами, диаграммы которых представлены на рис. 3.9



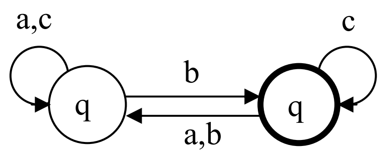


Рис. 3.9

Пример 3.4.

Построить недетерминированный конечный автомат, распознающий множество чисел, делящихся на 25.

4. Замкнутость класса регулярных языков относительно операций конкатенации, возведения в степень и итерации.

Пусть L_1 и L_2 – регулярные языки, распознаваемые детерминированными конечными автоматами $K^1=\{Q^1, A, q^1_0, g^1, F^1\}$ и $K^2=\{Q^2, A, q^2_0, g^2, F^2\}$ соответственно. Дадим алгоритм синтеза по имеющимся диаграммам переходов конечных автоматов K_1 и K_2 диаграммы переходов недетерминированного конечного автомата K, распознающего язык L_1L_2 ,т.е. конкатенацию (сцепку) языков L_1 и L_2.

Сначала предположим, что $q^1_0 \notin F^1$; выполнение данного условия будем именовать общим случаем. Через M обозначим совокупность тех дуг (вместе с надписанными над ними буквами) в диаграмме переходов автомата K^1, которые своими концами имеют вершины подмножества F^1. Специально отметим, что (в случае их присутствия в диаграмме автомата K^1) в M входят и петли, т.е. дуги вида (q^1_j, q^1_j), где $q^1_j \in F^1$. Для каждой принадлежащей множеству M дуги (q^1_i,q^1_j) с надписанной над ней буквой x входного алфавита строим дугу-дубликат (q^1_i,q^2_0) с той же надписанной буквой. Так реализуется «сцепка» диаграмм переходов конечных автоматов K_1 и K_2, результатом которой является диаграмма переходов конструируемого недетерминированного конечного автомата K; начальным состоянием автомата K объявляется состояние q^1_0, совокупность F «хороших» состояний автомата K считается совпадающей с F^2. Согласно выполненному построению, до перехода по некоторой дуге-дубликату в состояние q^2_0, конечный автомат K функционирует как автомат K_1, после перехода в q^2_0 и до завершения работы – как K_2.

Сконструированный конечный автомат проверяет, можно ли входное слово α, $\alpha \in A^*$, разбить на две последовательные части α_1 и α_2 так, что $\alpha_1 \in L_1$ и $\alpha_2 \in L_2$ (указанное разбиение оказывается возможным тогда и только тогда, когда существует способ работы данного автомата такой, что, обработав тестируемое слово α, автомат оказывается в состоянии из F).

Действительно, конечный автомат K, начиная работу из состояния q_0^1, начальную часть произвольного входного слова α обрабатывает в режиме автомата K_1. Пусть на некотором такте работы автомата K в данном режиме поступила буква x. Предположение о том, что она является последней в принадлежащей L_1 части α_1 входного слова α, возникает тогда, когда под воздействием x автомат K_1 может перейти в свое «хорошее» состояние. На данном такте для недетерминированного автомата K по его построению имеем две возможности:

1) выполнить переход в соответствии с диаграммой автомата K_1 (при этом считается, что данной буквой x начальная часть α_1, $\alpha_1 \in L_1$, входного слова не заканчивается);
2) под воздействием x по соответствующей дуге-дубликату перейти в состояние $q^2{}_0$ (здесь считается, что буквой x завершается первая часть α_1, $\alpha_1 \in L_1$, входного слова α). Далее автомат K будет функционировать как автомат K_2; его работа должна завершиться в состоянии, принадлежащем подмножеству F^2, если оставшаяся часть α_2 тестируемого слова α принадлежит языку L_2.

За счет своей недетерминированности, автомат K может по-разному разбивать входное слово α на части α_1, $\alpha_1 \in L_1$, и α_2. Слово α принадлежит языку, распознаваемому автоматом K тогда и только тогда, когда его можно представить в виде $\alpha = \alpha_1 \alpha_2$, где $\alpha_1 \in L_1$ и $\alpha_2 \in L_2$; отметим, что из условия $q^1{}_0 \notin F^1$ следует $\alpha_1 \neq \lambda$.

Первым специальным случаем назовем следующую ситуацию: $q^1{}_0 \in F^1$ и $q^2{}_0 \notin F^2$. Специфика ситуации заключается в том, что в язык L_1 входит и пустое слово. В силу возможного равенства $\alpha_1 = \lambda$, следует конструировать автомат K таким образом, чтобы для любой буквы x входного алфавита A имелась дополнительная возможность перехода автомата из начального состояния $q^1{}_0$ в состояние, в которое по данной букве из своего начального состояния переходит автомат K_2.

Для первого специального случая «сцепка» диаграмм переходов конечных автоматов K_1 и K_2, результатом которой является диаграмма переходов конструируемого недетерминированного конечного автомата K, реализуется в два этапа:

1) идентично тому, как это делается в общем случае, проводятся все дуги-дубликаты;
2) для каждой буквы x входного алфавита проводится дополнительная дуга $(q^1{}_0, g^2(q^2{}_0, x))$ с надписанной над ней буквой x. Как и ранее, начальным

состоянием автомата K объявляется состояние $q^1{}_0$, совокупность F «хороших» состояний автомата K считается совпадающей с F^2.

Вторым специальным случаем назовем следующую ситуацию: $q^1{}_0 \in F^1$ и $q^2{}_0 \in F^2$. Специфика ситуации заключается в том, что как в язык L_1, так и в язык L_2 входит пустое слово; следовательно, пустое слово принадлежит и конкатенации языков L_1L_2. Отличие алгоритма построения недетерминированного автомата K в данной ситуации от первого специального случая только в назначении множества «хороших» его состояний. В данном случае считается $F=F^2 \cup \{q^1{}_0\}$.

Отметим, что процедура синтеза автомата, распознающего конкатенацию языков, определяемых недетерминированными конечными автоматами $K^*{}_1$ и $K^*{}_2$, реализуется идентичным изложенному образом.

Рассмотрим ряд примеров.

Диаграммами переходов, представленными на рис. 4.1, заданы конечные автоматы, распознающие языки L_1 и L_2 (отметим, что автомат, распознающий язык L_1, недетерминированный). На рис. 4.2 представлена диаграмма переходов недетерминированного конечного автомата, распознающего язык L_1L_2.

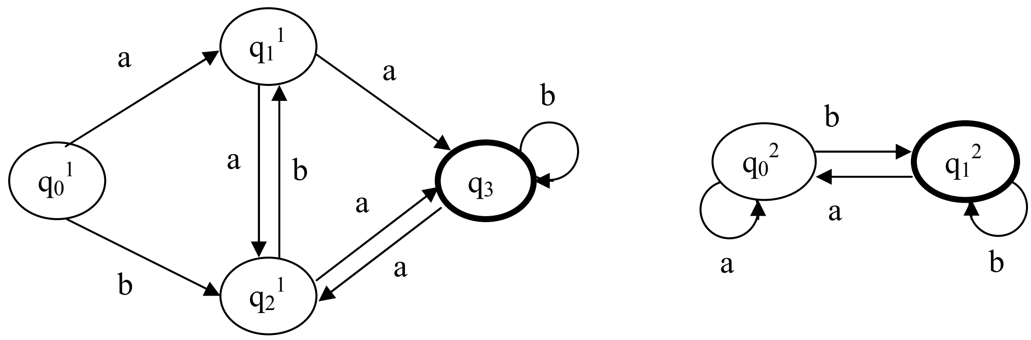


Рис. 4.1.

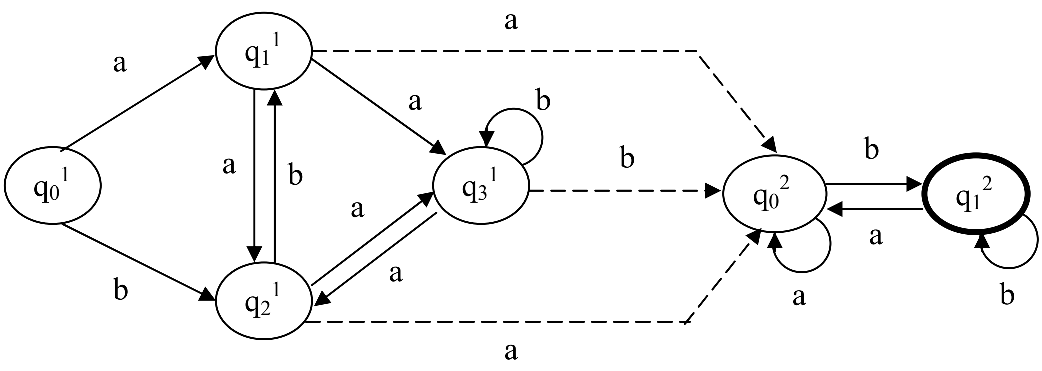


Рис. 4.2.

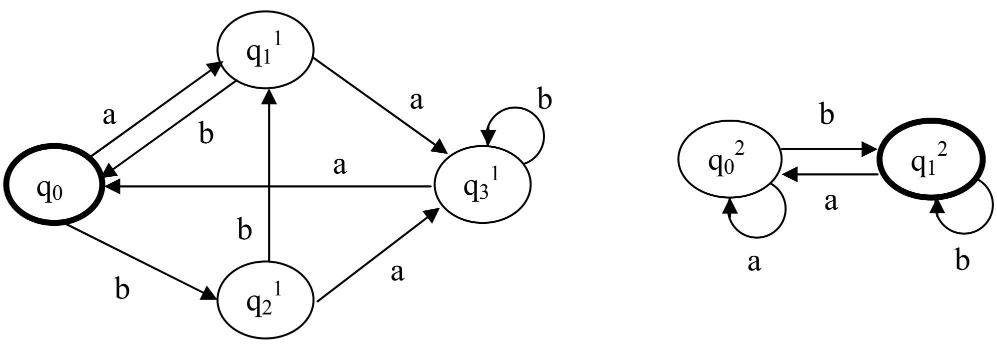


Рис. 4.3.

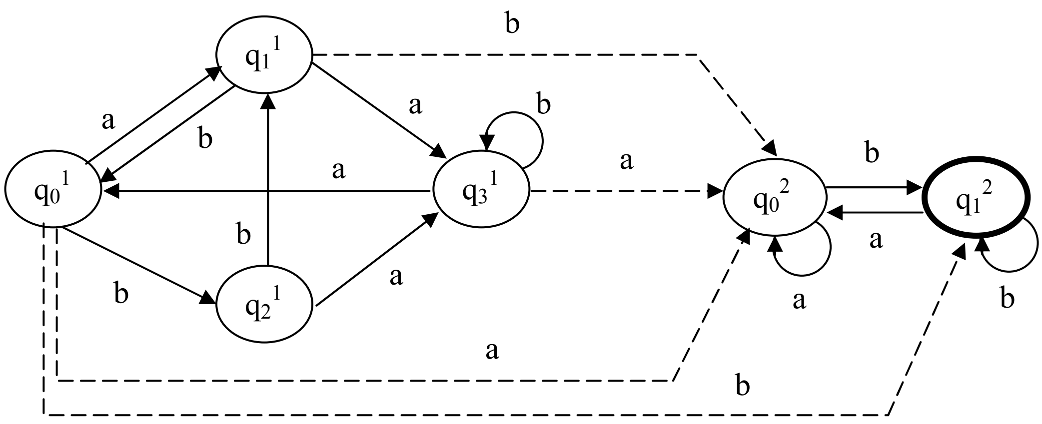


Рис. 4.4.

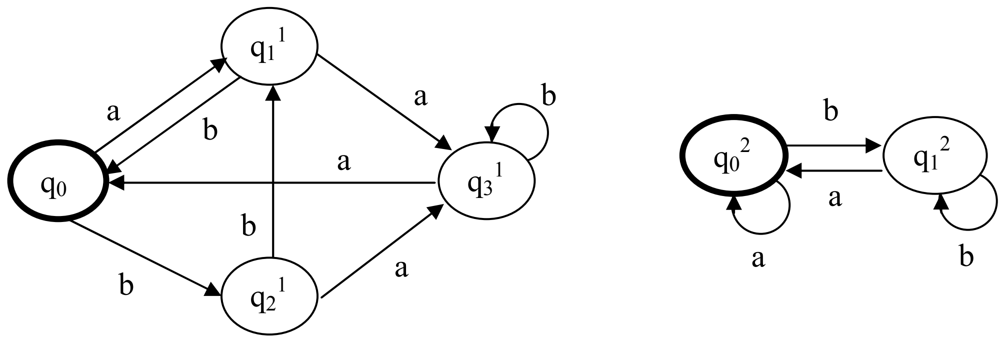


Рис. 4.5.

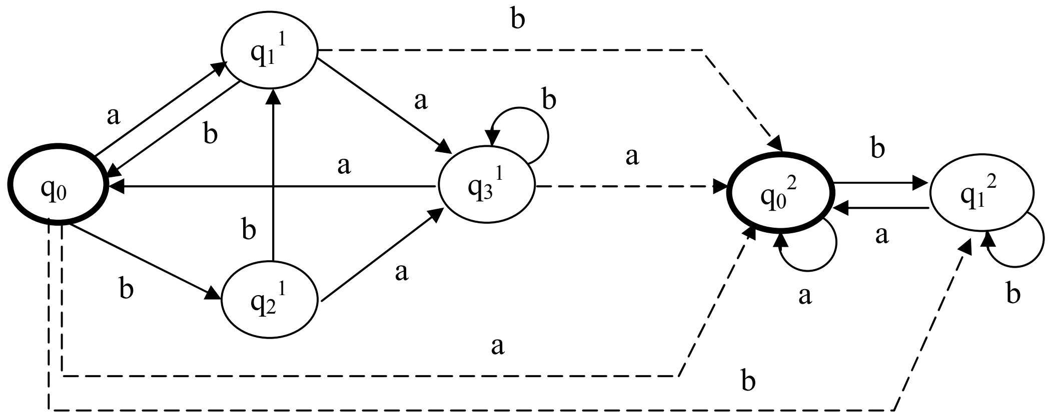


Рис. 4.6.

После того, как установлена замкнутость класса регулярных языков относительно операции конкатенации, легко доказать замкнутость этого класса и относительно операции возведения в любую целую неотрицательную степень. Пусть L – произвольный регулярный язык. Так как по определению степени языка $L^1=L$, получаем, что L^1 – регулярный язык. Предположим, что язык L^k регулярен, здесь k – произвольная натуральная константа. Так как язык $L^{k+1}=L^kL$ является результатом конкатенации регулярных языков L^k и L, то L^{k+1} – также регулярный язык. Таким образом, регулярность языка L^n при любом n=1, 2, 3, ... следует из принципа математической индукции. Язык L^0 также регулярен, ибо он конечен, имеет в своем составе только пустое слово λ.

Рассмотрение проблемы синтеза по распознающему язык L конечному автомату K конечного автомата K^* (вообще говоря, недетерминированного), распознающего язык L^*, начнем с частного случая, когда в диаграмме автомата K нет дуг, входящих в вершину q_0; состояние q_0 в такой ситуации является невозвратным – выйдя из q_0 на первом такте работы над любым словом α, автомат K в него никогда уже не вернется. Алгоритм построения по имеющейся диаграмме переходов конечного автомата K диаграммы переходов недетерминированного конечного автомата K^*, распознающего язык L^*, в данном частном случае (назовем его алгоритмом A) достаточно прост. Через M обозначим совокупность тех дуг (вместе с надписанными над ними буквами) в диаграмме переходов автомата K, которые своими концами имеют вершины из подмножества F данного автомата. Специально отметим, что (в случае их присутствия в диаграмме автомата K) в M входят и петли, т.е. дуги вида (q_j, q_j), где $q_j \in F$. Реализуя алгоритм A, для каждой принадлежащей множеству M 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 конструируемого автомата считаем q_0, это же состояние считаем единственным «хорошим» состоянием в K^*. Построение диаграммы переходов автомата K^* закончено.

Автомат K^* проверяет, принадлежит ли произвольное входное слово α языку L^*, т.е. можно ли разбить данное слово на некоторое число p последовательных частей $\alpha_1, \alpha_2, \ldots, \alpha_p$ так, что каждое подслово α_i, $i = \overline{1, p}$, принадлежит L (указанное разбиение оказывается возможным тогда и только тогда, когда существует способ работы данного автомата такой, что обработав тестируемое слово α, автомат оказывается в «хорошем» состоянии).

Действительно, начальную часть произвольного непустого входного слова α конечный автомат K^*, начиная работу из состояния q_0, обрабатывает в режиме автомата K. Предположение о том, что поступающая на некотором такте произвольная буква x является последней в принадлежащей L первой части входного слова, возникает тогда, когда под воздействием x автомат K может перейти в свое «хорошее» состояние. На данном такте для недетерминированного автомата K^* по его построению имеем две возможности:

1) выполнить переход в соответствии с диаграммой автомата K (при этом считается, что данной буквой x часть α_1, $\alpha_1 \in L$, входного слова не заканчивается);
2) под воздействием x по соответствующей дуге-дубликату перейти в состояние q_0 (здесь считается, что буквой x завершается часть α_1, $\alpha_1 \in L$, входного слова α). Если данная буква x оказалась в слове α последней, автомат работу заканчивает, его заключительное состояние оказывается «хорошим», слово α языку L^* принадлежит.

Если же данная буква x – не последняя в слове α, то, перейдя по этой букве в состояние q_0, автомат K^* далее будет вновь функционировать как автомат K; предположение о том, что поступающая в некоторый момент произвольная буква x является последней в принадлежащей L второй части входного слова возникает тогда, когда под воздействием x автомат K может перейти в свое «хорошее» состояние. На данном такте для автомата K^* вновь имеются две вышеописанные возможности. Реализация второй возможности означает что данной буквой x заканчивается вторая часть α_2, $\alpha_2 \in L$, входного слова; если данная буква x оказалась в слове α последней, автомат работу заканчивает, его заключительное состояние оказывается «хорошим», слово α языку L^* принадлежит; если же данная буква x – не последняя в слове α, то далее, на каждом очередном этапе, начиная от состояния q_0, автомат K^* будет вновь функционировать как автомат K, выделяя очередную принадлежащую L часть входного слова.

Представление входного слова α в виде $\alpha = \alpha_1 \alpha_2 ... \alpha_p$, где каждое подслово α_i, $i = \overline{1, p}$, принадлежит L, оказывается возможным тогда и только тогда, когда существует способ работы автомата K^* такой, что, обработав тестируемое слово α, он оказывается в «хорошем» состоянии q_0.

Заметим, что, если множество «хороших» состояний автомата K^* изменить, считать его совпадающим с множеством F исходного автомата K, то получаемый автомат распознает язык L^+.

Применим алгоритм A к диаграмме конечного автомата K, представленной на рис. 4.7. В результате получим диаграмму конечного автомата K^* (рис 4.8). Автомат K функционирует в алфавите $\{a,b,c\}$и распознает язык L, состоящий из слов, которые начинаются на букву c и в каждом из которых буква c встречается ровно два раза. Автомат K^* распознает итерацию языка L. Непустое слово α принимается автоматом K^* тогда и только тогда, когда буква c встречается в нем любое четное, отличное от нуля число раз.

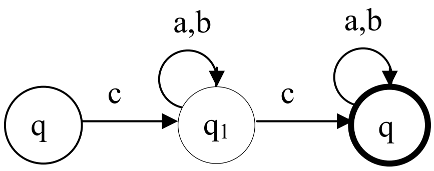


Рис. 4.7

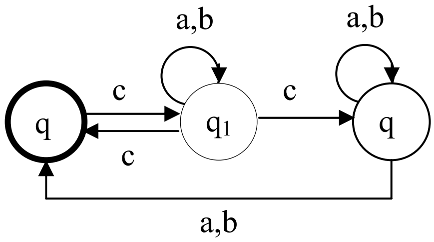


Рис. 4.8

Сейчас рассмотрим определяемый представленной на рис. 4.9 диаграммой конечный автомат K', распознающий язык L'. Слово α в алфавите $\{a,b,c\}$ принадлежит языку L' тогда и только тогда, когда оно имеет вид $a^i bc$, здесь i – произвольное целое неотрицательное число. Если не обратить внимание на тот факт, что начальное состояние автомата K' не является невозвратным, и непосредственно к диаграмме данного автомата применить алгоритм A, то получим в результате автомат, который распознает язык, являющийся расширением языка $(L')^*$. В частности, этому языку будут принадлежать не входящие в $(L')^*$ слова a, aa, aaa и т.д. Для того, чтобы построить автомат, распознающий язык $(L')^*$, перед применением алгоритма A выполним преобразование автомата K', обеспечивающее невозвратность начального состояния. Диаграмму переходов автомата K'' (см. рис. 4.10) по диаграмме автомата K' строим следующим образом: вводим дублирующее для q_0 состояние q_0^1; считаем, что по букве a автомат не остается в состоянии q_0, а переходит в состояние q_0^1; из состояния q_0^1 по букве a реализуется петля, по букве b – переход в состояние q_1. Легко видеть, что автоматы K' и K'' распознают один и тот же язык L', но для построения автомата, распознающего $(L')^*$, к автомату K'' (его начальное состояние является невозвратным) можно применить алгоритм A. В результате получаем автомат $(K'')^*$, диаграмма переходов которого представлена на рис.4.11. (Заметим, что состояние q_2 в автомате $(K'')^*$

становится отрицательно поглощающим, поэтому на диаграмме его можно не изображать.)

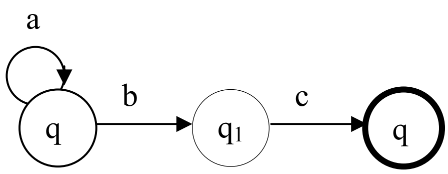


Рис. 4.9

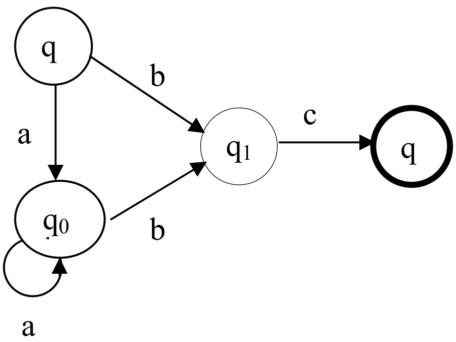


Рис. 4.10

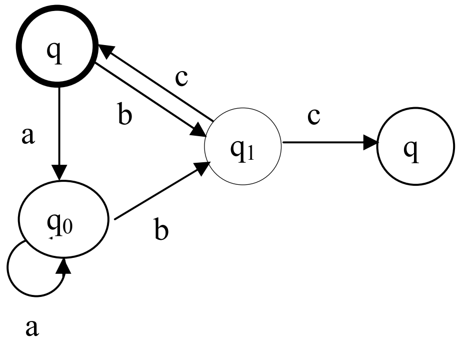


Рис. 4.11

Изложим общий алгоритм A_0 преобразования произвольного конечного автомата K, в котором состояние q_0 не является невозвратным, к распознающему тот же язык автомату K' с невозвратным начальным состоянием. С этой целью: для начального состояния q_0 автомата K 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)); по каждой имеющейся дуге (q_0, q_i) с надписанной над ней буквой, здесь q_i – произвольное состояние автомата K, проводим дугу (q_0^1, q_i) 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 автомата K' считаем совпадающим с множеством «хороших» состояний исходного автомата K; если $q_0 \in F$, то q_0^1 также принадлежит F.

Синтез по заданной диаграмме переходов конечного автомата *К*, распознающего язык *L*, диаграммы переходов конечного автомата *К**, распознающего язык *L**, в общем случае выполняется следующим образом: сначала применяется алгоритм A_0, обеспечивающий невозвратность начального состояния q_0, затем – алгоритм *A*.

Предварительное применение алгоритма A_0 излишне, когда состояние q_0 в исходном автомате *К* – невозвратное. Легко показать, что оно излишне и тогда, когда это состояние в автомате *К* является «хорошим». Действительно, в случае, когда начальное состояние исходного автомата не является невозвратным, автомат, конструируемый применением алгоритма *A*, может определять расширение языка *L** только за счет того, что состояние q_0 объявляется «хорошим» (данное состояние непременно должно стать «хорошим», ибо $\lambda \in L^*$). Но если оно уже в исходном автомате *К* было «хорошим», то для получения требуемого результата достаточно непосредственное применение алгоритма *A*. На рис. 4.12 представлена диаграмма конечного автомата, который определяет язык, состоящий из слов, в которых буква *c* не встречается или встречается два раза. Т.к. начальное состояние является «хорошим», то для построения итерации можно сразу применить алгоритм *A* (рис. 4.13).

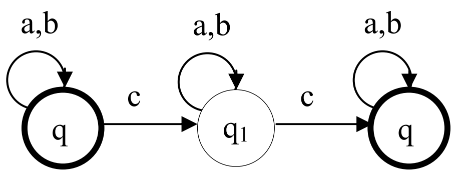


Рис. 4.12

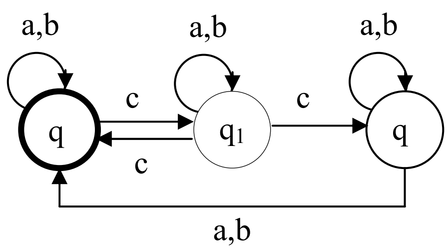


Рис. 4.13

Примеры.

Пример 4.1.

Построить конечный автомат, распознающий конкатенацию языков, заданных конечными автоматами, диаграммы которых представлены на рис. 4.14, в случаях, когда подмножества «хороших» состояний этих автоматов определяются следующим образом:

A. $F_1=\{q_1\}$, $F_2=\{q_3\}$;

B. $F_1=\{q_0\}$, $F_2=\{q_1, q_2\}$;

C. $F_1=\{q_0\}$, $F_2=\{q_0, q_3\}$.

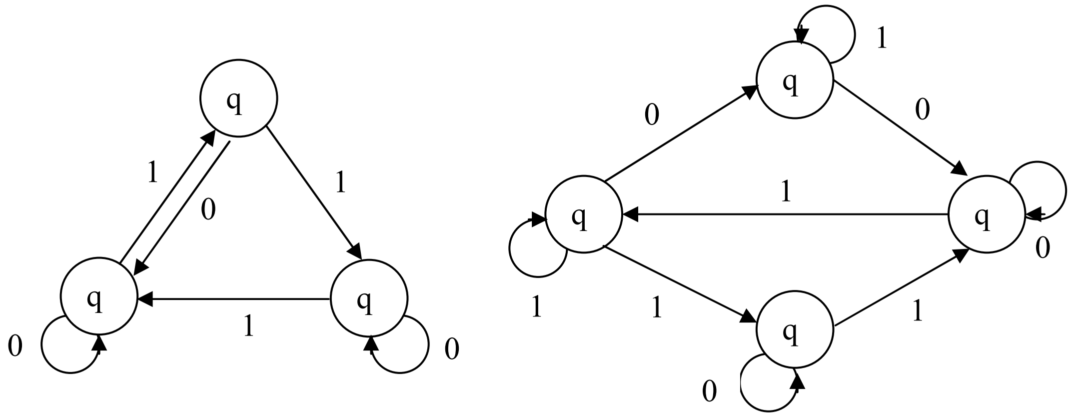


Рис. 4.14

Пример 4.2.

Построить конечный автомат, распознающий конкатенацию языков, заданных конечными автоматами, диаграммы которых представлены на рис. 4.15, в случаях, когда подмножества «хороших» состояний этих автоматов определяются следующим образом:

A. $F_1=\{q_1, q_2\}$, $F_2=\{q_0\}$;

B. $F_1=\{q_0\}$, $F_2=\{q_3\}$;

C. $F_1=\{q_0, q_1\}$, $F_2=\{q_0\}$.

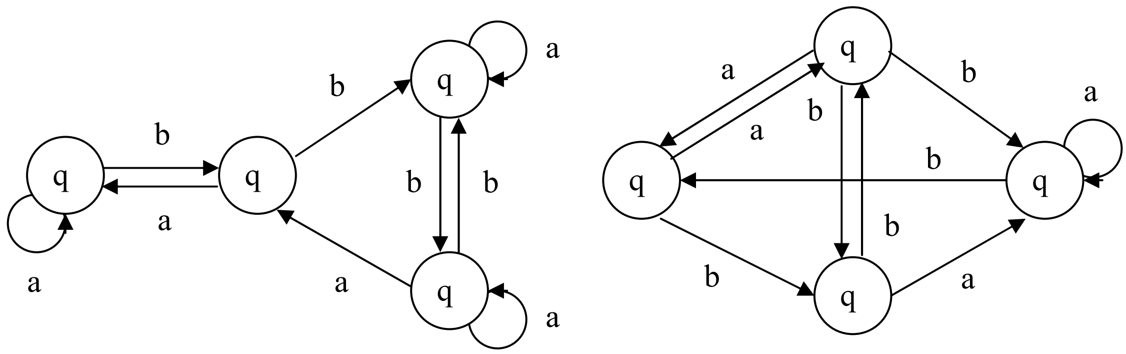


Рис. 4.15

Пример 4.3.

Построить конечный автомат, распознающий итерацию языка, заданного конечным автоматом, диаграмма которого представлена на рис. 4.16, в случаях, когда подмножество «хороших» состояний этого автомата определяется следующим образом:

A. $F=\{q_1, q_2\}$;

B. $F=\{q_0, q_4\}$;

C. $F=\{q_3, q_4\}$.

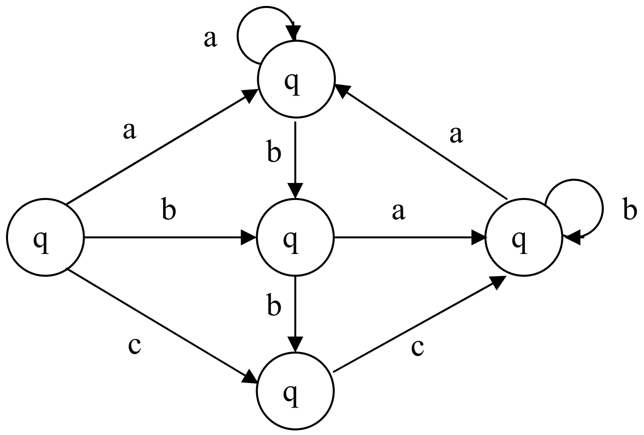


Рис. 4.16

Пример 4.4.

Построить конечный автомат, распознающий итерацию языка, заданного конечным автоматом, диаграмма которого представлена на рис. 4.17, в случаях, когда подмножество «хороших» состояний этого автомата определяется следующим образом:

A. $F=\{q_1, q_2\}$;

B. $F=\{q_0, q_4\}$;

C. $F=\{q_3, q_4\}$.

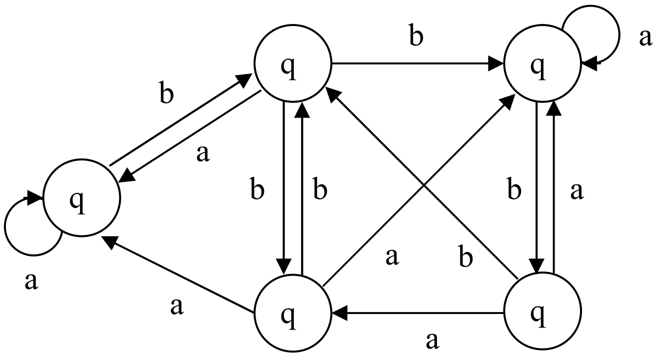


Рис. 4.15