Министерство образования и науки Российской Федерации
Государственное образовательное учреждение высшего профессионального образования
«Нижегородский государственный архитектурно-строительный университет»

М.И.Лиогонький, Т.А.Береговая

Элементы теории конечных автоматов и регулярных языков

Методические указания и расчетно-графическая работа
по курсу «Дискретная математика» для студентов ННГАСУ
специальности «Информационные системы и технологии»

Нижний Новгород
ННГАСУ
2010

УДК 519.95

ББК В18

К–57

Лиогонький М.И., Береговая Т.А. Элементы теории конечных автоматов и регулярных языков. Методические указания и расчетно-графическая работа по курсу «Дискретная математика» для студентов ННГАСУ специальности «Информационные системы и технологии». Нижегород.гос.архит.-строит.ун-т. – Н.Новгород: ННГАСУ, 2010.- 64с.

В методической разработке вводится понятие конечного автомата как простейшей математической модели систем дискретного действия. Изучаются свойства конечных автоматов и языков, ими распознаваемых. Приводятся алгоритмы синтеза и анализа конечных автоматов. Вводится понятие R-языка. Доказывается теорема Клини о совпадении класса R-языков с классом языков, распознаваемых конечными автоматами. Даются задания для выполнения расчетно-графической работы.

Составители: доц. Лиогонький М.И.,

доц. Береговая Т.А.

Введение

Настоящая методическая разработка предназначена для студентов ННГАСУ специализации «Информационные системы и технологии», изучающих курс «Дискретная математика». В ней изложены основные понятия теории конечных автоматов, которые являются простейшими математическими моделями систем дискретного действия. Теория систем дискретного действия охватывает широкий круг вопросов, связанных с исследованием возможности автоматизации тех или иных процессов переработки информации, созданием эффективных методов синтеза автоматов и анализа их работы. Разработка быстродействующих эффективных программ переводчиков и программ компиляторов опирается на интенсивно развивающуюся теорию формальных языков и грамматик. В данной методической разработке вводятся и изучаются простейшие формальные языки, которые называются регулярными языками, и показывается, что класс регулярных языков совпадает с классом языков распознаваемых конечными автоматами.

Предлагаемая методическая разработка восполняет отсутствие в ННГАСУ доступной литературы по данным вопросам. Авторы разработки с некоторыми разъяснениями конспективно изложили материал, содержащийся в работе Д.И.Когана и Т.С. Бабкиной «Основы теории конечных автоматов и регулярных языков» [1].

Дополнительные сведения по вопросам, рассматриваемым в методической разработке можно найти в книгах [2], [3], [4], [5], [6].

§1. Языки над конечным алфавитом. Операции над языками

Конечным алфавитом будем называть произвольное непустое конечное множество $A=\{a_1, a_2, ..., a_n\}$, элементы которого будем называть ***буквами алфавита***. В качестве примеров алфавитов можно рассматривать латинский алфавит, состоящий из 26 букв, или кириллицу, состоящую из 33 букв. Для записи неотрицательных целых чисел в десятичной системе счисления используется алфавит $A=\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, а для их записи в двоичной системе счисления используется алфавит $A=\{0,1\}$.

Словом в алфавите $A=\{a_1, a_2, ..., a_n\}$, называется произвольная запись α вида $a_{i_1}a_{i_2}... a_{i_l}$, где $a_{i_1}, a_{i_2},..., a_{i_l}$ какие-то буквы этого алфавита, причем одна и та же буква в составе слова может встречаться многократно. Количество букв l в слове α называется его ***длиной***. Длина слова α обозначается так: $l(\alpha)$. Символом λ обозначается единственное ***пустое слово***, имеющее нулевую длину. В алфавите $A=\{a_1, a_2, ..., a_n\}$ можно записать n^l различных слов длины l, где l=0, 1, 2, ... Запись A^* будет в дальнейшем обозначать множество всех слов в алфавите A, включая пустое слово. Очевидно, что множество A^* является ***счетным*** множеством, т.к. все его элементы можно выстроить в виде некоторой последовательности. Например, в начало этой последовательности можно записать пустое слово, затем записать все слова длины 1, затем все слова длины 2 и т.д.

Если α и β – два произвольных слова в алфавите A, то запись $\alpha\beta$ будет обозначать новое слово γ в алфавите A, получаемое путем приписывания к буквам слова α справа всех букв слова β. Для любых слов α и β считается, что $\alpha\lambda=\lambda\alpha=\alpha$, $\alpha\lambda\beta=\alpha\beta$.

Языком в алфавите A называется произвольное подмножество слов L из A^*. Если множество L конечно (бесконечно), то и язык, определяемый этим множеством, называется ***конечным*** (***бесконечным***). Язык L называем ***пустым***, если множество L пусто ($L=\varnothing$). Совокупности всех слов русского и всех слов английского языков представляют собой примеры конечных языков. Совокупность записей всех четных чисел в десятичной системе счисления представляет собой бесконечный язык. Множество всех слов русского алфавита (но не русского языка!), заканчивающихся гласной буквой, – тоже бесконечный язык. Язык L в алфавите A называется ***универсальным***, если $L=A^*$.

Теоретико-множественные операции позволяют ввести соответствующие операции над языками. Пусть L_1 и L_2 – языки в алфавите A.

1) ***Объединением*** языков L_1 и L_2 называется язык, обозначаемый через $L_1 \cup L_2$, слова которого принадлежат по меньшей мере одному из этих языков.
2) ***Пересечением*** языков L_1 и L_2 называется язык, обозначаемый через $L_1 \cap L_2$, слова которого принадлежат каждому из этих языков.
3) ***Разностью*** языков L_1 и L_2 называется язык, обозначаемый через $L_1 \backslash L_2$, слова которого принадлежат языку L_1, но не принадлежат языку L_2.
4) Пусть L – язык в алфавите A. ***Дополнением*** этого языка до универсального называется язык, обозначаемый через L^c, слова которого принадлежат A^*, но не входят в состав языка L. Иными словами, $L^c=A^* \backslash L$.

Если, например, язык L состоит из всех слов латинского алфавита A, содержащих хотя бы одну букву a, то язык L^c состоит из слов, не содержащих букву a. Очевидно также, что операция разности представима через операции пересечения и дополнения, а именно: $L_1 \backslash L_2 = L_1 \cap (L_2)^c$.

Наряду с теоретико-множественными, существует еще несколько операций над языками.

5) ***Конкатенацией*** (***сцепкой***) языков L_1 и L_2 называется язык, обозначаемый через $L_1 \circ L_2$, произвольное слово γ которого может быть представлено в виде $\alpha\beta$, где α является словом языка L_1, а β является словом языка L_2.

Если один из языков L_1 и L_2 является пустым, то пустым является и язык $L_1 \circ L_2$. Отметим также, что если слово $\lambda \in L_1$, то $L_2 \subseteq L_1 L_2$, если же $\lambda \in L_2$, то

$L_1 \subseteq L_1 L_2$. Знак $\circ$ нередко опускается, и тогда конкатенация языков L_1 и L_2 записывается так: $L_1 L_2$.

Если языки L_1 и L_2 конечны, причем в составе первого языка m слов, а в составе второго n слов, то язык $L_1 L_2$ состоит максимум из mn слов.

Пусть дан алфавит $A=\{a,b\}$ и пусть язык $L_1 =\{a,\ abb\}$, а язык $L_2 =\{\lambda,\ bb\}$. Тогда язык $L_1 L_2 =\{a,\ abb,\ abbbb\}$.

6) Операция ***возведения языка L в целую неотрицательную степень*** определяется следующим образом:

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

$$L^1=L;$$

$$L^2=LL;$$

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

7) Пусть L– язык в алфавите A. ***Итерацией языка*** L называется язык, который обозначается как L^* и получается объединением всех неотрицательных степеней языка L. Иными словами $L^* = \bigcup_{i=0}^{\infty} L^i$.

Из определения следует, что пустое слово принадлежит итерации любого языка, а непустое слово α принадлежит итерации языка L тогда и только тогда, когда это слово можно разбить на некоторое количество последовательных частей так, что каждая часть принадлежит языку L. Если язык L содержит все однобуквенные слова алфавита A, то итерацией этого языка является универсальный язык A^*. Очевидно, что для любого языка L имеет место $(L^*)^*=L^*$.

Запись L^+ будет в дальнейшем обозначать язык, который получается из языка L^* путем изъятия пустого слова.

Проблема принадлежности слова языку состоит в том, чтобы по заданному языку L в алфавите A и по произвольному слову α из A^*, определить принадлежит ли слово языку L . Сложность решения данной проблемы существенно зависит от способа задания языка L. Если, например, язык L есть совокупность слов в русском алфавите, содержащих гласные буквы, то проблема принадлежности решается тривиально путем одного просмотра тестируемого слова α. Существенно сложнее проблема определения по заданному в десятичной системе счисления целому положительному числу, является ли оно простым. Если язык L состоит из всех слов русского языка, то по произвольному слову α, записанному кириллицей, не совсем просто выяснить, принадлежит ли слово α языку L , т.к. в данном случае задание языка L ***не достаточно формализовано***.

Если рассматриваемая проблема принадлежности имеет решающий алгоритм, то говорят, что она ***алгоритмически разрешима.*** Оказывается, что существуют формализовано определенные языки, для которых проблема принадлежности ***алгоритмически неразрешима***.

Если проблема принадлежности имеет решающий алгоритм, то его сложность можно охарактеризовать зависимостью числа выполняемых элементарных операций от длины тестируемого слова α. Изучаемый в данной методической разработке класс ***регулярных языков*** характеризуется тем, что для языков из этого класса существуют решающие алгоритмы, сложность которых линейно зависит от длины тестируемого слова.

§2. *R-выражения* и *R-языки*

Выше говорилось о том, что совокупность всех слов русского языка, несмотря на свою конечность, представляет собой неформализованный язык, ибо не существует какой-либо конструкции, которая бы порождала слова только русского языка. Ниже будет приведена одна из возможных конструкций, порождающая языки в конечном алфавите, использующая вышеприведенные операции над языками.

Пусть имеются конечный алфавит $A=\{a_1, a_2, ..., a_n\}$. Рассмотрим расширенный алфавит $\tilde{A}=\{a_1, a_2, ..., a_n, \varnothing, \lambda, \cup, (\ ,\ ), \circ, *\}$, получаемый из A добавлением букв $\varnothing, \lambda, \cup, (\ ,\ ), \circ, *$.

Некоторые, так называемые, правильно построенные слова в алфавите $\tilde{A}$ называются ***R-выражениями.*** Правила построения R-выражений следующие.

1) Символ $\varnothing$ является R-выражением;

2) символ λ является R-выражением;

3) символ a_i, где a_i – любая буква алфавита A, является R-выражением;

4) если слова φ и ψ являются R-выраженими, то слово $(\varphi\cup\psi)$ также является R-выражением;

5) если слова φ и ψ являются R-выражеными, то слово $(\varphi\circ\psi)$ также является R-выражением;

6) если слово φ является R-выражением, то слово $(\varphi)^*$ также является R-выражением;

7) R-выражениями являются только такие слова в алфавите $\tilde{A}$, которые могут быть построены путем конечного применения правил 1)- 6).

Выражения, записанные в соответствии с правилами 1) – 3), называются ***элементарными R-выражениями***, все остальные R-выражения именуются ***составными***. Составные R-выражения конструируются из элементарных по правилам 4) – 6); каждое правило может использоваться многократно.

R-выражение $(\varphi)^*$ называют ***взятием R-выражения φ в итерационные скобки.***

Часто в записях R-выражений опускаются знаки $\circ$ и скобки, отсутствие которых не может привести к недоразумениям, при этом предполагается, что наивысшим приоритетом обладает *, затем идет $\circ$, далее $\cup$. Если допустим $A=\{a, b\}$, то запись $a\cup ba^*$ трактуется как $(a\cup(b\circ(a)^*))$.

Из выше сформулированных правил нетрудно понять, что для произвольного слова в алфавите $\tilde{A}$ несложно определить, является оно R-выражением или нет. Если, допустим, $A=\{a, b\}$, то слово $(((a\cup b)\text{o}(b)^*))^*$ является R-выражением, а слово $(((a\cup b)\text{o}(b)^*))^*b)$ не является R-выражением, так как число открывающих скобок меньше, чем закрывающих.

Каждому R-выражению φ в алфавите $\tilde{A}$ можно поставить в соответствие некоторый язык L в алфавите A. При этом R-выражениям $\varnothing$, λ, a_i естественно сопоставить соответственно: пустой язык; язык, состоящий из одного пустого слова; и язык, состоящий из одного однобуквенного слова a_i. Если R-выражениям φ и ψ сопоставлены соответственно языки L_1 и L_2 то: R-выражению $(\varphi \cup\psi)$ естественно сопоставить объединение этих языков, т.е. язык $L_1\cup L_2$; R-выражению $(\varphi\circ \psi)$ естественно сопоставить конкатенацию этих языков, т.е. язык $L_1\circ L_2$; R-выражению $(\varphi)^*$ естественно сопоставить итерацию языка L_1, т.е. язык L_1^* .

В дальнейшем, если R-выражению φ в алфавите $\tilde{A}$ поставлен в соответствие язык L в алфавите A, то будем говорить, что язык L, определяется R-выражением φ . Языки в алфавите A, определяемые R-выражениями будем называть ***R-языками***. Приведем несколько примеров.

1) R-выражение $(a\cup b\cup c)^*$ определяет универсальный язык A^* для алфавита $A=\{a, b, c\}$.
2) R-выражение $a(a\cup b\cup c)^*a\cup a$ определяет язык, состоящий из всех слов алфавита $A=\{a, b, c\}$, которые начинаются и заканчиваются буквой a.

3) R-выражение $(aa \cup ab \cup ac \cup ba \cup bb \cup bc \cup ca \cup cb \cup cc)^*$ определяет множество всех слов четной длины в том же алфавите A.

4) R-выражение $(a \cup b \cup c)^* bbbb (a \cup b \cup c)^*$ определяет множество всех слов алфавита A, которые содержат по меньшей мере одно четырехкратное вхождение буквы b подряд.

5) Если рассмотреть алфавит A={0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, то R-выражение

$$(1 \cup 2 \cup 3 ... \cup 9)\ (0 \cup 1 \cup 2 \cup 3 ... \cup 9)^*(0 \cup 5)$$

определяет множество всех целых положительных чисел, записанных в десятичной системе счисления, которые кратны пяти.

Отметим, что взаимно однозначного соответствия между R-выражениями и определяемыми ими языками нет. Каждое R-выражение определяет один язык. Но один и тот же язык может определяться различными R-выражениями. Как R-выражение $(a \cup b \cup c)^*(b \cup a \cup cb)^*$, так и R-выражение $(a \cup c \cup b)^*$ задает множество всех слов алфавита $A=\{a, b, c\}$.

§3. Концепции конечного автомата и регулярного языка

Конечным автоматом называется некоторое устройство, имеющее конечное число состояний, предназначенное для прочтения слов некоторого конечного алфавита. Предполагается, что слово записано на некоторой ленте, составленной из ячеек, в каждой из которых записана одна буква алфавита. Прочтение ленты происходит в дискретные такты времени слева направо. Считается, что прочтение произвольного слова α автомат начинает из специально выделенного начального состояния. Прочтение очередной буквы в данном такте времени сопровождается переходом вправо к чтению следующей буквы и переходом в новое состояние, которое определяется читаемой в данном такте буквой и состоянием, в котором автомат в данный такт находится. Над словом длины l автомат работает l тактов. Результат прочтения слова α определяется состоянием, в котором автомат оказывается по завершению этого слова.

Данное словесное описания конечного автомата и его работы может быть заменено следующим формальным определением.

Конечный автомат *К* * определяется как совокупность ***К*={ *A, Q, q*₀, *g, F*}**, где

$A=\{a_1, a_2, ..., a_n\}$ – входной алфавит;

$Q=\{q_0, q_1, q_2, ..., q_m\}$ – множество состояний автомата;

q_0 –выделенное начальное состояние автомата;

g – функция переходов, определенная на множестве $Q\text{x}A=\{(q_i, a_j)\mid i=1,..,m;\ j=1,...,n\}$ и принимающая значения во множестве Q (если $g(q_i,a_j)=q_k$, то это означает, что конечный автомат, находясь в состоянии q_i, по прочтении буквы a_j переходит в состояние q_k);

F –выделенное подмножество ($F\subseteq Q$) «хороших» состояний автомата.

Работа конечного автомата K по прочтению входного слова $\alpha = a_{i_1}a_{i_2}...a_{i_p}$ определяется следующим образом. Пусть $q_\alpha(t)$ обозначает состояние, в котором оказывается автомат K через t тактов работы над этим словом (здесь t=0,1,2,..., p):

$q_\alpha(0) = q_0$ (к началу работы автомат находится в состоянии q_0),

$q_\alpha(1) = g(q_\alpha(0), a_{i_1})$ (прочитав первую букву, автомат переходит в состояние $q_\alpha(1)$),

$q_\alpha(2) = g(q_\alpha(1), a_{i_2})$ (прочитав вторую букву, автомат переходит из состояния $q_\alpha(1)$ в состояние $q_\alpha(2)$),

. . .

$q_\alpha(p) = g(q_\alpha(p-1), a_{i_p})$ (прочитав все слово, автомат переходит из состояния $q_\alpha(p-1)$ в состояние $q_\alpha(p)$).

Считается, что слово α ***принимается*** автоматом K, если $q_\alpha(p) \in F$, т.е. по прочтению всего слова автомат находится в одном из «хороших» состояний.

Пусть $L(K)$ есть совокупность слов, принимаемых автоматом K. Совокупность $L(K)$ называется ***языком, распознаваемым данным конечным автоматом К***.

Язык L назовем ***регулярным***, если он распознается некоторым конечным автоматом.

Конечный автомат $K = \{A, Q, q_0, g, F\}$ удобно задавать так называемой ***диаграммой переходов***. Диаграмма переходов представляет собой ориентированный граф, множество вершин которого совпадает с множеством состояний $Q = \{q_0, q_1, q_2, ..., q_m\}$ и если $g(q_i, a_j) = q_k$, то из вершины-состояния q_i, идет дуга в вершину-состояние q_k с надписанной над ней буквой a_j. 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 надписываются над соответствующей дугой (вместо перечня всех букв допускается сокращенная запись «$x \in S$» или просто «S»). Если вершина-состояние q_i входит в F, то на диаграмме она выделяется жирным кружком.

На рис. 3.1 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

осуществляется 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.

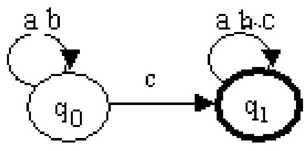


Рис.3.1

Другим способом задания конечного автомата является ***таблица переходов***. Таблица переходов автомата $K=\{Q, A, q_0, g, F\}$, где $Q=\{q_0, q_1, q_2, ..., q_m\}$ и $A=\{a_1, a_2, ..., a_n\}$, изображена в виде таблицы 1. Она имеет, кроме заголовочных, $m+1$ строку и n столбцов. Строки названы именами состояний, а столбцы именами букв входного алфавита. В клетке, стоящей на пересечении строки q_i и столбца a_j, записывается значение $q_k=g(q_i,a_j)$.

таблица 1

	a_1	a_2	….	a_n
q_0	$g(q_0,a_1)$	$g(q_0,a_2)$		$g(q_0,a_n)$
q_1	$g(q_1,a_1)$	$g(q_1,a_2)$		$g(q_1,a_n)$
q_m	$g(q_m,a_1)$	$g(q_m,a_2)$		$g(q_m,a_n)$

Состояния, входящие в совокупность F, выделяются в таблице переходов жирным шрифтом. Для автомата на рис.3.1 таблица переходов изображена в таблице 2.

таблица 2

	a	b	c
q_0	q_0	q_0	q_1
$\boldsymbol{q_1}$	q_1	q_1	q_1

Приведем еще два примера регулярных языков в алфавите $A=\{a, b, c\}$. L_2 – множество слов, содержащих подслово $\alpha=abbc$; и L_3 – 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_3, распознающих языки L_2 , L_3 соответственно, представлены на рисунках 3.2 , 3.3.

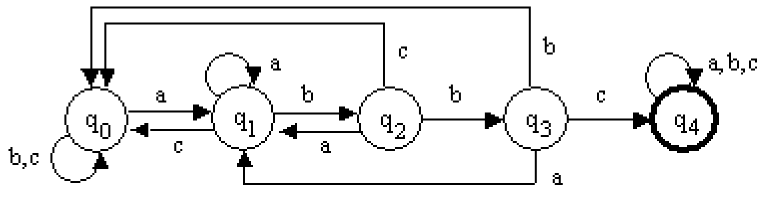


Рис. 3.2

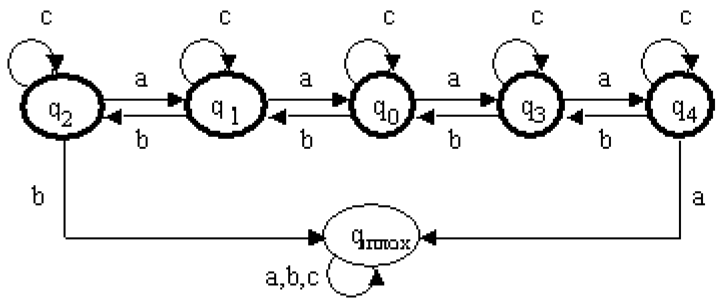


Рис. 3.3

Таблица 3 является таблицей переходов автомата K_3.

таблица 3

	a	b	c
$\boldsymbol{q_0}$	q_3	q_1	q_0
$\boldsymbol{q_1}$	q_0	q_2	q_1
$\boldsymbol{q_2}$	q_1	$q_{плох}$	q_2
$\boldsymbol{q_3}$	q_4	q_0	q_3
$\boldsymbol{q_4}$	$q_{плох}$	q_3	q_4
$q_{плох}$	$q_{плох}$	$q_{плох}$	$q_{плох}$

Информацию об обработанной части входного слова конечный автомат «помнит» своим состоянием. Так, например, автомат K_3, обработав некоторую часть входного слова, оказывается в состоянии q_2 (или q_1, или q_0), если в прочитанной им части входного слова число встреченных букв b превышает число встреченных букв a на 2 (или 1,или 0). Если же в прочитанной части входного слова число встреченных букв a превышает число встреченных букв b на 2 (или 1), автомат K_3 оказывается в состоянии q_4 (или q_3). Если в некоторый момент работы автомата абсолютная величина разности между числом обработанных букв a и числом обработанных букв b превысила 2, то автомат оказывается в состоянии $q_{\text{плох}}$.

Состояние конечного автомата называется ***поглощающим***, если любая входная буква не выводит автомат из этого состояния. Поглощающее состояние называется ***положительно поглощающим***, если оно принадлежит F, и ***отрицательно поглощающим*** в противном случае. В автомате K_1 (K_2) положительно поглощающим является состояние q_1 (q_4), в автомате K_3 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 (отрицательно) поглощающих состояний несколько, то они легко могут быть заменены одним поглощающим состоянием. Обычно в диаграмме переходов отрицательно поглощающее состояние не изображается. Если из некоторого состояния переход по некоторой букве не указан, то предполагается, что он ведет в отрицательно поглощающее состояние. На диаграмме автомата K_3 (рис.2.3) можно было отрицательно поглощающее состояние $q_{\text{плох}}$ не указывать и не рисовать стрелки в него идущие.

Как указывалось выше, неотрицательные целые числа, записанные в десятичной системе счисления, можно рассматривать как слова в алфавите $B=\{0, 1, 2, \ldots, 9\}$. Пусть язык $L^{(p)}$ состоит из пустого слова λ и из записей таких неотрицательных целых чисел, которые делятся нацело на некоторое положительное число p. Оказывается, что $L^{(p)}$ является регулярным язык. Распознающий $L^{(p)}$ конечный автомат $K^{(p)}$ может быть построен следующим образом. Состояний у конечного автомата $K^{(p)}$ будет ровно p, т.е. столько, сколько существует остатков при делении целых чисел на число p. Итак, пусть

$Q=\{q_0, q_1, q_2, ..., q_{p-1}\}$ и пусть ix есть целое неотрицательное число, десятичная запись которого получается приписыванием цифры x ($x\in\{0, 1, 2, ..., 9\}$) справа к десятичной записи числа i. Функция переходов g автомата $K^{(p)}$ определяется так: для произвольного состояния q_i и для произвольной цифры x, $g(q_i,x)= q_j$, если остаток от деления числа ix на p равен j. Единственным «хорошим» состоянием автомата $K^{(p)}$ следует считать q_0.

Из определения функции g следует, что автомат, обработав, начиная от начального состояния q_0, целое неотрицательное число α, записанное в десятичной системе счисления, оказывается в состоянии q_j тогда и только тогда, когда остаток от деления α на p равен j. На рис. 3.4 и 3.5 представлены диаграммы конечных автоматов $K^{(2)}$ и $K^{(3)}$ распознающих числа, делящиеся на 2 и 3 соответственно.

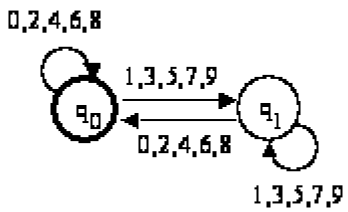


Рис.3.4

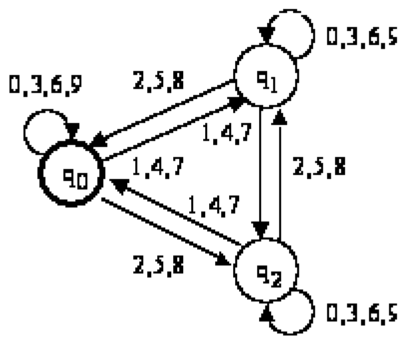


Рис.3.5

Пусть язык $L^{(p)}[\lambda]$ отличается от языка $L^{(p)}$ только тем, что он не содержит пустого слова λ. Очевидно, что язык, распознаваемый произвольным конечным автоматом, содержит пустое слово тогда и только тогда, когда начальное состояние q_0 этого автомата принадлежит множеству F. На рис. 3.6 представлен конечный автомат $K^{(3)}[\lambda]$, распознающий язык $L^{(3)}[\lambda]$.

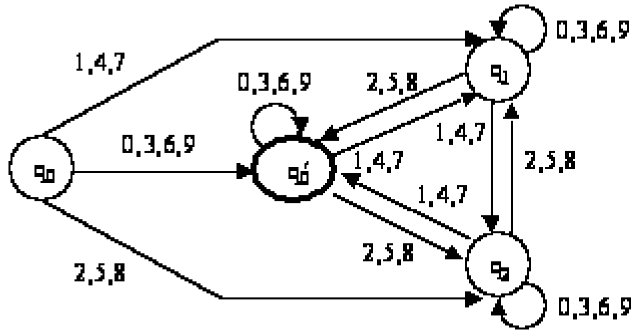


Рис.3.6

Для получения диаграммы переходов автомата $K^{(3)}[\lambda]$ в имеющейся диаграмме переходов автомата $K^{(3)}$ делаются следующие изменения: 1) начальное состояние q_0 автомата $K^{(3)}$ переименовывается в q_0'; 2) вводится новое начальное состояние q_0, в котором автомат $K^{(3)}[\lambda]$ «ведет» так же, как в состоянии q_0'; 3) единственным «хорошим» состоянием автомата $K^{(3)}[\lambda]$ считается q_0'.

Состояния, подобные состоянию q_0 на рис.2.6, называются ***невозвратными,*** выйдя из таких состояний, автомат в них никогда не возвращается.

Теорема 3.1. 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.

Пусть теперь язык L представляет конечный набор слов $\{\alpha_1, \alpha_2, \dots, \alpha_p\}$ в алфавите $A=\{a_1, a_2, \dots , a_n\}$. Пусть r длина самого длинного слова в языке L. Диаграмму конечного автомата $K(L)$, распознающего слова языка L можно построить в виде r-ярусного дерева. В нулевом ярусе находится начальное состояние q_0 . Из состояния q_0 выходят n дуг с надписями $a_1, a_2, \dots , a_n$ в различные состояния 1-ого уровня. Из каждого состояния 1-ого уровня выходят n дуг с надписями $a_1, a_2, \dots , a_n$ в свои различные состояния 2-ого уровня так, что в 2-ом уровне имеется ровно n^2 состояний. Из каждого состояния 2-ого уровня выходят n дуг с надписями $a_1, a_2, \dots , a_n$ в свои различные состояния 3-ого уровня так, что на 3-ем уровне имеется ровно n^3 состояний. И так далее до получения n^r состояний на уровне с номером r. Из всех состояний r-го уровня все дуги с любыми буквами ведут в отрицательно поглощающее состояние $q_{\text{плох}}$. Прочитав любое слово α, имеющее не более, чем r букв, автомат $K(L)$ переходит в одно вполне определенное состояние $q(\alpha)$. Состояния автомата $K(L)$, соответствующие словам $\alpha_1, \alpha_2, \dots, \alpha_p$ языка L объявляются «хорошими», т.е. определяющими множество F. Очевидно, что только эти слова и будут распознаваться построенным автоматом $K(L)$. На рис.3.7 приведен автомат, распознающий множество слов $L=\{\lambda, a, ab, bab, bbb\}$ в алфавите $A=\{a,b\}$ (состояние $q_{плох}$ не указано).

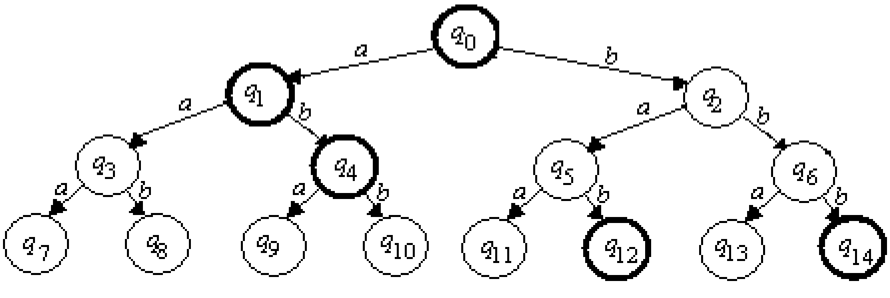


Рис.3.7

Теорема 3.2. Класс регулярных языков замкнут относительно основных теоретико-множественных операций – дополнения, объединения и пересечения

Пусть $K=\{Q, A, q_0, g, F\}$ – конечный автомат, распознающий язык L. Для того, чтобы получить конечный автомат, распознающий язык $L^c=A^*\backslash L$ (дополнение регулярного языка до универсального) надо в автомате K поменять местами «хорошие» и «плохие» состояния. Поэтому автоматом, распознающим язык L^c, является автомат $K^c=\{Q, A, q_0, g, Q\backslash F\}$.

Пусть L_1 и L_2 – регулярные языки, распознаваемые конечными автоматами $K^1=\{Q^1=\{q^1_0, q^1_1, q^1_2, ..., q^1_f\}, A, q^1_0, g^1, F^1\}$ и $K^2=\{Q^2=\{q^2_0, q^2_1, q^2_2, ..., q^2_h\}, A, q^2_0, g^2, F^2\}$ соответственно. Автомат $K^{\cup}=\{Q, A, q_0, g, F^{\cup}\}$, распознающий язык $L_1\cup L_2$, строится следующим образом:

1) множество $Q=\{(q^1_i, q^2_j) \mid i=0,1,\ldots f;\ j=0,1,\ldots h)$ то есть содержит $(f+1)*(h+1)$ внутренних состояний и каждое внутреннее состояние из Q содержит две компоненты: левая компонента является внутренним состоянием автомата K^1, а правая компонента является внутренним состоянием автомата K^2;

2) начальным состоянием автомата $K^{\cup}$ считается $q_0=(q^1_0, q^2_0)$;

3) функция переходов g определяется следующим образом: находясь в состоянии (q^1_i, q^2_j) и воспринимая букву a_k, автомат $K^{\cup}$ переходит в состояние $(g^1(q^1_i, a_k), g^2(q^2_j, a_k))$, т.е. по первой компоненте автомат $K^{\cup}$ выполняет действия автомата K^1, а по второй компоненте повторяет действия автомата K^2;

4) множество $F^{\cup}=(F^1\times Q^2)\cup(Q^1\times F^2)$, т.е. состояние автомата $K^{\cup}$ является «хорошим», когда после обработки слова α автомат $K^{\cup}$ окажется в состоянии, у которого либо первая компонента принадлежит совокупности F^1 (т.е. слово α распознается автоматом K^1 и, значит, $\alpha\in L_1$), либо вторая компонента принадлежит совокупности F^2 (т.е. слово α распознается автоматом K^2 и, значит, $\alpha\in L_2$).

Автомат $K^{\cap}=\{Q, A, q_0, g, F^{\cap}\}$, распознающий язык $L_1\cap L_2$, отличается от $K^{\cup}$ только тем, что $F^{\cap}=F^1\times F^2$. То есть состояние автомата $K^{\cap}$ является «хорошим», когда слово α распознается и автоматом K^1, и автоматом K^2 .

Теорема доказана.

На рис. 3.8 представлены два конечных автомата, распознающих языки L_1 и L_2. На рис. 3.9 представлен автомат, распознающий язык L_2^c, а на рис.3.10 и 3.11 изображены автоматы, распознающие языки $L_1\cup L_2$ и $L_1\cap L_2$ соответственно.

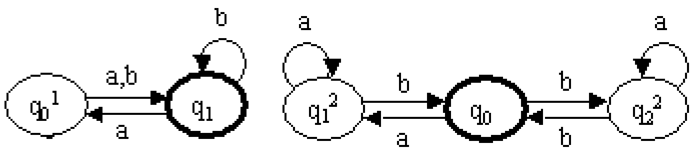


Рис. 3.8

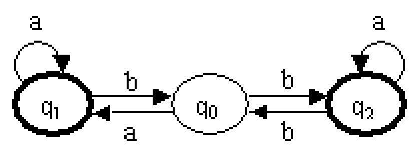


Рис. 3.9

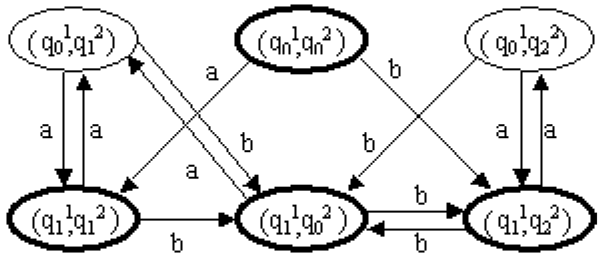


Рис. 3.10

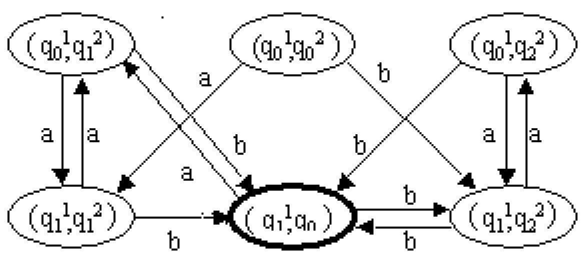


Рис. 3.11

Сделаем пояснение к рис.3.10 и рис.3.11 . Конечный автомат K_1 на рис.3.8, распознающий язык L_1 , имеет 2 состояния q^1_0 , q^1_1, а конечный автомат K_2 на рис.3.8, распознающий язык L_2 имеет 3 состояния q^2_0, q^2_1 , q^2_2. Поэтому конечные автоматы K_3 (на рис.3.10) и K_4 (на рис.3.11), распознающие языки $L_1\cup L_2$. и $L_1\cap L_2$, имеют 6 состояний, которые обозначены как: (q^1_0, q^2_0), (q^1_0, q^2_1), (q^1_0, q^2_2), (q^1_1, q^2_0), (q^1_1, q^2_1), (q^1_1, q^2_2). Начальным состоянием этих автоматов является состояние $(q^1{}_0, q^2{}_0)$. Выделенными состояниями в автомате для языка $L_1 \cup L_2$ (рис.3.10) будут такие пары, в которых хотя бы один компонент пары является выделенным состоянием в соответствующем автомате, т.е. это будут состояния: $(q^1{}_0, q^2{}_0)$, $(q^1{}_1, q^2{}_1)$, $(q^1{}_1, q^2{}_0)$, $(q^1{}_1, q^2{}_2)$. Выделенными состояниями в автомате для языка $L_1 \cap L_2$ (рис.3.11) будут такие пары, в которых каждый компонент пары является выделенным состоянием в соответствующем автомате, т.е. это будет единственная пара $(q^1{}_1, q^2{}_0)$. Поясним диаграммы переходов в данных конечных автоматах, которые, как выше было указано, являются совпадающими. Рассмотрим, например, переход из состояния $(q^1{}_0, q^2{}_0)$ при чтении входной буквы a. Автомат K_1 из состояния $q^1{}_0$ при чтении буквы a переходит в состояние $q^1{}_1$, автомат K_2 из состояния $q^2{}_0$ при чтении буквы a переходит в состояние $q^2{}_1$, поэтому рис.3.10 и рис.3.11 стрелка с буквой a идет из состояния $(q^1{}_{,}\, q^2{}_0)$ в состояние $(q^1{}_1, q^2{}_1)$. Аналогично реализованы и остальные переходы в автоматах на рис.3.10 и рис.3.11. Рассмотрим четыре слова в алфавите $\{a,b\}$: *abb, bbb, aab, aba*. Первое слово воспринимается автоматом K_1, но не воспринимается автоматом K_2, второе слово воспринимается автоматом K_2, но не воспринимается автоматом K_1, третье слово воспринимается обоими автоматами, а четвертое слово не воспринимается ни одним из автоматов K_1 и K_2. Из диаграмм переходов автоматов K_3 и K_4 легко видеть, что слова *abb, bbb* воспринимаются автоматом K_3, но не воспринимаются автоматом K_4, слово *aab* воспринимается обоими автоматами, а слово *aba* обоими автоматами не воспринимается.

Приведем пример языка в алфавите $A=\{a, b\}$, не являющегося регулярным. Пусть x^n (здесь x – произвольная буква алфавита) обозначает слово, состоящее из буквы x, повторенной n раз, $n \in \{1, 2, 3, ...\}$. Запись $a^n b^m$, обозначает результат приписывания справа к слову a^n слова b^m. Через $L^{a=b}$ 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, здесь n=1, 2, 3, ...

Теорема 3.3. Язык $L^{a=b}$ нерегулярен.

Доказательство проводим методом от противного. Пусть данный язык регулярен. Тогда существует конечный автомат $K^{a=b}$, распознающий $L^{a=b}$. Рассмотрим последовательность слов $a^1, a^2, ..., a^n$,.... Так как число состояний этого автомата конечно, то найдется пара слов a^k и a^m ($k \neq m$) по прочтению которых этот автомат окажется в одном и том же состоянии. Но в таком случае в одном и том же состоянии предполагаемый автомат $K^{a=b}$ окажется по прочтению слова $a^k b^k$ и слова $a^m b^k$. Но если первое слово принадлежит языку $L^{a=b}$, то второе слово этому языку не принадлежит. И, следовательно, предположение о регулярности языка $L^{a=b}$ приводит к противоречию.

Упражнения

В упражнениях 3.1-3.4 алфавит $A=\{a, b, c\}$; в упражнениях 3.5-3.8 алфавит $A=\{0, 1, 2, \ldots, 9\}$, каждое слово этого алфавита трактуется как запись в десятичной системе счисления целого неотрицательного числа.

Упражнение 3.1. Построить конечный автомат, распознающий язык L, в каждом слове которого не встречается более двух букв a подряд.

Упражнение 3.2. Построить конечный автомат, распознающий язык L, в каждом слове которого сочетание ab встречается не более двух раз.

Упражнение 3.3. Построить конечный автомат, распознающий язык L, в каждом слове которого содержится подслово $bbcc$.

Упражнение 3.4. Построить конечный автомат, распознающий язык L, каждое слово которого имеет длину не более 8 и содержит одинаковое число букв a и b.

Упражнение 3.5. Требуется построить конечный автомат, распознающий числа, кратные 4.

Упражнение 3.6. Требуется построить конечный автомат, распознающий числа, кратные 6.

Упражнение 3.7. Требуется построить конечный автомат, распознающий числа, кратные 7.

Упражнение 3.8. Требуется построить конечный автомат, распознающий числа, кратные 10.

Упражнение 3.9. Языки L_1 и L_2 распознаются конечными автоматами, диаграммы которых представлены на рис. 3.12 и 3.13 соответственно. Построить конечный автомат, распознающий языки $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 \setminus L_2$.

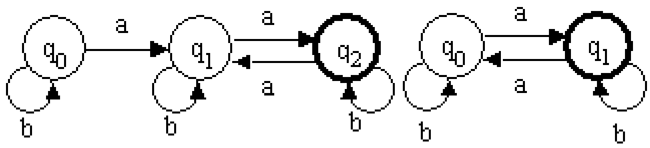


Рис. 3.12. Рис. 3.13

Упражнение 3.10. Построить конечный автомат, распознающий числа, делящиеся или на 5, или на 3.

Упражнение 3.11. Построить конечный автомат, распознающий числа, делящиеся или на 5, но не делящиеся на 3 .

§4. Недетерминированные конечные автоматы и определяемые ими языки

Недетерминированный конечный автомат (НДКА) есть совокупность $K' = \{Q, A, q_0, g', F\}$, где Q, A, q_0 и F имеют тот же смысл, что и для конечного автомата, а функция переходов g' каждой паре $(q_i \in Q, a_j \in A)$ ставит в соответствие некоторое непустое подмножество $g'(q_i, a_j)$ множества Q, и это означает, что НДКА, находясь в состоянии q_i и воспринимая букву a_j, к следующему такту может перейти в любое из состояний подмножества $g'(q_{i,j})$. Будучи запущенным в работу над произвольным словом $\alpha = a_{i_1} a_{i_2} .... a_{i_p}$ из начального состояния q_0, НДКА имеет много вариантов функционирования. Считается, что слово α ***распознается*** автоматом K' тогда и только тогда когда имеется такой вариант функционирования, при котором после прочтения последней буквы НДКА находится в одном из «хороших» состояний, т.е. существует последовательность состояний автомата $q_{r_0}, q_{r_1}, q_{r_2}, ... q_{r_p}$ такая, что

$$q_{r_0} = q_0;\quad q_{r_1} \in g'(q_{r_0}, a_{i_1});\quad q_{r_2} \in g'(q_{r_1}, a_{i_2}); \ldots.. q_{r_p} \in g'(q_{r_{p-1}}, a_{i_p}),$$

и при этом $q_{r_p} \in F$.

Язык L(K′), распознаваемый НДКА K', состоит из всех слов, которые он распознает.

Конечные автоматы, введенные в предыдущем разделе, называются ***детерминированными*** и являются частным случаем НДКА, когда все множества $g'(q_i, a_j)$ являются одноэлементными.

На рис. 4.1 дан пример НДКА 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).

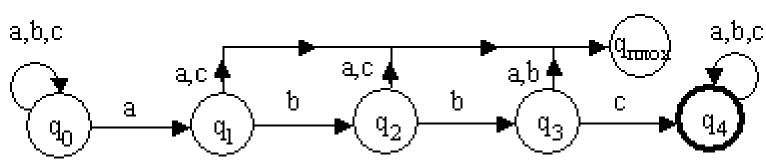


Рис.4.1.

Язык $L(K'_1)$ совпадает с совокупностью слов, содержащих в своем составе подслово *abbc* и, следовательно, совпадает с регулярным языком, распознаваемым конечным автоматом, изображенным на рис.3.2. Этот факт не является случайным, ибо справедлива следующая теорема.

Теорема 4.1. Языки, определяемые недетерминированными конечными автоматами, являются регулярными.

По произвольному НДКА $K'=\{Q, A, q_0, g', F\}$, распознающему язык $L(K')$, следующим образом строится конечный автомат $K''=\{Q'', A, q''_0, g'', F''\}$, такой что $L(K'')=L(K')$.

1) Множество Q'' состояний автомата K'' состоит из всевозможных непустых подмножеств множества Q (таким образом, если НДКА K' имеет n состояний, то строящийся конечный автомат K'' будет иметь (2^n-1) состояний).
2) $q''_0=\{q_0\}$, т.е. в качестве начального состояния автомата K'' принимается подмножество множества Q'', состоящее из одного элемента q_0.
3) Функция переходов автомата K'' строится следующим образом. Пусть q''_i есть некоторое состояние из Q'', (т.е. некоторое подмножество состояний НДКА K') и a_j есть некоторая буква алфавита A. Тогда значение $g''(q''_i, a_j)$ будет совпадать с подмножеством таких состояний множества Q, в которые автомат K' может перейти при чтении буквы a_j, находясь в одном из состояний подмножества q''_i. Иначе говоря, $g''(q''_i, a_j) = \bigcup_{q_s \in q''_i} g'(q_s, a_j)$.
4) Совокупность F'' определятся следующим образом: состояние q''_i будет «хорошим» в автомате K'', когда в подмножестве q''_i состояний автомата K' существует хотя бы один элемент из множества F. Т.е. $q''_i \in F'' \Leftrightarrow [q''_i \cap F] \neq \varnothing$.

Очевидно, что построенный изложенным способом детерминированный конечный автомат K'' распознает язык $L(K')$. Теорема доказана.

На рис. 4.2 представлен недетерминированный автомат K'_2, имеющий 3 состояния. На рис. 4.3 представлена диаграмма переходов ***эквивалентного ему*** (т.е. распознающего тот же самый язык) детерминированного конечного автомата K''_2, построенного в соответствии с приведенными выше правилами. На этой диаграмме обращает на себя тот факт, что в состояния $\{q_1\}$, $\{q_0, q_2\}$, $\{q_0, q_2\}$ и $\{q_0, q_1, q_2\}$ не входит ни одна стрелка и, следовательно, при начальном состоянии $\{q_0\}$ в функционировании автомата они не принимают участия. Такие состояния называются ***недостижимыми***. На рис. 4.4 представлена диаграмма переходов конечного автомата, эквивалентного K''_2, полученного из K''_2 исключением недостижимых состояний.

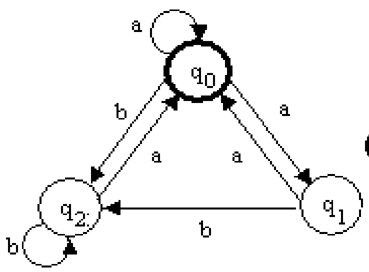


Рис.4.2

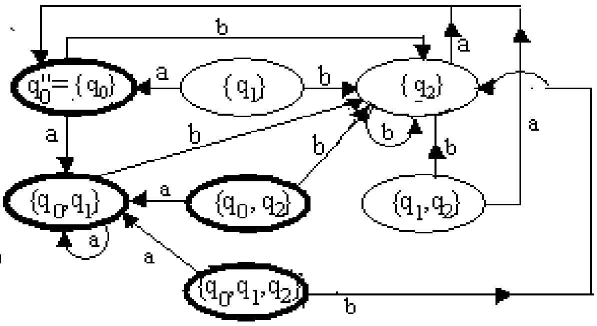


Рис.4.3

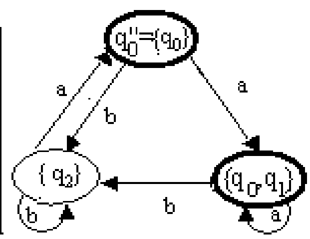


Рис.4.4

Пусть L_1 и L_2 – регулярные языки, распознаваемые конечными автоматами $K_1=\{Q^1, A, q^1_0, g^1, F^1\}$ и $K_2=\{Q^2, A, q^2_0, g^2, F^2\}$. Пусть n_1 и n_2 число состояний автоматов K_1 и K_2 соответственно. В §3 был приведен алгоритм построения конечного автомата, распознающего объединение языков L_1 и L_2, который содержал $n_1 * n_2$ состояний. Концепция недетерминированного конечного позволяет построить НДКА, распознающий объединение языков L_1 и L_2, но содержащий (n_1+n_2+1) состояний. Пусть D_1 и D_2 – диаграммы переходов, автоматов K_1 и K_2. Диаграмма D переходов искомого НДКА K', строится следующим образом. К диаграмме D_1 пристраивается диаграмма D_2 и вводится еще дополнительное состояние q_0, которое будет начальным в НДКА K'. Для каждой буквы x входного алфавита A из q_0 проводится две дуги с буквой x: одна дуга ведет в диаграмму D_1 туда же, куда ведет дуга с буквой x из состояния q^1_0, а другая ведет в диаграмму D_2 туда же, куда ведет дуга с буквой x из состояния

q^2_0. Множество F «хороших» состояний НДКА K', определяется как объединение множеств F^1 и F^2. В случае, когда начальное состояние хотя бы в одном из автоматов K_1 или K_2 является «хорошим», состояние q_0 также считается «хорошим» и включается в F.

На рис.4.5a представлена диаграмма конечного автомата K_1, распознающего язык L_1 в алфавите A={0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, состоящий из десятичных записей целых неотрицательных чисел, делящихся на 2. На рис.4.5b представлена диаграмма конечного автомата K_2, распознающего язык L_2 , состоящий из десятичных записей целых неотрицательных чисел, делящихся на 5. На рис.4.5c представлена диаграмма конечного автомата K', распознающего язык $L_1 \cup L_2$ состоящий из десятичных записей целых неотрицательных чисел, делящихся на 2 или на 5.

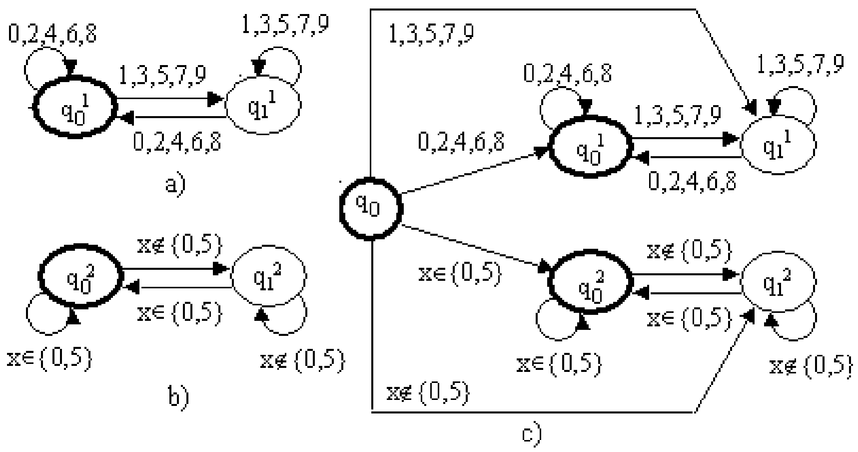


Рис.4.5

На рис.4.6c приведена диаграмма недетерминированный конечного автомата K', распознающего объединение двух языков, заданных конечными автоматами, изображенными на рис.4.6a и рис.4.6b. Состояние $q_0{}^2$ в построенном автомате K' оказывается недостижимым и его можно не изображать.

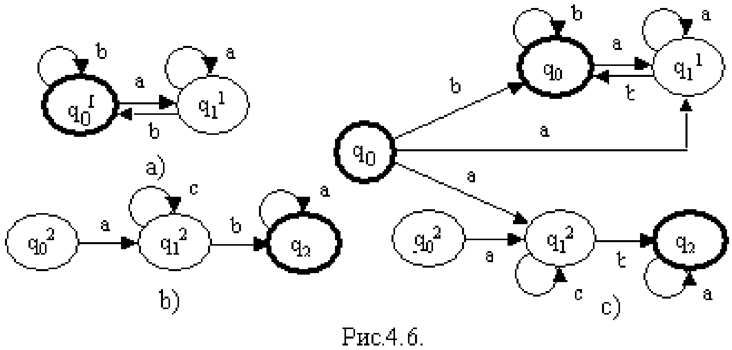


Рис.4.6.

Изложенная схема построения НДКА, распознающего объединение двух регулярных языков, применима и тогда, когда исходные языки определены НДКАми.

Упражнения.

Упражнение 4.1. По НДКАу, диаграмма которого представлена на рис. 4.7, построить эквивалентный ему детерминированный автомат.

Упражнение 4.2. По НДКАу, диаграмма которого представлена на рис. 4.8, построить эквивалентный ему детерминированный автомат.

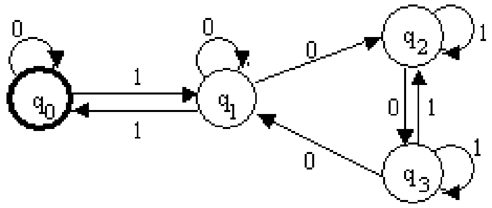


Рис.4.7

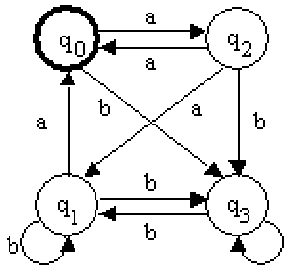


Рис.4.8

Упражнение 4.3. Языки L_1 и L_2 распознаются конечными автоматами, диаграммы которых представлены на рис. 4,9 и 4.10 соответственно. Построить конечный автомат, распознающий язык $L_1 \cup L_2$.

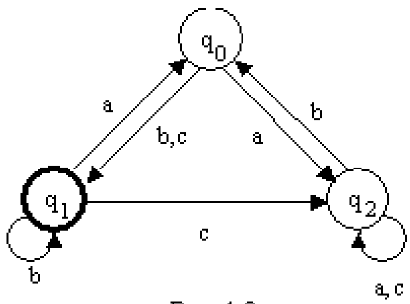


Рис.4.9

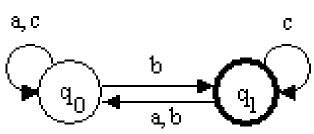


Рис.4.10

Упражнение 4.4. Построить недетерминированный конечный автомат, распознающий множество чисел, делящихся: а) на 25; б) на 20; в) на 125. Система счисления десятичная.

Упражнение 4.5. Пусть L_1 и L_2 есть языки в алфавите $\{a, b, c\}$. Язык L_1 состоит из слов, содержащих подслово $acbca$. Язык L_2 состоит из слов, содержащих подслово $bbba$. Построить НДКА, распознающий объединение этих языков.

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

Операции конкатенации, возведения в степень и итерации введены в § 1.

Теорема 5.1. Класс регулярных языков замкнут относительно операций конкатенации и возведения в целую положительную степень.

Доказательство данной теоремы конструктивно – ниже излагаются алгоритмы синтеза соответствующих конечных автоматов (вообще говоря, недетерминированных).

Пусть L_1 и L_2 – регулярные языки, распознаваемые автоматами (детерминированными или недетерминированными) $K_1=\{Q^1, A, q^1_0, g^1, F^1\}$ и $K_2=\{Q^2, A, q^2_0, g^2, F^2\}$ соответственно. Процедура синтеза диаграммы недетерминированного конечного автомата K, распознающего язык L_1L_2 зависит от того, являются ли начальные состояния автоматов K_1 и K_2 «хорошими» или нет, и, как будет видно из дальнейшего, эта процедура не зависит от того, являются автоматы K_1 и K_2 детерминированными или недетерминированными. При описании процедуры синтеза будет иметься в виду, что запись (q_i, x, q_j) обозначает дугу диаграммы автомата, на которой написана буква x и которая ведет из состояния q_i в состояние q_j.

<u>*1-ый случай.*</u> $q^1_0 \notin F^1$ (начальное состояние автомата K_1 не является «хорошим»).

В этом случае диаграмма искомого НДКАа K строится путем сцепления нарисованных рядом диаграмм D_1 и D_2 автоматов K_1 и K_2 дополнительными дугами. Именно, для каждой дуги (q^1_i, x, q^1_j) диаграммы D_1, такой что $q^1_j \in F^1$, строится дуга-дубликат (q^1_i, x, q^2_0), ведущая из состояния q^1_i диаграммы D_1 в начальное состояние q^2_0 диаграммы D_2. Начальным состоянием автомата K считается q^1_0. Совокупность F «хороших» состояний автомата K считается совпадающей с F^2. Таким образом, по прочтению слова α автомат K может попасть в свое «хорошее» состояние только в том случае, если его функционирование пройдет в какой-то такт через некоторую дополнительную дугу

(q^1_i, x, q^2_0). Но эта дуга является дубликатом некоторой дуги (q^1_i, x, q^1_j), в которой $q^1_j \in F^1$, что равносильно тому, что по окончанию этого такта прочитано некоторое слова α_1 из языка L_1. Чтение следующей буквы начинается из начального состояния автомата K_2, и попадание в «хорошее» состояние из F^2 означает, что оставшаяся часть α_2 слова α принадлежит L_2. Таким образом, $\alpha=\alpha_1\alpha_2$, где $\alpha_1 \in L_1$, а $\alpha_2 \in L_2$, следовательно, $\alpha \in L_1 \mathrm{o} L_2$.

На рис.5.2 изображена диаграмма НДКА автомата, распознающего язык L, который является конкатенацией языков L_1 и L_2, распознаваемых автоматами K_1 и K_2, диаграммы которых изображены на рис.5.1a и рис.5.1b, соответственно. Как видно из рисунка 5.1, автомат K_1 является недетерминированным. Жирные дуги на рис.5.2 изображают дуги –дубликаты.

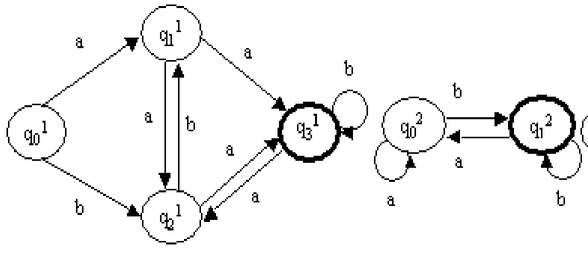


Рис.5.1

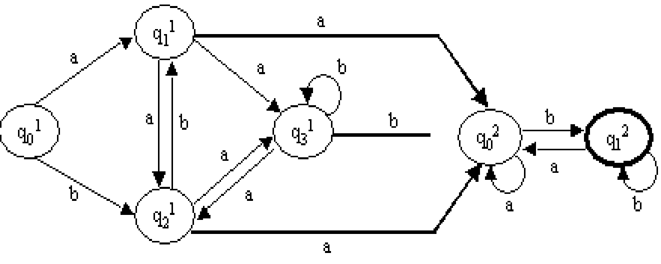


Рис.5.2

<u>*2-ой случай.*</u> $q^1_0 \in F^1$ и $q^2_0 \notin F^2$ (начальное состояние автомата K_1 является «хорошим», а начальное состояние автомата K_2 не является «хорошим»).

В этом случае в язык L_1 входит пустое слово и, следовательно, язык $L_1 \mathrm{o} L_2$ должен полностью содержать язык L_2, поэтому необходимо предусмотреть, чтобы искомый автомат K с самого первого такта мог повторять работу автомата K_2. В связи с этим для построения диаграммы искомого НДКАа K сцепка нарисованных рядом диаграмм D_1 и D_2 автоматов K_1 и K_2 происходит в 2 этапа. Сначала вводятся все дуги-дубликаты, описанные в 1-ом случае, а затем для каждой дуги (q^2_0, x, q^2_i) диаграммы D_2 вводится дуга-дубликат (q^1_0, x, q^2_i). Как и ранее, начальным состоянием автомата K объявляется состояние q^1_0. Совокупность F «хороших» состояний автомата K считается совпадающей с F^2. Таким образом, Таким образом, по прочтению слова α автомат K может попасть в свое «хорошее» состояние или, пройдя через дугу–дубликат первого типа (и тогда слово α должно иметь вид, $\alpha=\alpha_1\alpha_2$, где $\alpha_1 \in L_1$, а $\alpha_2 \in L_2$), или, пройдя через дугу–дубликат второго типа (и тогда слово $\alpha \in L_2$).

На рис.5.4 изображена диаграмма НДКА автомата, распознающего язык L, который является конкатенацией языков L_1 и L_2, распознаваемых автоматами K_1 и K_2 , диаграммы которых изображены на рис.5.3а и рис.5.3b, соответственно. Жирные дуги на рис.5.4 изображают дуги–дубликаты, описанные в первом случае, а пунктирные дуги изображают дуги–дубликаты, описанные во втором случае.

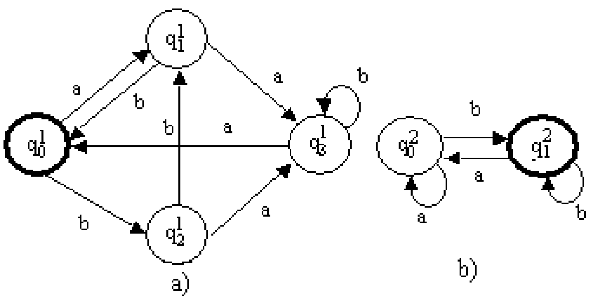


Рис.5.3

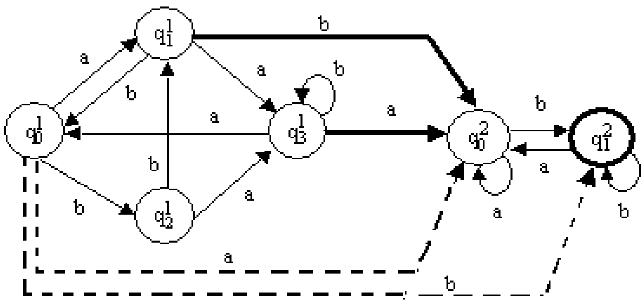


Рис. 5.4

<u>*3-ий случай.*</u> $q^1{}_0 \in F^1$ и $q^2{}_0 \in F^2$ (начальные состояния обоих автоматов являются «хорошими»).

В этом случае в язык L_1 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 2-го случая только в назначении множества его «хороших» состояний. В данном случае считается, что $F=F^2 \cup \{q_0{}^1\}$.

На рис.5.6 изображена диаграмма НДКА автомата, распознающего язык L, который является конкатенацией языков L_1 и L_2, распознаваемых автоматами K_1 и K_2, диаграммы которых изображены на рис.5.5а и рис.5.5b, соответственно. От диаграммы автомата на рис.5.4 диаграмма автомата на рис.5.6 отличается только дополнительным «хорошим» состоянием $q^1{}_0$.

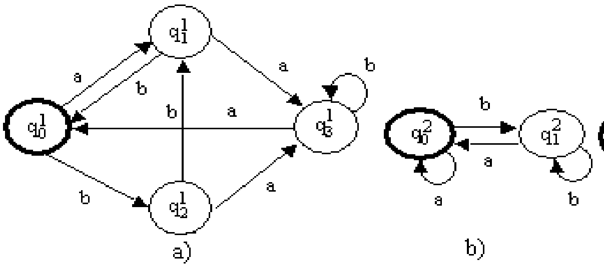


Рис.5.5

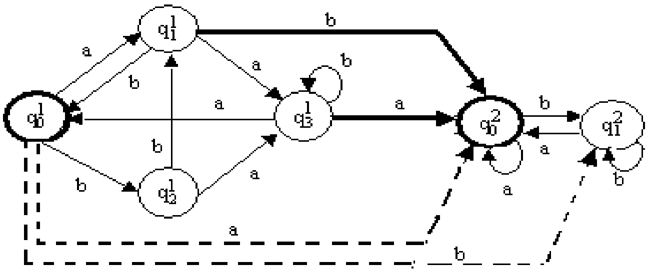


Рис.5.6

Замкнутость класса регулярных языков относительно операции конкатенации, влечет за собой замкнутость этого класса языков относительно операции возведения в любую целую неотрицательную степень.

Действительно, пусть L – произвольный регулярный язык. Язык $L^0 = \{\lambda\}$ регулярен в силу своей конечности. Так $L^1=L$, то L^1 – регулярный язык. Из предположения, что язык L^k регулярен, следует, что язык $L^{k+1}=L^kL$ как результат конкатенации регулярных языков L^k и L, является также регулярным. Из принципа математической индукции следует регулярность языка L^n при любом n.

Теорема 5.2. Класс регулярных языков замкнут относительно операции итерации.

Пусть L регулярный язык, распознаваемый автоматом (детерминированным или недетерминированным) $K=\{Q, A, q_0, g, F\}$.

Проще всего синтез диаграммы недетерминированного конечного автомата K^*, распознающего язык L^*, по диаграмме автомата K осуществляется в случае, когда начальное состояние q_0 автомата K является ***невозвратным*** (в диаграмме автомата K нет дуг, входящих в вершину q_0) . В этом случае для каждой дуги (q_i, x, q_j) такой, что $q_j \in F$, строим дугу-дубликат (q_i, x, q_0). Состояние q_0 считаем начальным и единственным «хорошим» состоянием в автомате K^*. Изложенный алгоритм синтеза автомата K^* называется алгоритмом P^*.

Нетрудно показать, что автомат K^* распознает язык L^*. В самом деле, поскольку начальное состояние q_0 является «хорошим», то слово λ распознается автоматом K^*. Пусть слово непустое α распознается автоматом K^* и пусть в процессе его чтения m раз автомат K^* оказывается в состоянии q_0 . Это равносильно тому, что слово α можно представить в виде конкатенации m подслов $\alpha_1, \alpha_2, \ldots, \alpha_m$ таких, что чтение каждого подслова α_s (s=$1,\ldots,m$) начинается из состояния q_0 и заканчивается переходом также в состояние q_0 . Но т.к. в автомате K состояние q_0 являлось невозвратным, то переход в состояние q_0 в автомате K^* может быть реализован только через некоторую дугу-дубликат (q_i, x, q_0). Автомат K при чтении подслова α_i, ведет себя также как автомат K^*, за исключением последнего такта, когда переход в автомате K будет осуществляться по некоторой дуге (q_i, x, q_j), являющейся одним из прообразов дуги (q_i, x, q_0). Но для любого такого прообраза состояние q_j является «хорошим» в K и, следовательно, при чтении подслова α_s, в диаграмме автомата K есть путь, ведущий из начального состояния в некоторое «хорошее» и, значит, $\alpha_s \in L$, а само слово α принадлежит L^m, а, следовательно, и L^* .

Из вышеизложенного ясно, что если для некоторого m слово $\alpha \in L^m$, т.е. $\alpha = \alpha_1 \alpha_2, \ldots, \alpha_m$, то в диаграмме построенного автомата K^* существует маршрут прочтения слова α, содержащий m дуг-дубликатов, ведущий в начальное («хорошее») состояние. Таким образом, автомат K^* действительно распознает язык L^* .

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

На рис.5.8 представлена диаграмма автомата K^*, распознающего язык L^*, являющийся итерацией языка, распознаваемого автоматом K, диаграмма которого изображена на рис.5.7. Начальное состояние автомата K, является невозвратным. Дуги дубликаты на диаграмме автомата K^* выделены толстыми линиями.

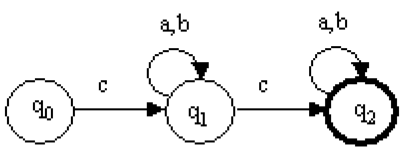


Рис.5.7

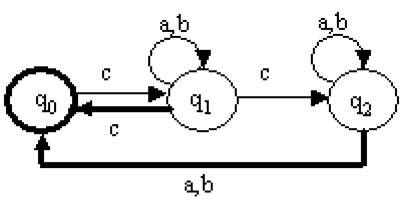


Рис.5.8

Автомат K распознает слова, которые начинаются на букву c и в каждом из которых буква c встречается ровно два раза. Автомат K* кроме пустого слова распознает слова α тогда и только тогда, когда выполняются два условия: 1) это слово начинается буквой c; 2) число вхождений буквы c в данное слово кратно двум.

Если состояние q_0 автомата K, распознающего язык L, является ***возвратным***, но «хорошим», процедура построения диаграммы автомата K*, распознающего язык L*, также реализуется с помощью алгоритма P*.

Если состояние q_0 автомата K, распознающего язык L, является ***возвратным,*** но не является «хорошим», то предварительно с помощью описанного ниже алгоритма P^0 строится диаграмма автомата K_1, эквивалентного автомату K, но имеющего невозвратное начальное состояние, а затем по диаграмме автомата K_1, с помощью алгоритма P* синтезируется диаграмма автомата K*, распознающего язык L*.

Алгоритм P^0 состоит в следующем. К диаграмме автомата K присоединяется новое состояние q'_0 и для каждой дуги диаграммы автомата K вида (q_0, x, q_i) вводится дуга-дубликат вида (q'_0, x, q_i), ведущая из нового состояния q'_0, в состояние q_i автомата K. Состояние q'_0 объявляется начальным состоянием автомата K_1, а множество его «хороших» состояний считается совпадающим с множеством хороших состояний автомата K. Непосредственно из построения следует, что слово α, распознается автоматом K тогда и только тогда, когда оно распознается автоматом K_1.

На рис.5.9 приведена диаграмма автомата K, начальное состояние q_0 которого является ***возвратным,*** но не является «хорошим» (дуги на диаграмме с отсутствующими буквами предполагаются идущими в отрицательное поглощающее состояние). На рис.5.10 приведена диаграмма автомата K_1 с начальным состоянием q'_0, эквивалентного автомату K, построенного по алгоритму P^0 (дуги-дубликаты выделены толстыми линиями). На рис.5.11 приведена диаграмма автомата K*, построенного с помощью алгоритма P* по диаграмме автомата K_1 (толстыми линиями выделены дуги-дубликаты, введенные по алгоритму P^0, а пунктирной линией выделена дуга-дубликат, введенная по алгоритму P^*).

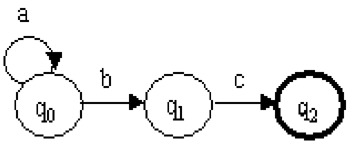


Рис.5.9

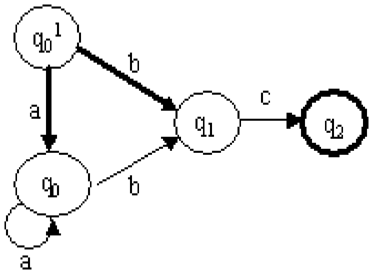


Рис.5.10

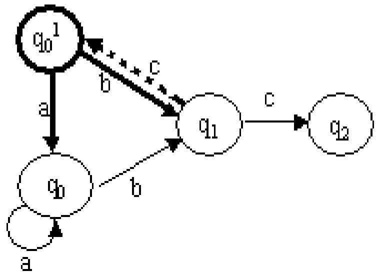


Рис.5.11

На рис. 5.12 представлена диаграмма переходов конечного автомата K, определяющего язык L, состоящий из слов, в которых буква c не встречается или встречается два раза. Так как начальное состояние автомата K является «хорошим», то для построения конечного автомата K^*, распознающего язык L^*, сразу применяется алгоритм P^*; полученный результат представлен на рис. 5.13.

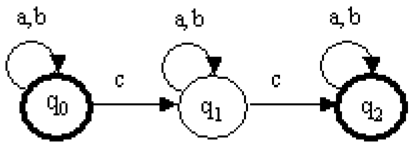


Рис.5.12

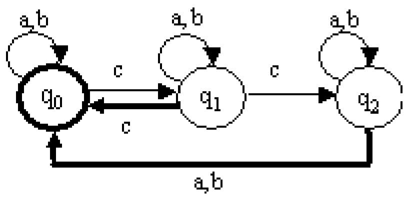


Рис.5.13

Упражнения.

Упражнение 5.1. Языки L_1 и L_2 распознаются конечными автоматами, диаграммы которых представлены на рис. 4.14 и 4.15 соответственно. Построить конечный автомат, распознающий язык L_1L_2. Рассмотреть следующие варианты задания совокупностей «хороших» состояний исходных автоматов:

1)$F^1=\{q_1\}, F^2=\{q_3\}$; 2) $F^1=\{q_0\}, F^2=\{q_1, q_2\}$; 3) $F^1=\{q_0\}, F^2=\{q_0, q_3\}$

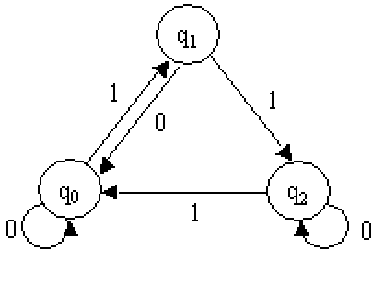


Рис.5.14

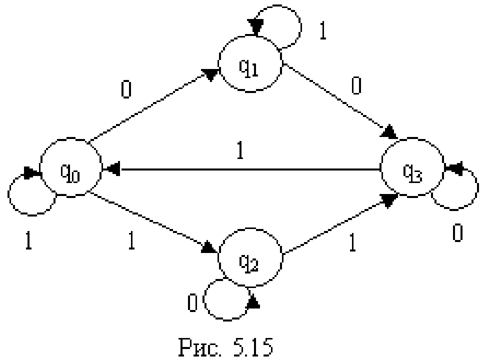


Рис. 5.15

Упражнение 5.2. Язык L распознается конечным автоматом, диаграмма которого представлена на рис. 4.16. Построить конечный автомат, распознающий итерацию этого языка. Рассмотреть следующие варианты определения совокупности «хороших» состояний исходного автомата: 1)$F=\{q_1, q_2\}$; 2) $F=\{q_0, q_4\}$.

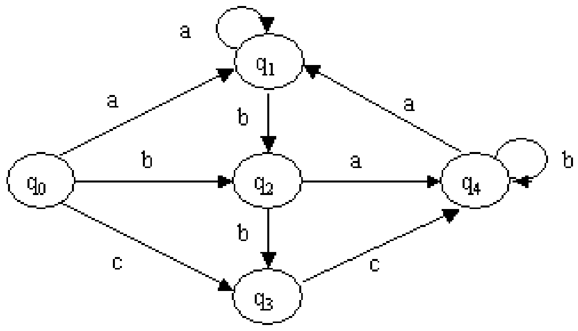


Рис. 5.16

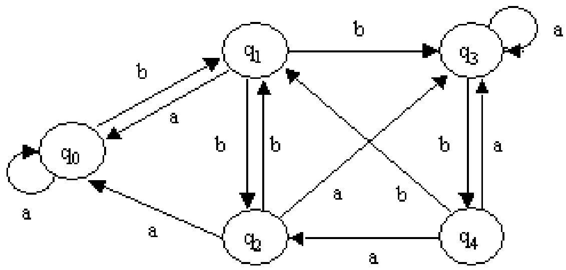


Рис. 5.17

Упражнение 5.3. Построить конечный автомат, распознающий итерацию языка, заданного автоматом, диаграмма которого представлена на рис. 4.17. Рассмотреть следующие варианты определения совокупности «хороших» состояний исходного автомата: 1)$F=\{q_1, q_2\}$; 2)$F=\{q_0, q_4\}$.

§6. Регулярные языки и *R*-языки. Терема Клини

В §2 было введены понятия R-выражения и R-языка в некотором алфавите $A=\{a_1, a_2,\ldots, a_n\}$. В каждом R-выражении использовались операции объединения, конкатенации и итерации, начиная с изначального множества, каковым являлся алфавит $A=\{a_1, a_2,\ldots, a_n\}$. В §3 были введены, регулярные языки, или иначе языки, распознаваемые конечными автоматами. В §3-§5 было показано, что любое конечное множество слов с буквами из алфавита $A=\{a_1, a_2,\ldots, a_n\}$ является регулярным множеством, и что класс регулярных множеств замкнут относительно операций объединения, конкатенации и итерации. Из всего этого следует, что ***каждый R-язык является регулярным.*** Но оказывается верно и обратное, и об этом говориться в следующей теореме.

Теорема 7.1. (Теорема Клини). Классы R-языков и регулярных языков совпадают.

В связи с данным результатом R-выражения обычно именуются *регулярными выражениями*. Теорема Клини содержит два утверждения:

1. Каждое регулярное выражение определяет регулярный язык
2. Каждый регулярный язык может быть задан некоторым R- выражением.

Справедливость теоремы Клини будет вытекать из существования алгоритмов решения двух задач: ***задачи синтеза и задачи анализа.***

Задача синтеза состоит в построении конечного автомата, распознающего язык, заданный некоторым регулярным выражением. Существование алгоритма решения этой задачи следует из приведенных выше рассуждений. И в этом параграфе более детализировано рассматриваются отдельные фрагменты такого алгоритма.

Задача анализа состоит в том, чтобы по диаграмме конечного автомата K записать R-выражение, для которого соответствующий ему R-язык совпадает с языком, распознаваемым автоматом K. Существование и описание алгоритма решения задачи анализа конечного автомата составляет содержание следующего параграфа.

Общий алгоритм синтеза конечного автомата по заданному R-выражению U может быть вкратце описан следующим образом.

1) В R-выражении U выбираются элементарные фрагменты, которые не содержат операций объединения и итерации (и которым соответствуют отдельные слова в алфавите A) и для каждого такого фрагмента строятся конечные автоматы, распознающие слова, соответствующие этим фрагментам.
2) Последовательно в соответствии с построением выражения U из этих фрагментов путем применения операций объединения, конкатенации и итерации, строятся конечные автоматы путем применения соответствующих операций к уже ранее построенным конечным автоматам. Причем следует отметить, что проще описанные процедуры реализовывать в классе недетерминированных конечных автоматов.

На рис.6.1a, рис.6.1b, рис.6.1c приводятся несколько простейших примеров синтеза конечных автоматом по R-выражениям. А именно, нетрудно видеть, что изображены диаграммы конечных автоматов соответствующих R-выражениям abc, $(a\cup b\cup c)^*$, $a^*b^*c^*$.

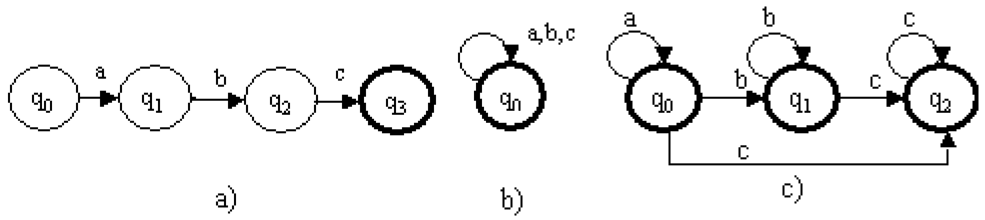


Рис.6.1

Пример. Построить автомат, распознающий язык, соответствующий следующему R-выражению $U= bc(abc\cup bd)^*caa$.

Данное R-выражение записано в алфавите A=$\{a, b, c, d\}$. Элементарными фрагментами в этом выражении являются: bc, abc, bd, caa. На рис.6.2 под буквами a), b), c), d) приведены диаграммы конечных автоматов, распознающих слова bc, abc, bd, caa,

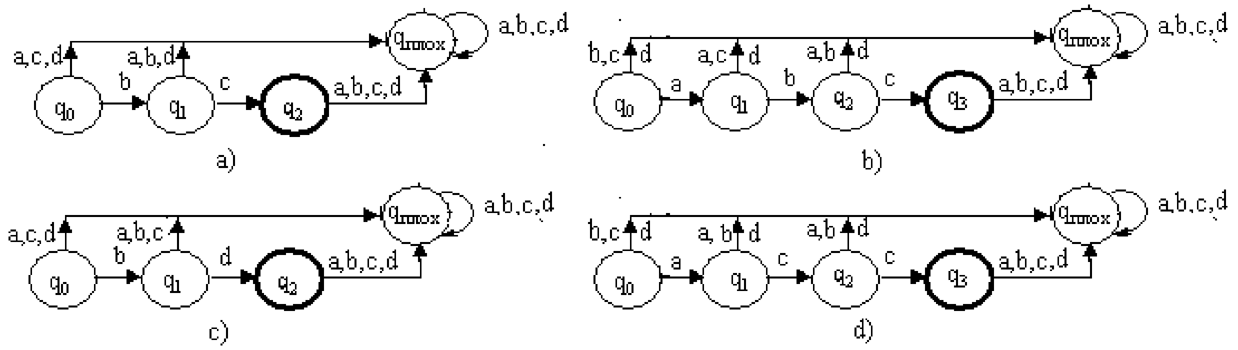


Рис.6.2

R-выражение *U* из выделенных фрагментов строится в такой последовательности: (*abc*∪*bd*), (*abc*∪*bd*)*, *bc*(*abc*∪*bd*)*, *bc*(*abc*∪*bd*)**caa*.

В такой же последовательности будут производиться операции над строящимися конечными автоматами. Конечный автомат, соответствующий выражению(*abc*∪*bd*), строится путем операции объединения диаграмм автоматов на рис.6.2b на рис.6.2c.

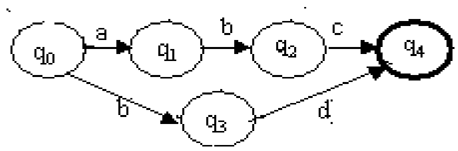


Рис.6.3

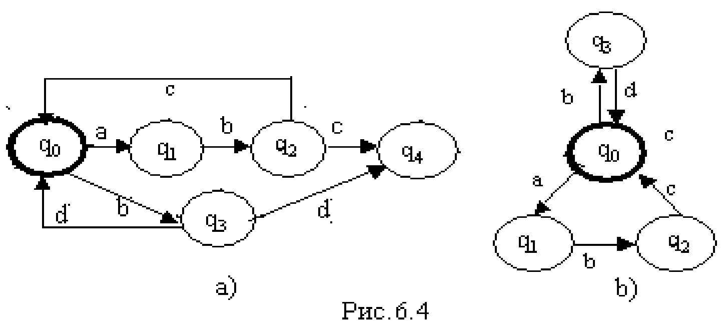


Рис.6.4

Нетрудно видеть, что на рис.6.3 изображена диаграмма искомого конечного автомата (здесь для наглядности изображения не указываются дуги, идущие в отрицательное поглощающее состояние), Т.к. состояние построенного конечного автомата является невозвратным, то применение алгоритма *P** (см. § 5) позволяет построить диаграмму автомата, распознающего язык, соответствующий выражению (*abc*∪*bd*)*. Эта диаграмма изображена на рис.6.4a. На рис.6.4b изображена диаграмма этого же автомата, на которой изображение состояния q_3 упущено т.к. из него дуги ведут лишь в отрицательное поглощающее состояние.

Конечный автомат, адекватный выражению *bc*(*abc*∪*bd*)*, строится в результате конкатенации конечных автоматов, изображенных не рис.6.2a и рис.6.4b. Результат конкатенации представлен рис.6.5a (для большей ясности состояния из диаграммы на рис. 6.4b помечены индексом 1). Так как начальное состояние на диаграмме рис.6.2a не является «хорошим», то применяется алгоритм, описанный в 1-ом случае при доказательстве теоремы 5.1. А именно: 1) из состояния q_1 на рис. 6.2a дугу-дубликат (утолщенная дуга) направляется в начальное состояние автомата $q^1{}_0$ на рис. 6.4b; 2) состояние q_0 диаграммы на рис. 6.2a принимается за начальное состояние искомого автомата; 3) состояние $q^1{}_0$ на диаграмме рис. 6.4b принимается за единственное «хорошее» состояние искомого автомата. На рис.6.5a видно, что состояние q_2 можно не изображать. Результирующая диаграмма конечного автомата, распознающего язык, соответствующий выражению $bc(abc\cup bd)^*$ (с общей нумерацией состояний) изображена на рис. 6.5b.

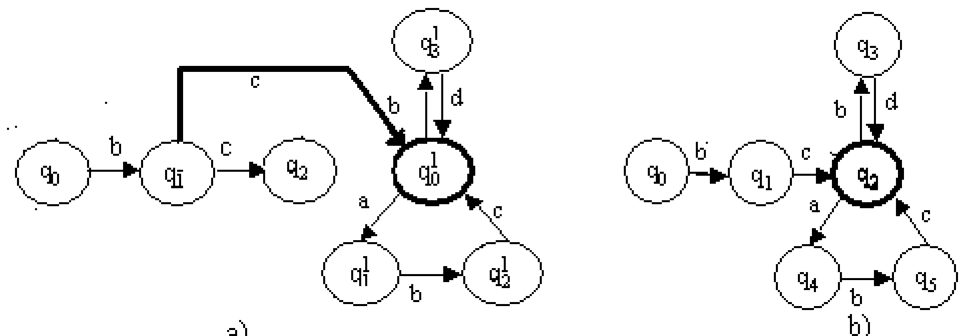


Рис.6.5

И, наконец, конечный автомат, адекватный выражению U, строится в результате конкатенации конечных автоматов, изображенных на рис.6.5b и рис.6.2d. Результат конкатенации представлен на рис.6.6a (состояния из диаграммы на рис.6.2d помечены сверху индексом 1).Утолщенные линии соответствуют введенным дугам-дубликатам. На рис.6.6a видно, что состояния, обозначенные символами q_2 и $q^1{}_0$ могут быть объединены. Окончательно диаграмма конечного автомата, распознающего язык, соответствующий выражению U (с общей нумерацией состояний) изображена на рис. 6.6b.

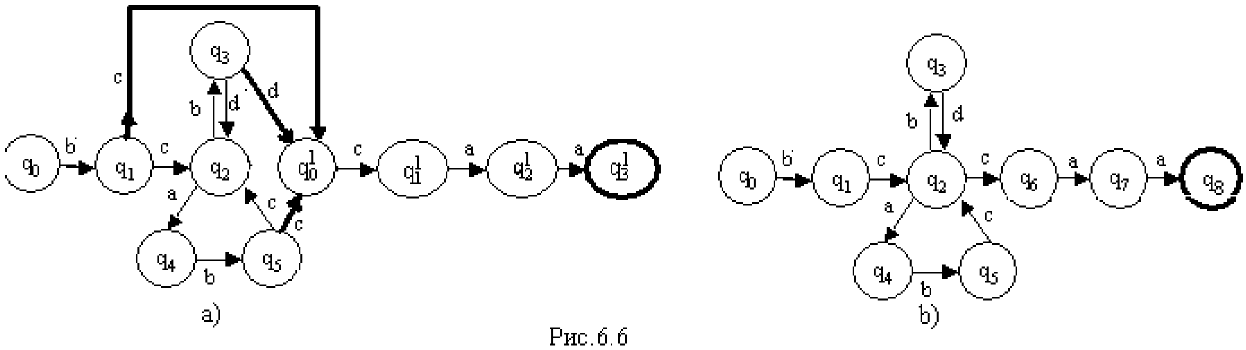


Рис.6.6

Пример Решить задачу синтеза конечного автомата $K(U)$, где $U=(a^*bc \cup cd)^* \circ (bac^* \cup aa^*c)^*$

Данное R-выражение U целесообразно представить в виде конкатенации $(U_1)^* \circ (U_2)^*$, где $U_1=a^*bc \cup cd$ и $U_2=bac^* \cup aa^*c$ и строить конечные автоматы в такой последовательности:

1) для U_1, 2) для $(U_1)^*$, 3) для U_2, 4) для $(U_2)^*$, 5) для $(U_1)^* \circ (U_2)^*$.

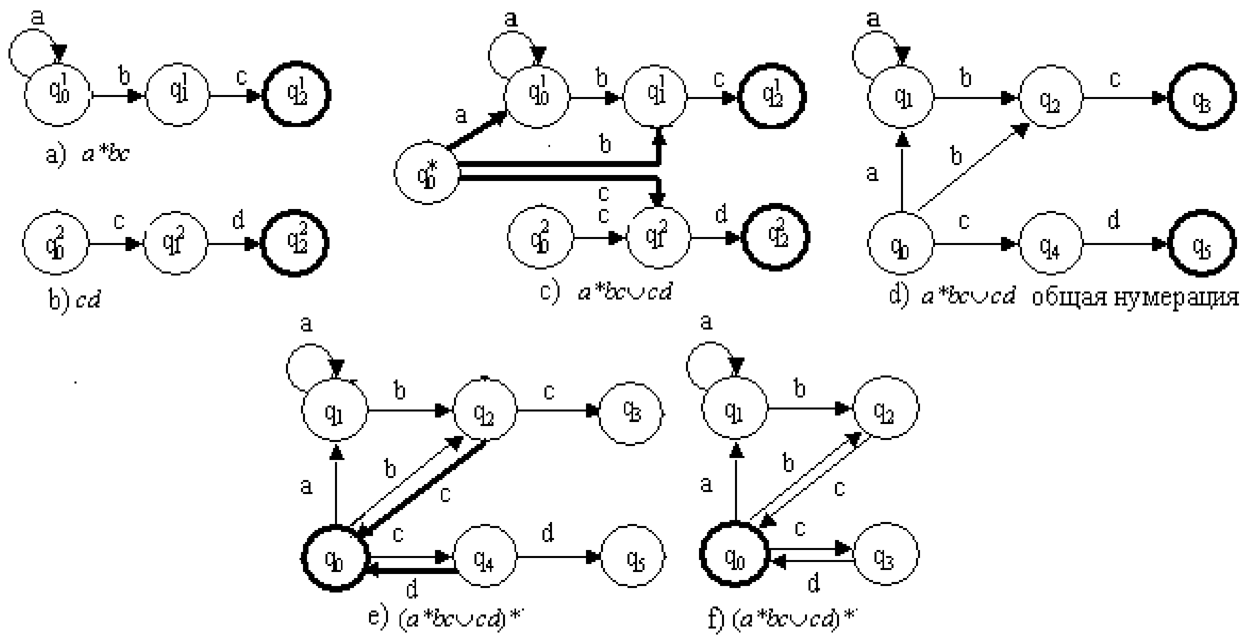
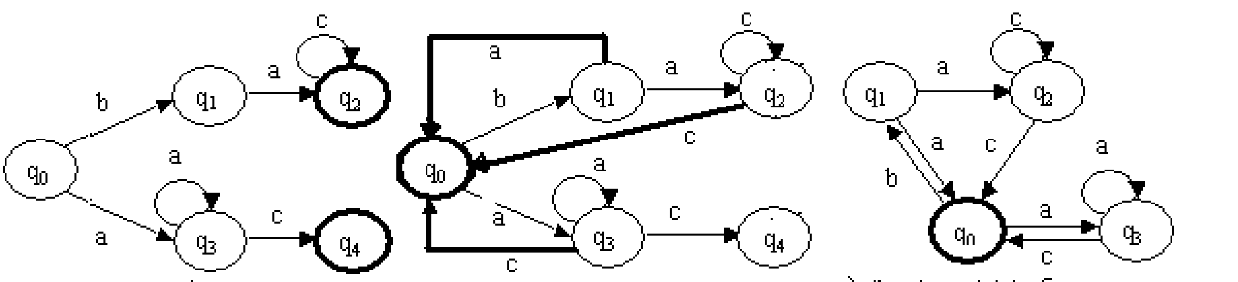


Рис.6.7

На рис.6.7, рис.6.8, рис.6.9 показаны последовательности синтеза диаграмм конечных автоматов, распознающих R-языки, соответствующие R-выражениям $(U_1)^*$, $(U_2)^*$, $(U_1)^* \circ (U_2)^*$.соответственно. Под каждым фрагментом рисунков записано R-выражение, которому соответствует изображенная диаграмма. Для отчетливого понимания каждого фрагмента необходимо вновь обратиться к тем параграфам, где описаны алгоритмы синтеза диаграмм для соответствующих операций. Например, на рис.6.7a и рис.6.7b изображены диаграммы автоматов распознающих языки, соответствующие R-выражениям a^*bc и cd.

На рис. 6.7c показано, как из диаграмм этих автоматом (состояния которых помечены верхними индексами 1 и 2 соответственно) синтезируется диаграмма объединения этих двух автоматов с помощью недетерминированных автоматов (см. §4): 1) вводится новое начальное состояние q^*_0, из которого выводятся дуги (толстые линии), повторяющие действия начальных состояний q^1_0 и q^2_0; 2) за «хорошие» состояния» принимаются «хорошие» состояния обоих автоматов.

На рис.6.7d приведена диаграмма того же автомата, но со сквозной нумерацией: нумерация состояний второго автомата продолжает нумерацию состояний первого автомата (при этом излишним оказывается состояние q^2_0). На рис.6.7f приведена диаграмма автомата, распознающая итерацию языка, распознаваемого автоматом, диаграмма которого приведена на рис.6.7d. Т.к. состояние q_0 на рис.6.7d является невозвратным, то для синтеза используется алгоритм P^* (см. §5 теорема 5.2). На рис:6.7e изображена диаграмма того же автомата, но состояния q_3 и q_5 как отрицательно поглощающие не изображаются.



a) $bac^* \cup aa^*c$ b) $(bac^* \cup aa^*c)^*$ c) $(bac^* \cup aa^*c)^*$ общая нумерация

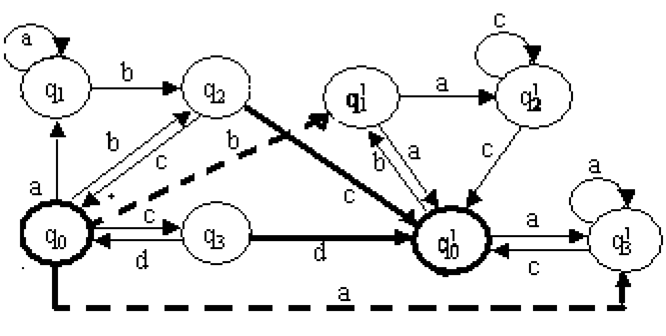
Рис.6.8

Аналогично, на рис.6.8b приведена диаграмма автомата, распознающая итерацию языка, распознаваемого автоматом, диаграмма которого приведена на рис.6.8a. Т.к. состояние q_0 на рис.6.8a является невозвратным, то снова для синтеза используется алгоритм P^*.

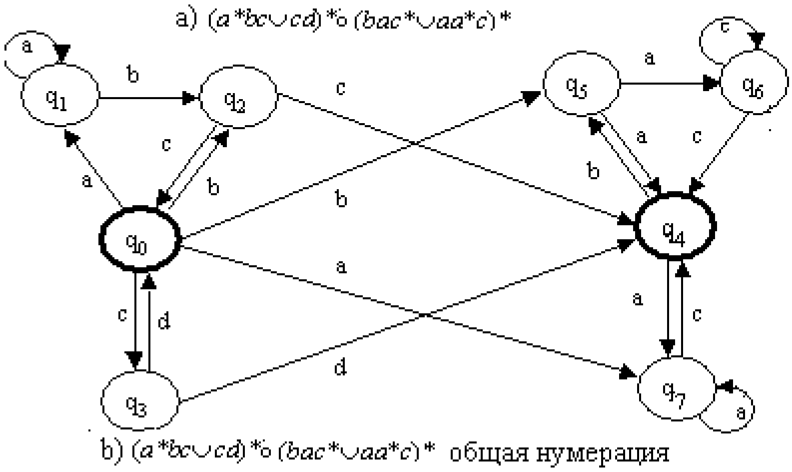
На рис. 6.8с изображена диаграмма того же автомата, но состояния q_4 как отрицательно поглощающее не изображается.

На рис.6.9а приведена диаграмма автомата, распознающая конкатенацию языков, распознаваемых автоматами, диаграммы которых приведена на рис.6.7е и на рис.6.8с, причем состояния из диаграммы на рис.6.8с помечены верхним индексом 1. Т.к. состояния q_0 и $q^1{}_0$ одновременно являются «хороши-ми» в каждом автомате, то для синтеза диаграммы искомого автомата исполь-зуется алгоритм, описанный в 3-ем случае при доказательстве теоремы 5.1. Толстые дуги являются дугами-дубликатами, описанными в 1-ом случае при доказательстве этой теоремы, а пунктирные дуги являются дугами-дубликатами, описанными во 2-ом случае при доказательстве этой теоремы.

На рис.6b приведена диаграмма того же автомата, но со сквозной нумера-цией: нумерация состояний второго автомата продолжает нумерацию состоя-ний первого автомата.



a) $(a^*bc \cup cd)^* \circ (bac^* \cup aa^*c)^*$



b) $(a^*bc \cup cd)^* \circ (bac^* \cup aa^*c)^*$ общая нумерация

Рис.6.9

Упражнения

В упражнениях 6.1-6.5 требуется решить задачу синтеза конечного автомата $K(\Phi)$ по заданному R-выражению Ф.

Упражнение 6.1. $\Phi=(ab^*c\cup ba)^*\cup ab^*c^*$.

Упражнение 6.2. $\Phi=((ba^*c\cup ba)^*\cup(a^*c)^*)^*$.

Упражнение 6.3. $\Phi=(ab^*a^*b\cup bac^*)^*$.

Упражнение 6.4. $\Phi=(a^*bc^*\cup ab^*c)^*\cup(ac^*\cup a^*c)^*$.

Упражнение 6.5. $\Phi=((abc^*\cup(ab)^*)^*cde(ab^*\cup b^*ca)^*)^*$.

§7. Алгоритм анализа конечного автомата

В этом параграфе доказывается второе утверждение теоремы Клини: каждый регулярный язык может быть задан соответствующим *R*-выражением. Доказательство приводится конструктивное, т.е. показывается, как построить *R*-выражение, *R*-язык для которого совпадает с языком, распознаваемым данным конечным автоматом (детерминированным или недетерминированным).

Излагаемый алгоритм принадлежит В.М. Глушкову.

Пусть задан конечный автомат $K=\{Q, A, q_0, g, F\}$. Путем, ведущим из состояния q_{i_0} в состояние q_{i_t}, в диаграмме данного автомата называется запись $\pi = q_{i_0} a_{j_1} q_{i_1} a_{j_2} q_{i_2} ... q_{i_{t-1}} a_{j_t} q_{i_t}$, в которой каждая тройка $(q_{i_{s-1}} a_{j_s} q_{i_s})$ (s=1,..,t) означает дугу, на которой записана буква a_{j_s} и которая ведет из состояния $q_{i_{s-1}}$ в состояние q_{i_s}. Начальное состояние q_{i_0}, а q_{i_t} - конечное состояние пути π. Путь π называется ***простым***, если в нем совпадают разве лишь начальное и конечное состояния. Простой путь, в котором совпадают начальное и конечное состояния называется ***простым циклом***. Если из записи простого цикла $q_x a_{j_1} q_{i_1} a_{j_2} ... q_{i_{t-1}} a_{j_t} q_x$ исключить символ состояния q_x, то получившаяся запись называется ***открытым путем*** из q_x в q_x. или ***x-циклом***.

Начальным комплексом $C_{нач}$ конечного автомата *K* называется запись, представляющая собой соединение с помощью символа "$\cup$'' всех простых путей этого автомата, ведущих из начального состояния q_0 в состояния, принадлежащие множеству $F\backslash\{q_0\}$; если $q_0 \in F$, то символ q_0 дополнительно присоединяется к записи как отдельный (вырожденный) путь.

Рассмотрим конечный автомат K_1, диаграмма которого представлена на рис.7.1.

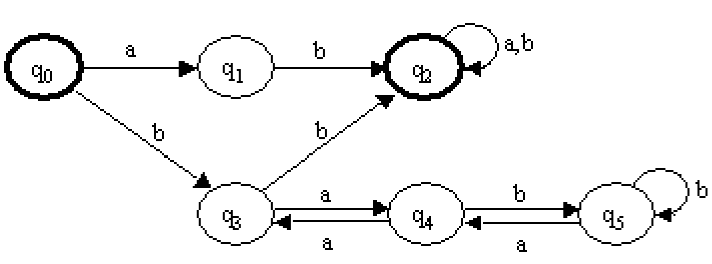


Рис.7.1

Начальный комплекс $C_{нач}$ автомата K_1 представляет собой следующую запись:

$$C_{нач} = q_0\, a\, q_1\, b\, q_2 \cup q_0\, b\, q_3\, b\, q_2 \cup q_0. \qquad (1)$$

Изъяв из записи (1) символы состояний, получаем R- выражение, определяющее список из трех слов: ab, bb, λ. Это полный перечень принадлежащих языку $L(K_1)$ слов, при чтении которых автоматом в соответствующих движениях по диаграмме отсутствуют циклы.

Комплексом состояния q_i конечного автомата K называется запись, обозначаемая $\boldsymbol{C_i}$ и представляющая собой взятое в итерационные скобки соединение с помощью символа "$\cup$'' всех i-циклов данного автомата. Если для состояния q_i нет i-циклов, то $\boldsymbol{C_i}=\lambda$.

Комплексами состояний конечного автомата K_1 (см. рис. 7.1) являются следующие записи:

$$\begin{aligned}
&C_0=C_1=\lambda;\\
&C_2=(a\cup b)^*;\\
&C_3=(a\, q_4\, a)^*; \qquad (2)\\
&C_4=(a\, q_3\, a \cup b\, q_5\, a)^*;\\
&C_5=(a\, q_4\, b \cup b)^*.
\end{aligned}$$

Пусть M – произвольное подмножество номеров состояний конечного автомата K, и пусть $i \notin M$; ***условным комплексом*** состояния q_i называется запись, обозначаемая $\boldsymbol{C_i(M)}$ и получающаяся из $\boldsymbol{C_i}$ путем удаления тех i-циклов данного автомата, в которых присутствуют состояния с номерами из подмножества M. Если $\boldsymbol{C_i}=\lambda$, то и $\boldsymbol{C_i(M)}=\lambda$.

Комплекс C_x произвольного состояния q_x – это частный случай условного комплекса данного состояния, когда $M=\varnothing$ т.е. $C_x=C_x(\varnothing)$.

Для конечного автомата K_1 (см. рис. 7.1), например, условный комплекс $C_4(3)$ получается путем удаления из записи C_4 (4-цикла) aq_3a, а условный комплекс $C_4(3,5)$ получается путем удаления из записи C_4 (4-циклов) aq_3a и bq_5a. Таким образом: $C_4(3)=(bq_5a)^*$, $C_4(3,5)=\lambda$.

Для описания алгоритма решения ***задачи анализа конечного автомата***, т.е. построения R-выражения, R-язык для которого совпадает с языком, распознаваемым данным конечным автоматом, необходимо ввести следующие три операции.

1) ***Операция подстановки в пути начального комплекса*** состоит в следующем. Пусть

$$\pi = q_0 a_{j_1} q_{i_1} a_{j_2} q_{i_2} a_{j_3} q_{i_3} \ldots q_{i_{t-1}} a_{j_t} q_{i_t}$$

– произвольный принадлежащий начальному комплексу путь. Результатом проведения ***операции подстановки над данным путем*** является запись, обозначаемая $P(\pi)$ и имеющая вид:

$$P(\pi) = C_0 a_{j_1} C_{i_1}(0) a_{j_2} C_{i_2}(0,i_1) a_{j_3} C_{i_3}(0,i_1,i_2) .. a_{j_{t-1}} C_{i_{t-1}}(0,i_1,i_2..i_{t-2}) a_{j_t} C_{i_t}(0,i_1,i_2..i_{t-2},i_{t-1}).$$

Если $\pi = q_0$, то $P(q_0) = q_0$. Иными словами, запись $P(\pi)$ получается из записи π путем замены символа q_0 на символ C_0 и замены любого символа q_s выражением $C_s(M)$, в котором множество M представляет собой перечисление номеров тех состояний, которые в пути π предшествовали состоянию q_s.

При выполнении ***операции подстановки в пути начального комплекса*** изложенным образом преобразуются все входящие в $C_{нач}$ пути.

Если обратиться к записи (1), то результатом выполнения операции подстановки в пути начального комплекса конечного автомата K_1, с диаграммой на рис.8.1, является следующая запись:

$$C_0\, a\, C_1(0)\, b\, C_2(0,1) \cup C_0\, b\, C_3(0)\, b\, C_2(0,3) \cup C_0. \qquad (3)$$

2) ***Операция расширения.*** Операция расширения состоит в замене в имеющейся записи каждого содержащегося в ней обозначения комплекса состояния или условного комплекса его значением, т.е. словом в алфавите, содержащем буквы входного алфавита, символы состояний автомата, знаки $\cup$, (,), *.

Применение операции расширения к записи (3) с использованием выражений (2) приводит к следующему R-выражению:

$$\lambda\, a\, \lambda\, b\, (a \cup b)^* \cup \lambda\, b\, (a q_4 a)^*\, b\, (a \cup b)^* \cup \lambda$$

или, что то же самое,

$$ab\,(a\cup b)^*\cup b\,(aq_4a)^*\,b\,(a\cup b)^*\cup\lambda.$$

3) ***Операция подстановки в открытые пути условных комплексов*** состоит в следующем. Пусть

$$\mu=a_{j_1}q_{i_1}a_{j_2}q_{i_2}a_{j_3}q_{i_3}\ldots q_{i_{t-1}}a_{j_t}$$

-произвольный входящий в $C_x(M)$ открытый путь (x-цикл). Результатом выполненной над данным путем операции подстановки является запись, обозначаемая $P_x(M,\mu)$ и имеющая вид

$$P_x(M,\mu)=a_{j_1}C_{i_1}(x,M)a_{j_2}C_{i_2}(x,M,i_1)a_{j_3}C_{i_3}(x,M,i_1,i_2)..a_{j_{t-1}}C_{i_{t-1}}(x,M,i_1,i_2,...,i_{t-2})a_{j_t}. \quad (4)$$

Замечание. В данной записи предполагается, что в записи $\boldsymbol{C_i}(M)$ множество представлено в виде перечисления номеров состояний в него входящих.

При выполнении операции подстановки в пути условного комплекса $C_x(M)$ указанным образом преобразуются все входящие в данный комплекс пути. Так, если $C_4(3)=(bq_5a)^*$, то в этот условный комплекс входит только один открытый путь (4-цикл) $\mu=bq_5a$ и, следовательно, $P_4(M,\mu)=b\,C_4(4,3)a$

Алгоритм анализа конечного автомата К содержит две процедуры:

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

Запись-результат номер 0 – это запись начального комплекса. Запись номер один – результат применения операции подстановки в пути начального комплекса. Запись номер два (и каждая следующая четная запись) – результат применения к предыдущей записи операции расширения. Запись номер три (и каждая следующая нечетная запись) – результат применения операции подстановки к путям всех условных комплексов, содержащимся в предыдущей записи. Процедура последовательного построения записей-результатов завершается, когда в последней полученной применением операции расширения записи отсутствуют символы состояний конечного автомата; эта запись является искомым *R*-выражением*,* результатом выполненного анализа.

В применении к представленному на рис. 7.1 конечному автомату K_1 первая входящая в состав алгоритма анализа процедура уже выполнена, начальный комплекс и комплексы всех состояний определены в (1) и (2). Вторая процедуры заключается в последовательном построении следующих записей:

0) $q_0\, a\, q_1\, b\, q_2 \cup q_0\, b\, q_3\, b\, q_2 \cup q_0$

(это запись начального комплекса рассматриваемого конечного автомата);

1) $C_0(a\ C_1(0)\ b\, C_2(0,1) \cup b\, C_3(0)\, b\, C_2(0,3) \cup \lambda)$

(это результат операции подстановки в пути начального комплекса; символ C_0 вынесен за скобку);

2) $ab\ (a\cup b)^* \cup b\underbrace{(a\ q_4\ a)^*}_{C_3(0)}\ b\,(a\cup b)^* \cup \lambda$

(это результат применения операции расширения к записи 1); учитывается, что C_0=$C_1(0)$=λ);

3) $ab\ (a\cup b)^* \cup b\ (a\ C_4(3,0)\ a)^*\ b\,(a\cup b)^* \cup \lambda$

(это результат применения операции подстановки к путям условных комплексов, имеющимся в предыдущей записи, а т.к. в записи 2) присутствует только один условный комплекс $C_3(0)=\ (a\ q_4\ a)^*$, то эта операция в соответствии с формулой (4) только к нему и применяется);

4) $ab\ (a\cup b)^* \cup b\ (a\ \underbrace{(bq_5a)^*}_{C_4(3,0)}\ a)^*\ b\,(a\cup b)^* \cup \lambda$

(это результат применения операции расширения к записи 3);

5) $ab\ (a\cup b)^* \cup b\ (a\ (b\ C_5(4,3,0)\ a)^*\ a)^*\ b\,(a\cup b)^* \cup \lambda$

(это результат применения операции подстановки к путям условных комплексов, имеющимся в предыдущей записи; а т.к. в записи 4) присутствует только один условный комплекс $C_4(3,0)=(bq_5a)^*$; то эта операция в соответствии с формулой (4) только к нему и применяется);

6) $ab(a\cup b)^*\cup b(a(b(b)^*a)^*a)^*b(a\cup b)^*\cup\lambda$

(это результат применения к предыдущей записи операции расширения).

Запись 6) не содержит символов состояний и работа алгоритма анализа конечного автомата заканчивается. Таким образом, для конечного автомата K_1 с представленной на рис.7.1 диаграммой переходов искомое R- выражение имеет вид:

$$R(K_1) = ab(a \cup b)^* \cup b(a(b(b)^*a)^*a)^*b(a \cup b)^* \cup \lambda .$$

Иногда некоторые пути начального комплекса имеют общее начало. В этом случае его удобно выносить за скобки.

Пример. Решить задачу анализа конечного автомата K_2, диаграмма которого представлена на рис. 7.2.

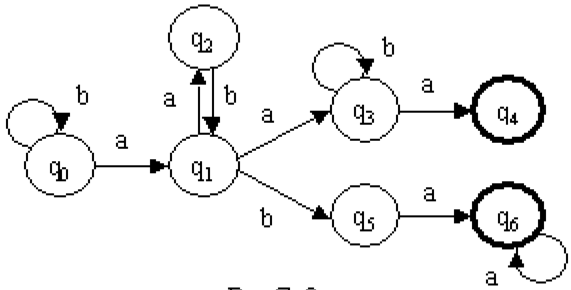


Рис.7.2

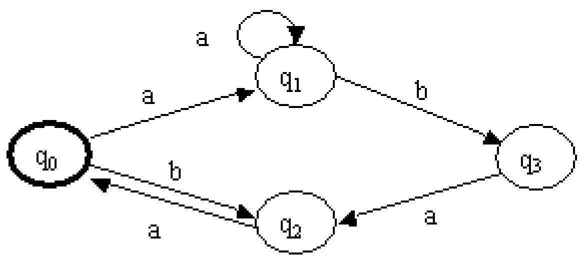


Рис. 7.3

В данном автомате в «хорошие» состояния ведут два пути: $q_0aq_1aq_3aq_4$ и $q_0aq_1bq_5aq_6$. Поэтому начальным комплексом данного автомата является запись:

$$C_{нач} = q_0aq_1aq_3aq_4 \cup q_0aq_1bq_5aq_6.$$

Т.к. указанные пути имеют общее начало q_0aq_1, то его можно вынести за скобки. Таким образом

$$C_{нач} = q_0aq_1(aq_3aq_4 \cup bq_5aq_6).$$

Комплексы его состояний таковы:

$$C_0 = b^*;\quad C_1 = (aq_2b)^*;\quad C_2 = (\,bq_1a)^*;\quad C_3 = b^*;\quad C_4 = \lambda;\quad C_5 = \lambda;\quad C_6 = a^*.$$

Записи-результаты выглядят следующим образом:

0) $q_0aq_1(aq_3aq_4 \cup bq_5aq_6)$;

1) $C_0aC_1(0)\ (aC_3(0,1)aC_4(0,1,3) \cup bC_5(0,1)aC_6(0,1,5))$;

2) $b^*a\underbrace{(aq_2b)^*}_{C_1(0)}(ab^*a \cup baa^*)$;

3) $b^*a(aC_2(0,1)b)^*(ab^*a \cup baa^*)$;

4) $b^*a(ab)^*(ab^*a \cup baa^*)$.

Задача анализа конечного автомата K_2 решена, $R(K_2)= b^*a(ab)^*(ab^*a\cup baa^*)$.

Пример. Решить задачу анализа конечного автомата K_3, диаграмма которого представлена на рис. 7.3.

Так как $F=\{q_0\}$, то начальный комплекс автомата K_3 состоит из единственного вырожденного пути:

$C_{нач}=q_0$.

Комплексы состояний данного автомата таковы:

$C_0=(bq_2a\cup aq_1bq_3aq_2a)^*$;

$C_1=(a\cup bq_3aq_2aq_0a)^*$;

$C_2=(aq_0b\cup aq_0aq_1bq_3a)^*$;

$C_3=(aq_2aq_0aq_1b)^*$.

Записи-результаты выглядят следующим образом:

0) q_0;

1) C_0;

2) $(\underbrace{bq_2a \cup aq_1bq_3aq_2a}_{C_0})^*$;

3) $(bC_2(0)a \cup aC_1(0)bC_3(0,1)aC_2(0,1,3)a)^*$;

4) $(ba \cup aa^*baa)^*$.

Итак, $R(K_3)=(ba \cup aa^*baa)^*$.

Пусть K – произвольный конечный автомат, имеющий N состояний. *Рангом условного комплекса* $C_i(M)$ состояния q_i данного автомата называется число элементов во множестве M. Ранг условного комплекса не может быть большим, чем N-1. Очевидно, что в записи любого условного комплекса ранга N-1 символы состояний отсутствуют.

Легко видеть, что в реализации изложенного алгоритма анализа конечного автомата при построении третьей и каждой следующей нечетной записи символ любого состояния заменяется символом условного комплекса ранга, как минимум, на единицу большего, чем ранг комплекса, в который данное состояние входило в предыдущей записи. Отсюда вытекает, что максимальное число нечетных записей, которые приходится построить в процедуре анализа конечного автомата K с N состояниями не превышает N. Данный факт подтверждает конечность изложенного алгоритма.

Упражнения

Упражнение 7.1. Решить задачу анализа конечного автомата, диаграмма переходов которого изображена на рис. 7.4.

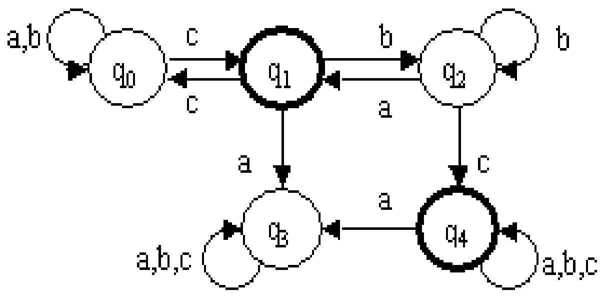


Рис. 7..4

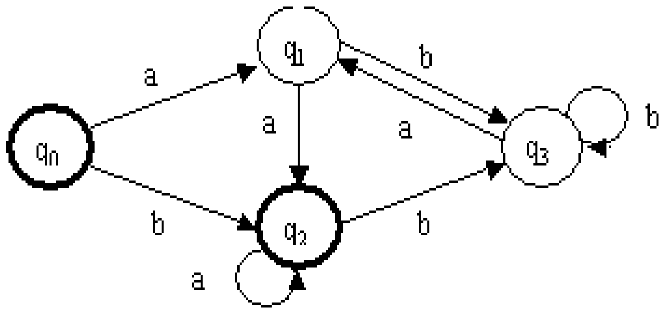


Рис. 7.5

Упражнение 7.2. Решить задачу анализа конечного автомата, диаграмма переходов которого изображена на рис. 7.5.

Упражнение 7.3. Решить задачу анализа конечного автомата, диаграмма переходов которого изображена на рис. 7.6.

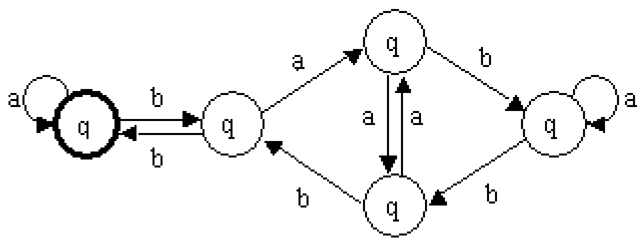


Рис. 7.6

§8. Алгоритмические проблемы для конечных автоматов и регулярных языков

Если задан конечный автомат $K=\{Q, A, q_0, g, F\}$, возникают вопросы относительно свойств, которыми обладает он и распознаваемый им язык $L(K)$. К числу вопросов, вызывающих наибольший интерес, можно отнести следующие.

1. Если задано произвольное слово α, то принадлежит оно языку $L(K)$.
2. Является ли распознаваемый этим автоматом язык $L(K)$ непустым.
3. Является ли распознаваемый этим автоматом язык $L(K)$ универсальным.
4. Является ли распознаваемый этим автоматом язык $L(K)$ бесконечным.

Если заданы два конечных автомата K_1 и K_2, то вызывает интерес следующие вопросы.

5. Является ли пересечение языков $L(K_1)$ и $L(K_2)$ непустым
6. Выполняется ли включение $L(K_1)\subseteq L(K_2)$.
7. Совпадают ли языки $L(K_1)$ и $L(K_2)$.

Не вызывает сомнения факт существования алгоритмов, применяя которые можно получить ответы на сформулированные вопросы.

Действительно, для ответа на 1-ый вопрос достаточно слово α «пропустить» через конечный автомат K. Слово α принадлежит языку $L(K)$ тогда и только тогда, когда в результате обработки данного слова автомат перейдет в состояние, принадлежащее совокупности F.

Для ответа на 2-ой вопрос, можно принять во внимание тот факт, что если в диаграмме состояний конечного автомата существует путь из начального состояния в одно из «хороших» состояний, то существует путь, длина которого не превышает числа (m-1), где m есть число состояний автомата. Отсюда следует, что если язык $L(K)$ не является пустым, то в нем присутствует хотя бы одно слово α, для которого $l(\alpha)\leq(m-1)$. Поэтому для ответа на 2-ой вопрос достаточно перебрать все слова с такой длиной.

Поиск ответа на 3-ий вопрос сводится к получению ответа на 2-ой вопрос только не для языка $L(K)$, а для языка $L(\overline{K})$, где $\overline{K}$ есть конечный автомат с той же диаграммой состояний, что и для автомата K, только $\overline{F}=Q\setminus F$. В самом деле, если $L(\overline{K})$ пустой, то $L(K)$ универсальный.

Если в языке $L(K)$ присутствует слово α, такое, что $m \le l(\alpha) \le (2m-1)$, то это означает, что в диаграмме автомата существует путь из начального состояния в «хорошее», содержащий цикл, а это означает, что язык $L(K)$ бесконечен. Таким образом, для получения ответа на 4-ый вопрос достаточно перебрать все слова с такой длиной.

Поиск ответа на 5-ый вопрос сводится к поиску ответа на 2-ой вопрос. А именно, по заданным автоматам K_1 и K_2 строится конечный автомат K, распознающий пересечение языков $L(K_1)$ и $L(K_2)$, и для автомата K решается вопрос, является ли язык $L(K)$ пустым.

Для ответа на 6-ой вопрос по имеющемуся автомату K_2 строится автомат $\overline{K}_2$ и ищется ответ на вопрос, пусто или нет пересечение языков $L(K_1)$ и $L(\overline{K}_2)$. Если пусто, то имеет место включение $L(K_1)\subseteq L(K_2)$.

Т.к. языки языки $L(K_1)$ и $L(K_2)$ совпадают тогда и только тогда, когда одновременно $L(K_1)\subseteq L(K_2)$ и $L(K_2)\subseteq L(K_1)$, то поиск ответа на 7-ой вопрос сводится к поиску ответа на 6-ой вопрос. Сначала решается вопрос о пустоте пересечения языков $L(K_1)$ и $L(\overline{K}_2)$, а затем решается вопрос о пустоте пересечения языков $L(K_2)$ и $L(\overline{K}_1)$.

Итак, действительно, все проблемы, связанные с поисками ответов на сформулированные вопросы являются алгоритмически разрешимыми. Но в теории алгоритмов исследуется также трудоемкость разрешающих алгоритмов.

Нетрудно понять, что алгоритм поиска ответа на 1-ый вопрос пропорционален длине тестируемого слова. Описанный выше алгоритм поиска ответа на 2-ой вопрос требует пропускания через автомат всех слов, по длине не превосходящих числа $(m-1)$. Но таких слов существует n^{m-1} , где n есть число букв в алфавите A. Однако в более обширной теории конечных автоматов показано, что существует алгоритм поиска ответа на 2-ой вопрос, для которого верхней оценкой числа необходимых элементарных операций, является число Cm^2, где C – константа, не зависящая от значения m. Т.к. поиск ответа на 3-ий вопрос сводится к поиску ответа на 2-ой вопрос, то для получения ответа также существуют алгоритмы с такой же верхней оценкой их трудоемкости. С такой же верхней оценкой трудоемкости существует алгоритм для получения ответа на 4-го типа вопросы. Поиск ответов на вопросы 5-7-го типов сводятся к поискам ответов на вопросы 2-го типа, но т.к. в этих вопросах фигурируют два автомата K_1 и K_2, то трудоемкость алгоритмов, дающих ответы, не превосходят числа $Cm_1^2m_2^2$, где m_1 и m_2 – число состояний автоматов K_1 и K_2 соответственно.

Приведенные оценки трудоемкости алгоритмов решения алгоритмических проблем, связанных с поиском ответов на перечисленные семь вопросов показывают, что каждая из них разрешима в полиномиально зависящем от размера входной информации времени.

Такого типа проблемы называются в теории алгоритмов P-разрешимыми. Проблемы, связанные с поиском ответов на некоторые другие вопросы, возникающие в теории конечных автоматов и языков, оказываются труднорешаемыми, т.к. для их решения не существует полиномиальных по верхней оценке временной вычислительной сложности алгоритмов. Такие проблемы называются NP-полными. В числе таких NP-полных проблем в теории конечных автоматов можно отметить проблемы, связанные с поиском ответов на следующие вопросы:

1. Если заданы два недетерминированных конечных автомата K_1 и K_2, то являются ли различными распознаваемые этими автоматами языки;
2. Если задано регулярное R-выражение, то является ли универсальным определяемый этим выражением язык.

Расчетно- графическая работа

Задание 1. Требуется построить конечный автомат, распознающий язык L в алфавите А={a, b,c}, если язык L задается следующим условием: слово α принадлежит языку L тогда и только тогда,

1.0 когда буква a встречается в слове α ровно один раз, а буквы b,c не менее двух раз.

1.1. когда в этом слове не встречается более двух букв a подряд.

1.2. когда в слове α сочетание ab встречается не более двух раз.

1.3. когда в слове α содержится подслово $bbcc$.

1.4. когда слово α имеет длину не более 8 и содержит одинаковое число букв a и b

1.5. когда слово α содержит четное число букв.

1.6. когда слово α содержит нечетное число букв a.

1.7. когда при наличии в слове α буквы a там встречается также и буква b.

1.8. когда каждая буква алфавита встречается в слове α не более двух раз.

1.9. когда буквы b,c встречаются в слове α ровно один раз,а буква a не менее двух раз.

Задание 2. Требуется построить конечный автомат в алфавите А={0,1,2,3,4,5,6,7,8,9}, распознающий числа,

2.0. кратные 6 .

2.1. кратные 7.

2.2. кратные 8.

2.3. кратные 10.

2.4. кратные 15.

2.5. кратные 20.

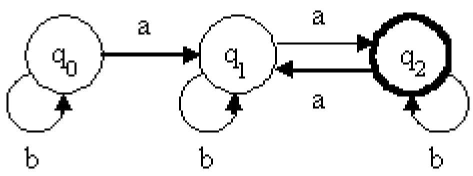
2.6. кратные 25.

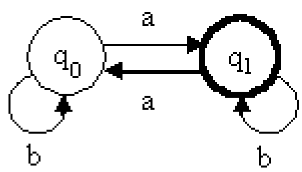
2.7. кратные 30.

2.8. кратные 50.

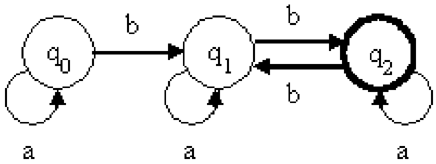
2.9. кратные 125.

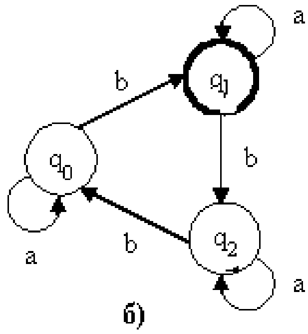
Задание 3. На диаграммах а) и б) представлены конечные автоматы, распознающие языки L_1 и L_2 соответственно. Построить конечный автомат, распознающий языки $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 \setminus L_2$, $L_1 L_2$



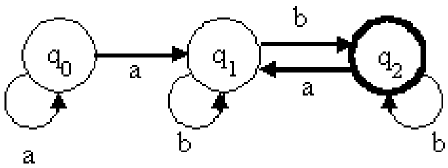


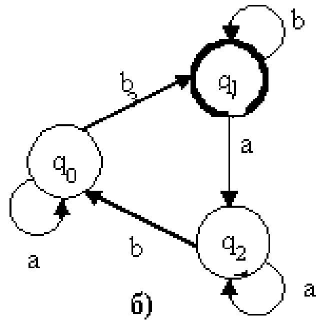
3.0. а) б)



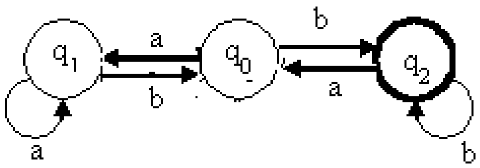


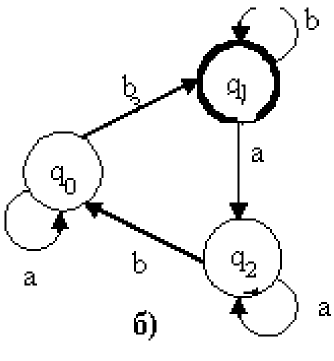
3.1. а) б)



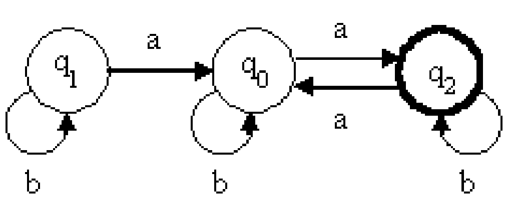


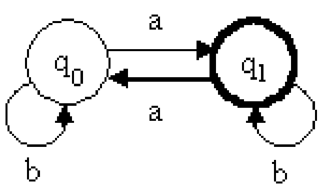
3.2. а) б)



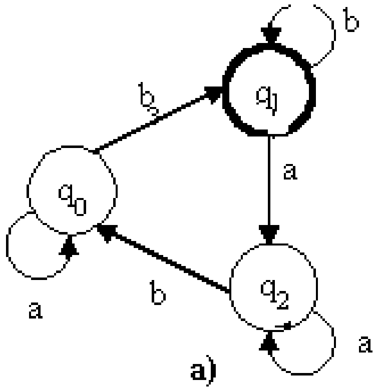


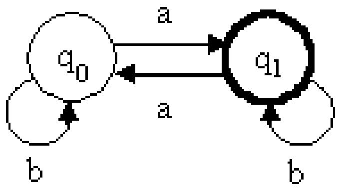
3.3. а) б)



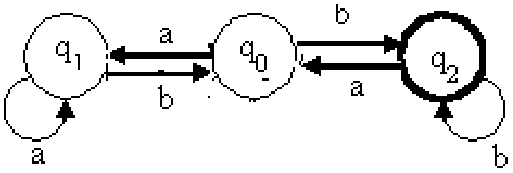


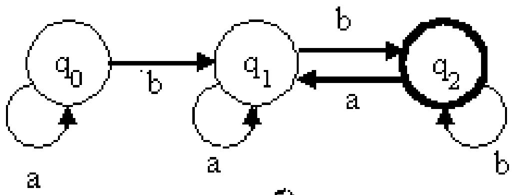
3.4. а) б)



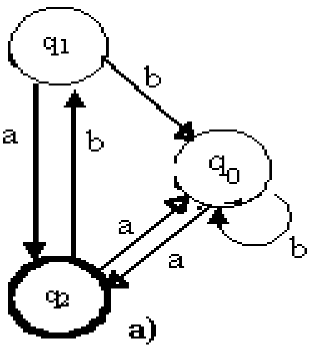


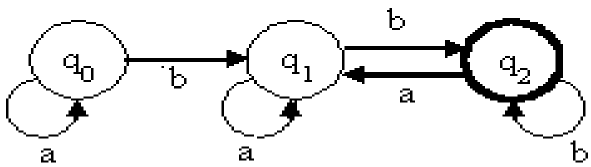
3.5. а) б)



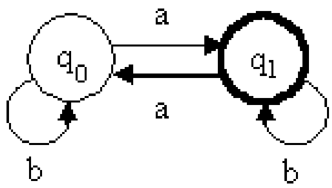


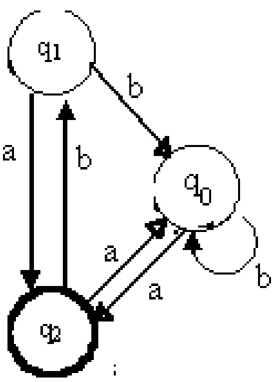
3.6. а) б)



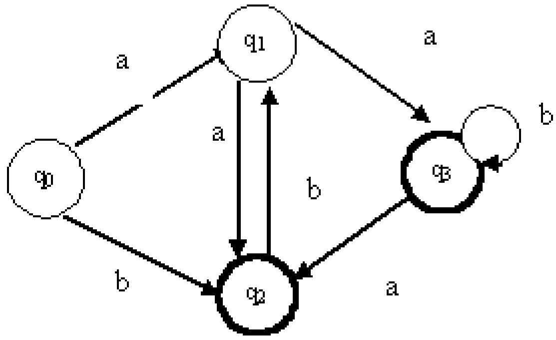


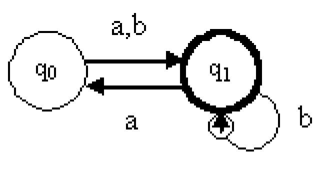
3.7 а) б)





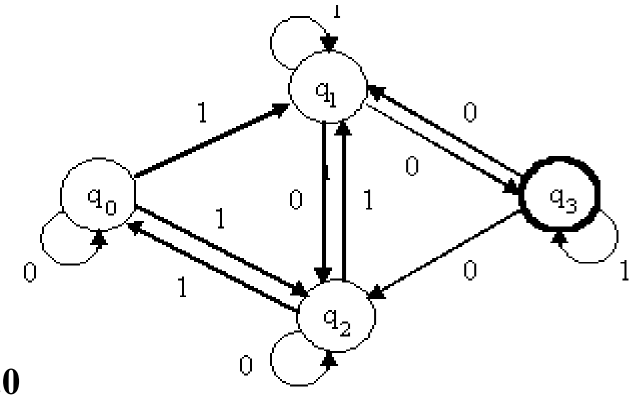
3.8. а) б)



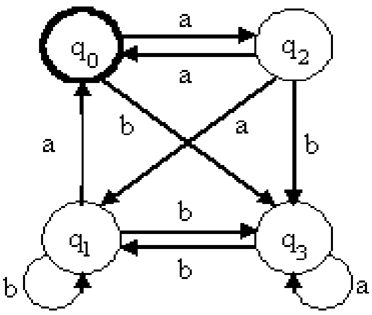


3.9 а) б)

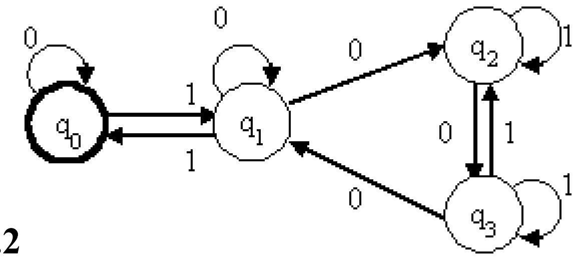
Задание 4. По недетерминированному конечному автомату, диаграмма которого представлена на рисунке построить эквивалентный ему детерминированный автомат.



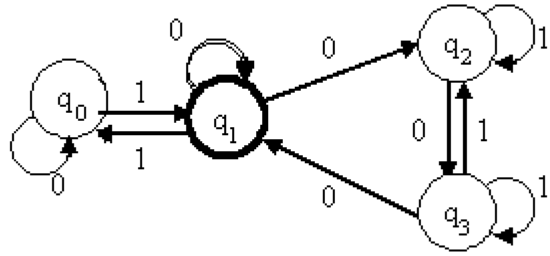
4.0



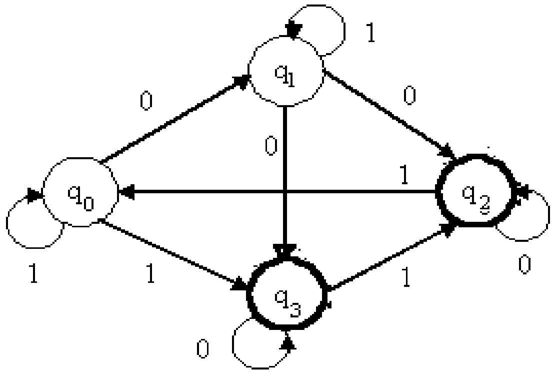
4.1



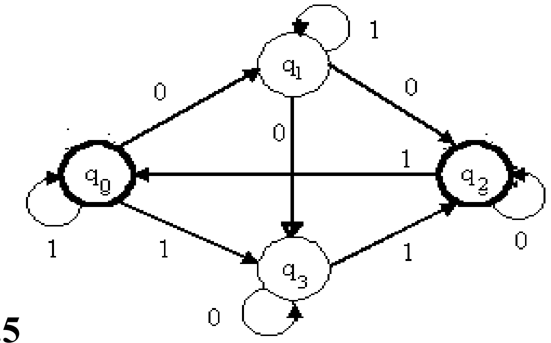
4.2



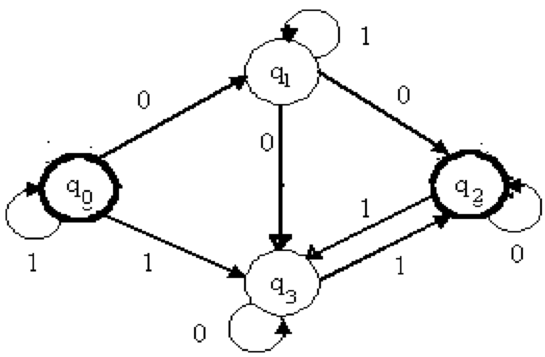
4.3



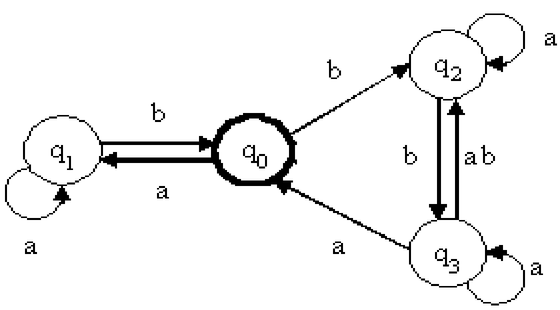
4.4



4.5

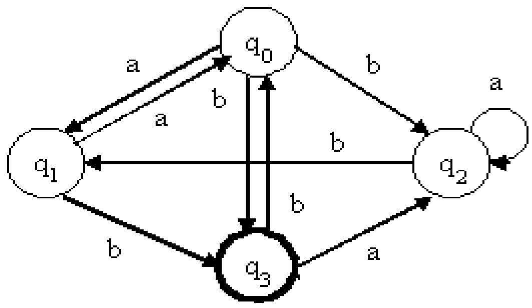


4.6

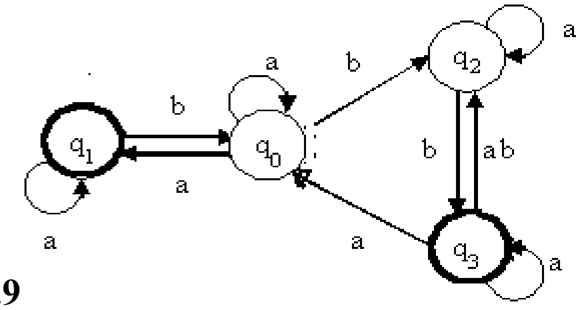


4.7

4.8

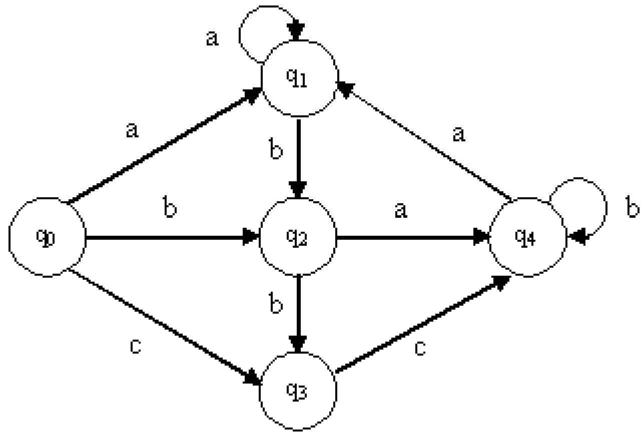


4.9



Задание 5.

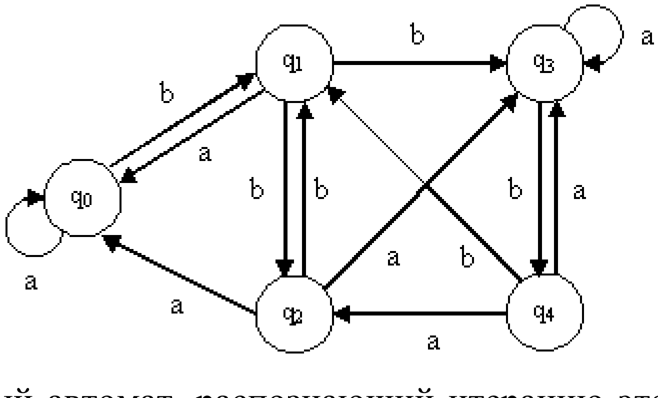
Задания 5.0-5.4 Язык *L* распознается конечным автоматом, диаграмма которого представлена на нижеприведенном рисунке.



Построить конечный автомат, распознающий итерацию этого языка, если совокупность «хороших» состояний исходного автомата следующая:

5.0 $\boldsymbol{F}=\{q_1, q_2\}$; **5.1** $\boldsymbol{F}=\{q_0, q_4\}$; **5.2** $\boldsymbol{F}=\{q_3, q_4\}$; **5.3** $\boldsymbol{F}=\{q_0, q_3\}$; **5.4** $\boldsymbol{F}=\{q_2, q_4\}$

Задания 5.0-5.4 Язык *L* распознается конечным автоматом, диаграмма которого представлена на нижеприведенном рисунке.



Построить конечный автомат, распознающий итерацию этого языка, если совокупность «хороших» состояний исходного автомата следующая:

5.5 $F=\{q_1, q_2\}$; **5.6** $F=\{q_0, q_4\}$; **5.7** $F=\{q_3, q_4\}$; **5.8** $F=\{q_0, q_3\}$; **5.9** $F=\{q_2, q_4\}$

Задание 6. Решить задачу синтеза конечного автомата $K(\Phi)$ по заданному R-выражению Φ.

6.0 $\Phi=(ab^*a^*b\cup bac^*)^*$

6.1 $\Phi=(ab^*c\cup ba)^*\cup ab^*c^*$

6.2 $\Phi=((ba^*c\cup ba)^*\cup(a^*c)^*)^*$

6.3 $\Phi=(ab^*acb\cup bac^*)^*$

6.4 $\Phi=(a^*bc\cup ab^*c)^*\cup(ac^*\cup c)^*$

6.5 $\Phi=(abc^*\cup(ab)^*(ab^*\cup ca)^*)^*$

6.6 $\Phi=(ba^*c\cup bc)^*(cb^*\cup ab)^*$

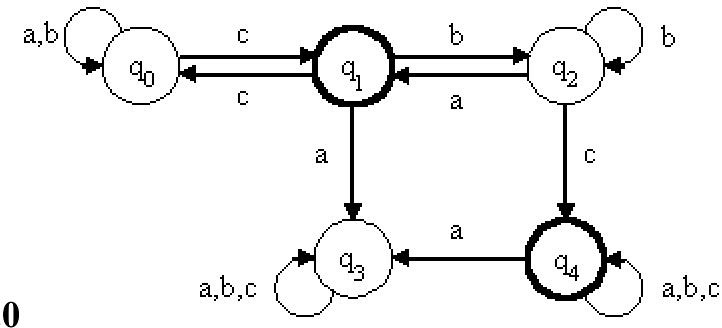
6.7 $\Phi=(((bac)^*a)^*\cup bc^*ac^*b)^*$

6.8 $\Phi=(a^*c\cup ba^*b)^*abac(a\cup(ab)^*)^*$

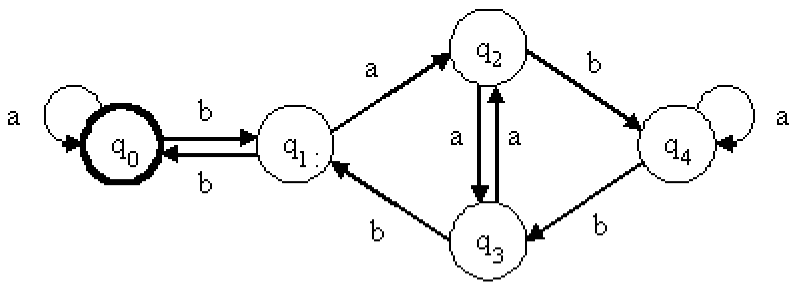
6.9. $\Phi=(ba^*c\cup b)^*(ca^*\cup ab)^*$

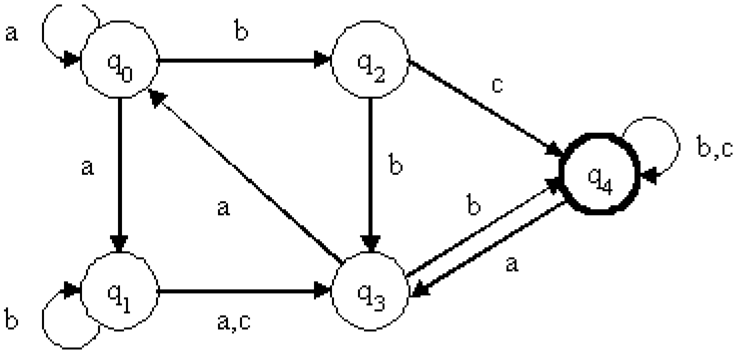
Задание 7. Решить задачу анализа конечного автомата, диаграмма переходов которого изображена на рисунке.

7.0

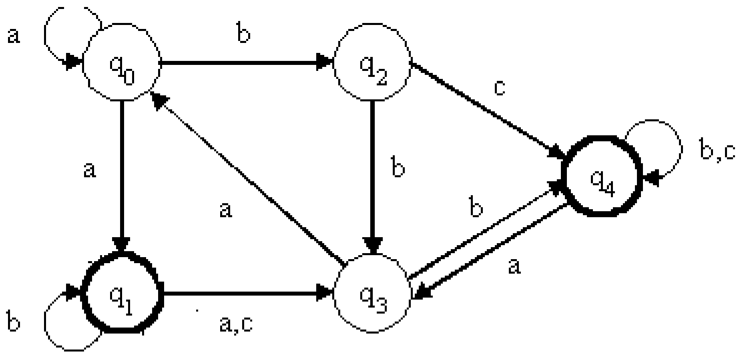


7.1

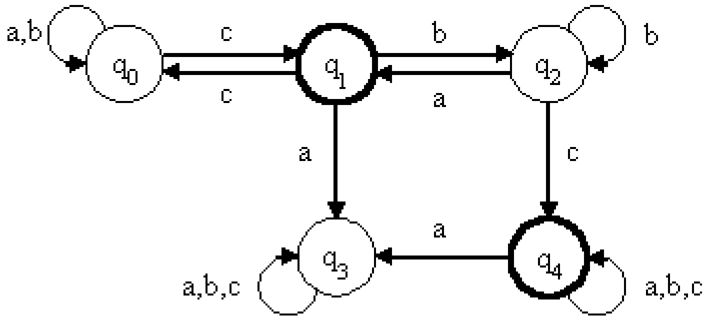



a
b
q_0
q_2
c
b,c
a
b
q_4
a
b
a
q_1
q_3
b
a,c

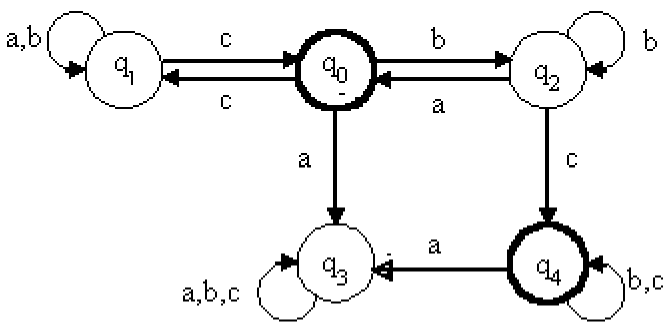
7.2


a
b
q_0
q_2
c
b,c
a
b
q_4
a
b
a
q_1
q_3
b
a,c

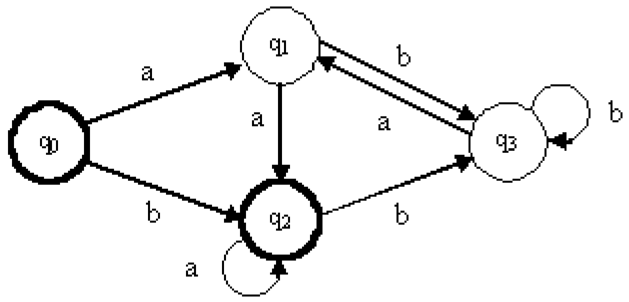
7.3


a,b
c
b
b
q_0
q_1
q_2
c
a
a
c
a
q_3
q_4
a,b,c
a,b,c

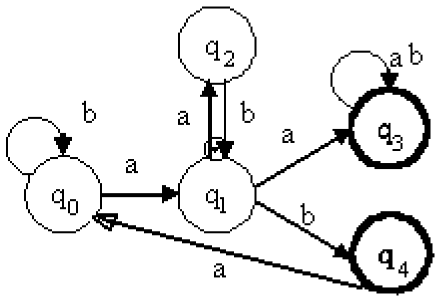
7.4


a,b
c
b
b
q_1
q_0
q_2
c
a
a
c
a
q_3
q_4
a,b,c
b,c

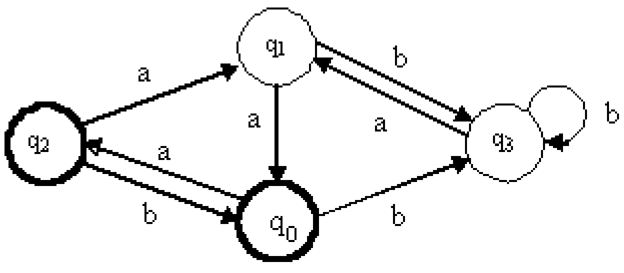
7.5


q0
q1
q2
q3
a
b
a
a
b
a
b
b

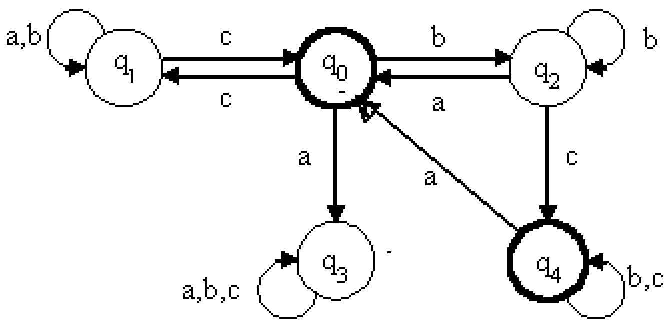
7.6


q2
q0
q1
q3
q4
b
a
a
b
a
a b
b
a

7.7


q1
q2
q0
q3
a
a
a
b
a
b
b
b

7.8


a,b
q1
c
c
q0
b
a
q2
b
a
a
c
q3
q4
a,b,c
b,c

7.9

Литература

1. Коган Д.И. Основы теории конечных автоматов и регулярных языков/ Д.И. Коган, Т.С. Бабкина. – Нижний Новгород: Издательство Нижегородского университета, 2002. – 83 с.
2. Ахо А. Теория синтаксического анализа, перевода и компиляции/ А. Ахо, Дж. Ульман. Том 1 - М.: Мир, 1978.- 611 с.
3. Брауэр В. Введение в теорию конечных автоматов/ В. Брауэр. - М.: Радио и связь, 1987.- 391 с.
4. Глушков В.М. Синтез цифровых автоматов/ В.М. Глушков. - М.: Физматгиз, 1962.- 476 с.
5. Трахтенброт Б.А. Конечные автоматы (поведение и синтез)/ Б.А. Трахтенброт, Я.М. Барздинь. - М.: Наука, 1970.- 400 с.
6. Шевелев Ю.П. Дискретная математика. Учебное пособие/ Ю.П. Шевелев. – СПб.: Издательство «Лань», 2008. – 592 с.

Оглавление

Лиогонький Марк Израилевич

Береговая Татьяна Александровна

Элементы теории конечных автоматов и регулярных языков

Методические указания и расчетно-графическая работа по курсу «Дискретная математика» для студентов ННГАСУ специальности «Информационные системы и технологии»

Подписано в печать____________ Формат 60х90 /1/16. Бумага газетная. Печать трафаретная.
Уч. изд. л.______. Усл. печ. л______. Тираж 100 экз. Заказ №_______
Государственное образовательное учреждение высшего профессионального образования
«Нижегородский государственный архитектурно-строительный университет»
603950, Н. Новгород, ул. Ильинская, 65.
Полиграфический центр ННГАСУ. 603950, Н. Новгород, ул. Ильинская, 65.