7. Автоматы распознавания языков

7.1. Языки

Алфавит $\boldsymbol{A}$ – это конечное непустое множество. *Буквы* (*символы*) – элементы алфавита $\boldsymbol{A}$. *Слово* над алфавитом $\boldsymbol{A}$ – это конечная цепочка (последовательность), состоящая из нуля или более букв из $\boldsymbol{A}$, причём одна и та же буква может входить в слово несколько раз. Последовательность, состоящая из нулевого количества букв, называется *пустым словом* и обозначается λ. Множество всех слов над алфавитом $\boldsymbol{A}$ обозначается $\boldsymbol{A}^*$, множество всех непустых слов над алфавитом $\boldsymbol{A}$ обозначается $\boldsymbol{A}^+$. Множество $\boldsymbol{A}^*$ бесконечно для любого $\boldsymbol{A}$.

Если α и β – слова над алфавитом $\boldsymbol{A}$, то результат их приписывания $\alpha\beta$ – тоже слово над $\boldsymbol{A}$ называется катенацией (конкатенацией). Катенация является ассоциативной операцией. Для любого слова α катенация с пустым словом $\alpha\lambda = \lambda\alpha = \alpha$. Если α слово, а i натуральное число, то α^i обозначает слово, полученное катенацией i слов, каждое из которых есть α. По определению α^0 – пустое слово λ.

Слово α называется *подсловом* слова β, если существуют слова α_1 и α_2 такие, что $\beta = \alpha_1\,\alpha\,\alpha_2$. Если при этом $\alpha_1 = \lambda$ (соответственно $\alpha_2 = \lambda$), то α называется *началом* или *префиксом* (соответственно *концом* или *суффиксом*) слова β.

О подмножествах множества $\boldsymbol{A}^*$ говорят как о (*формальных*) *языках* над $\boldsymbol{A}$. Конечный язык может быть, хотя бы теоретически, задан списком всех его слов. Подобная процедура для бесконечных языков невозможна. Финитный (конечный) механизм задания языка называется *грамматикой*.

7.2. Автоматы

Одним из таких механизмов является автомат. Автомат можно рассматривать как устройство, распознающее некоторый язык над входным алфавитом. В теории автоматов множество слов, т.е. язык над входным алфавитом принято называть *событием*.

Событие R_y *представимо* в инициальном автомате выходным символом y, если R_y состоит из всех тех и только тех слов во входном алфавите, образы которых при отображении, индуцируемом автоматом, оканчиваются буквой y. В автоматном графе инициального автомата событию R_y соответствуют все те слова, которые переводят автомат из начального состояния к дугам (в автомате Мили) или вершинам (в автомате Мура), помеченным выходным символом y. Событие называется *автоматным* (*представимым в автомате*), если существует автомат, в котором оно представимо. Удобно в качестве автомата, распознающего события, рассматривать разновидность автомата Мура – *автомат без выхода* (***A, S, B,*** **g**, s_0, $F \subseteq \boldsymbol{S}$) с некоторым подмножеством состояний F, которые называются *представляющими* или *финальными состояниями*. Событие R представимо в автомате без выхода, если R состоит из тех и только тех слов, которые переводят автомат из начального состояния в состояния из множества F. Если начальное состояние принадлежит множеству финальных состояний F, то событие будет включать слово нулевой длины – пустое слово λ. Пустое слово не следует путать с *пустым* (*невозможным*) событием (т.е. с пустым множеством слов). Автомат представляет пустое событие, если ни одно из его финальных состояний не достижимо из начального состояния.

Автомат представляет *систему* (*семейство*) *непустых событий* $\{R_i\}$, если каждое из событий представлено отдельными (своими) подмножествами финальных состояний $\{F_i\}$ или различающимися финальными выходными символами – индикаторами $\{y_i\}$. Пустое слово может принадлежать не более чем одному событию.

Любое конечное множество слов конечной длины представимо в автомате. В тоже время существуют бесконечные языки, не представимые в автомате. В некоторых случаях по описанию языка это свойство может быть установлено. Например, язык, состоящий из всех слов, в которых число нулей равно числу единиц, нерегулярен. Интуитивные соображения состоят в том, что автомат способен “считать” только до числа своих состояний, а поскольку число состояний автомата конечно, то он и не может распознать все слова из этого языка. Но в общем случае задача установления представимости языка алгоритмически не разрешима.

В силу сказанного следует, что надо, по возможности, пользоваться средствами спецификации, обеспечивающими предствимость языка. Перечислим некоторые из таких способов.

7.2.1. *Дефинитный (определенный)* язык может быть представлен как катенация A^*D, где A – входной алфавит, D – конечное множество слов ограниченной длины. Этот язык является бесконечным языком из слов, заканчивающихся словами из D.

Проектируя автомат, распознающий дефинитный язык, удобно сопоставить состояниям автомата все различные ***префиксы*** (***начала***) ***распознаваемых слов***. При поступлении нового символа автомат переходит в состояние, которое соответствует одному из префиксов, совпадающему с суффиксом нового слова, если таких совпадений более одного, то выбирается самое длинное. Одно из состояний должно соответствовать пустому началу. Будем его обозначать символом Λ.

Пример 7.2.-1. Спроектировать автомат, который устанавливает на выходе 1, если на двухразрядном входе автомата в последних 3-х тактах перед появлением кода 11 (в 4-м такте) появился только один раз код 00.

Введем, для удобства, следующие символы для обозначения комбинаций сигналов на входе автомата:

00 → O
11 → I
(01 или 10) → X
(I или X) → H

Автомат должен распознать в последовательности символов следующие слова: OHHI, HOHI, HHOI. Этим словам соответствуют следующие различные префиксы: Λ, O, H, OH, HO, HH, OHH, HOH, HHO. Для построения автомата Мили достаточно каждому из этих префиксов сопоставить свое состояние. Для построения автомата Мура надо добавить еще состояния, соответствующие словам: OHHI, HOHI, HHOI. Пусть имена состояний автомата совпадают с префиксами слов, им сопоставленным. Тогда автомат Мура задается следующей автоматной таблицей:

сост.	вых.	O	X	I
Λ	0	O	H	H
O	0	O	OH	OH
H	0	HO	HH	HH
OH	0	HO	OHH	OHH
HO	0	O	HOH	HOH
HH	0	HHO	HH	HH
OHH	0	HHO	HH	OHHI
HOH	0	HO	OHH	HOHI
HHO	0	O	HOH	HHOI
OHHI	1	HHO	HH	HH
HOHI	1	HHO	HH	OHHI
HHOI	1	HO	OHH	HOHI

7.2.2. *Асинхронный язык.* В асинхронный язык входят слова, в которых повторение любого из символов может быть произвольным. Точнее, если слово α принадлежит асинхронному языку, то в языке также содержатся все слова, полученные из α повторениями любых букв из α либо вычеркиванием из α некоторых повторений отдельных букв.

Функция переходов автомата, определяющего асинхронный язык, имеет следующую особенность: для любого состояния s выполняется: если s:=g(z,a), то s:=g(s,a). Автомат может перейти в другое состояние только при изменении входного символа.

Назовем *ядром* асинхронного языка язык, в котором нет повторений символов. В частности, если ядро – дефинитный язык, то при синтезе автомата, распознающего асинхронный язык с таким ядром можно воспользоваться приемом из пункта 7.2.1. – сопоставить состояниям автомата префиксы распознаваемых слов ядра.

При реализации автоматов, распознающих асинхронный язык, удобно использовать в кодах состояний коды входных символов, либо воспользоваться «условной синхронизацией» (см.п.II.4), либо использовать то и другое.

Пример 7.2.-2. Спроектировать автомат с двухразрядным входом (in1,in2) и одноразрядным выходом, который индицирует следующее событие: после нулей на обеих линиях по каждой из линий in1 и in2 прошло ровно по одному блоку единиц любой длительности.

Решение будем искать в виде симметричной композиции двух одинаковых автоматов см. п.7.1.4.

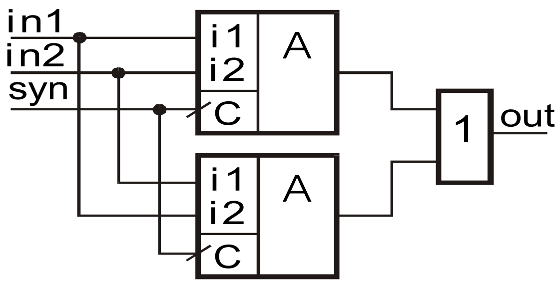

Рис.22. Композиция автоматов

Обозначим возможные символы (сочетания значений сигналов) на входах автомата следующим образом:

$$\frac{in1}{in2} \qquad \frac{0}{0} = a \qquad \frac{1}{0} = b \qquad \frac{0}{1} = q \qquad \frac{1}{1} = c$$

Если удалить все повторения символов, то получим для одного из автоматов дефинитный язык со следующими минимальными последовательностями, приводящими к событию, которое должен распознать автомат:

a c a	a b q a	a b a q a
	a c b a	a b c b a
	a b c a	a b c q a

(Другой автомат распознает последовательности, полученные из указанных заменой b на q, q на b.)

Так же как в примере 7.1.-1, состояниям автомата сопоставим все различные начала распознаваемых слов. Эти же начала и будут именами состояний (столбец 1) в автоматной таблице 1.

Таблица 7.2.-1

1	3				4	5
имя	a	b	c	q	№	код
Λ	a	Λ	Λ	Λ	1	x10
a	S	ab	ac	Λ	2	a00
ab	aba	S	abc	abq	3	b00
ac	I, a	acb	S	Λ	4	c00
aba	S	ab	ac	abaq	5	a01
acb	I, a	S	Λ	Λ	6	b01
abc	I, a	abcb	S	abcq	7	c01
abq	I, a	Λ	Λ	S	8	q00
abaq	I, a	Λ	Λ	S	8	q00
abcb	I, a	S	Λ	Λ	6	b01
abcq	I, a	Λ	Λ	S	8	q00

Таблица 7.2.-2

код	out a	b	c	q
b10	a00	S	c10	q10
c10	a00	b10	S	q10
q10	a00	b10	c10	S
a00	S	b00	c01	q10
b00	a01	S	c01	q00
c00	1, a00	b01	S	q10
a01	S	b00	c00	q00
b01	1, a00	S	c10	q10
c01	1, a00	b01	S	q00
q00	1, a00	b10	c10	S

Комментарии к автоматной таблице. Буква S означает переход в то же состояние в силу асинхронности языка. Использование такого символа позволяет упростить процесс минимизации состояний автомата – (эквивалентные состояния выглядят как явно эквивалентные). Чтобы не загромождать таблицу, выходное значение обозначено только там, где должно индицироваться искомое событие. В столбце 3 строки-состояния перенумерованы так, что эквивалентные состояния имеют один и тот же номер. В столбце 4 код состояний выбран так, что бы два разряда кода формировались кодом символа, поступившим последним. Для состояния Λ буква **x** означает любой символ кроме **a.**

Удалив эквивалентные состояния, получим автоматную таблицу 2 (состоянию Λ соответствуют первые три строки таблицы).

Реализовать автомат можно в виде схемы рис.23. Контурная часть схемы идентична выше расположенным CL—RG с иным подключением входов:

in1→i2 B1→b2
in2→i1 B2→b1

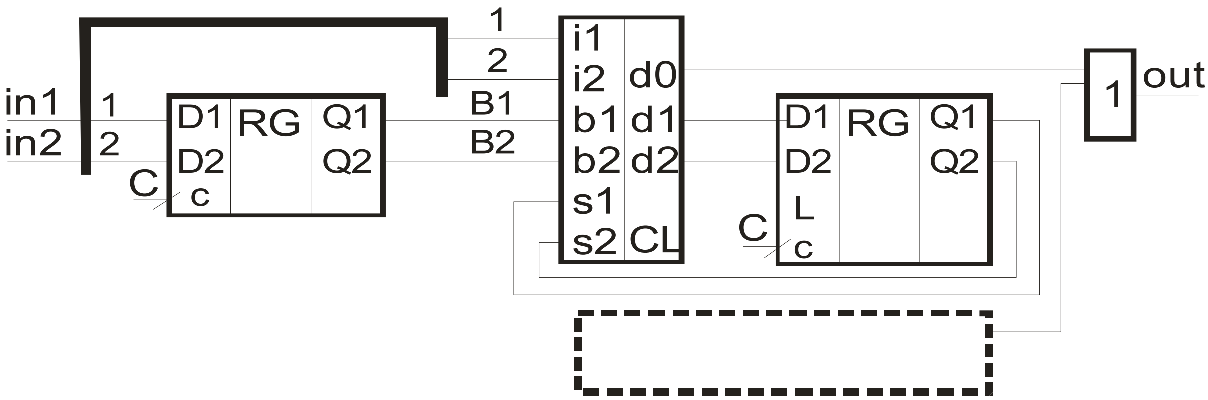

Рис.23. Реализация автомата

7.3. Источники

На графе автомата событие (язык), представляемое автоматом, изображается множеством путей из начальной вершины в финальные. Язык может быть определён с помощью произвольных ориентированных графов.

7.3.1. *Определения. Источник* - это ориентированный размеченный (конечный) граф, у которого выделены подмножества вершин называемые множествами *начальных* и *заключительные* вершин (вершина может быть одновременно и начальной и заключительной). Каждая дуга графа либо помечена символом алфавита $\boldsymbol{A}$, либо непомечена. Источник определяет язык L над алфавитом $\boldsymbol{A}$, порождаемый множеством всех путей из начальных вершин в заключительные. Каждому такому пути соответствует слово, образованное символами алфавита $\boldsymbol{A}$, являющихся пометками на дугах пути. Непомеченной (пустой) дуге соответствует пустой символ.

В источнике может быть нарушено условие однозначности автоматного графа, т.е. из одной вершины могут выходить несколько дуг, помеченных одинаково, выходить непомеченные дуги, вообще не выходить дуг (только для заключительных вершин), может быть множество начальных вершин. Источник, у которого нарушено условие однозначности, называют также *недетерминированным* автоматом.

При выполнении вычислений синхронным автоматом может существовать только один последовательный процесс, каждому шагу (такту) которого соответствует только одно состояние автомата. В источнике можно считать, что так же существует один последовательный процесс, который в любом такте может находиться более чем в одной вершине источника. Недетерминизм источника не следует смешивать со «случайностью», при которой автомат может случайно выбрать одно из следующих состояний, такой автомат называется вероятностным.

Источник всегда может быть преобразован (*детерминизирован*) в эквивалентный абстрактный детерминированный конечный автомат или просто автомат. Для этого все те вершины источника, которые в каком либо такте могли быть использованы одновременно, сопоставляются единственному состоянию автомата. Это можно сделать, если проследить все возможные варианты развития последовательных процессов в источнике.

Переход от источника к соответствующему автомату даёт увеличение числа состояний (верхняя граница 2^n, где n – число вершин источника), поэтому удобно использовать источники, поскольку для представления события эта модель требует меньшего числа состояния и с более понятной структурой.

7.3.2. *Детерминизация источника.* Состояния автомата будут находится во взаимно-однозначном соответствии с подмножествами множества вершин источника. Некоторое множество вершин π источника G назовём замкнутым при выполнении следующего условия: если $q_b \in \pi$ и существует пустая дуга, ведущая из q_b в q_e, то $q_e \in \pi$. Замыканием множества Qi назовём наименьшее замкнутое множество $\pi(Qi)$, включающее Qi .

Множество начальных вершин источника Q_H заменим его замыканием $\pi(Q_H)$ и поставим ему в соответствие начальное состояние автомата s_1 Пусть символ x переводит множество $\pi(Q_H)$ в некоторое множество $Q^x{}_2$, т.е. для каждого $q_b \in \pi(Q_H)$ существует $q_e \in Q^x{}_2$ такое, что из q_b в q_e ведёт дуга с меткой x. Замыканию $\pi(Q^x{}_2)$ поставим в соответствие состояние автомата s_2 и переход $s_2 = g(s_1, x)$. Новые состояния автомата порождаются аналогично. Множество финальных состояний автомата состоит из тех и только тех состояний, которые поставлены в соответствие замкнутым подмножествам вершин источника, имеющих непустое пересечение с подмножеством Q_K заключительных вершин источника.

Приведённая ниже процедура детерминизаии более формализована и удобна для программирования.

Процедура детерминизации источника

Рассмотрим процедуру, позволяющую преобразовать источник в автомат Мура.

***Действия*:**

0) «Чистка» источника (не обязательное действие, но сокращающее число состояний и трудоёмкость алгоритма детерминизации). Из источника удаляются 1) петли без меток; 2) вершины, недосягаемые из начальных вершин; 3) вершины, из которых недостижимы заключительные вершины, такие вершины называют тупиковыми; (удаление тупиковых вершин, если они были, сделает автомат частично определённым).

1) Нумерация дуг. Каждая помеченная дуга получает уникальный номер, начиная с 2 и далее 3,4,...

2) Таблица источника. Заготавливается таблица (пустая), в которой должны быть следующие столбцы: имена вершин, индексы вершин, остальные столбцы соответствуют всем символам входного алфавита.

3) Индексация вершин. Образуется новая единственная входная вершина, который получает индекс 1. Эта вершина соединяется дугами без меток со всеми старыми входными вершинами. Каждой вершине (за исключением вершин, из которых выходят только пустые дуги) присваиваются уникальные имена, которые помещаются в таблицу, вершинам присваивается индекс, состоящий из номеров всех входящих в эту вершину дуг. Индекс вершины, из которой выходит пустая дуга, приписывается к индексу вершины, в которую эта дуга входит. Индексы именованных вершин помещаются в таблицу.

4) В столбцы, именованные входными символами, записываются номера дуг, выходящих из вершины, соответствующей данной строке и данному входному символу.

5) Автоматная таблица. Заготавливается таблица (пустая), в которой должны быть следующие столбцы: шифр состояния, выходное значение, остальные столбцы соответствуют всем символам входного алфавита.

6) Заполнение строки автоматной таблицы. Первой строке, соответствующей начальному состоянию автомата, присваивается шифр 1. Начиная с этой строки, в столбцы, именованные входными символами, записываются номера дуг, имеющих метку, совпадающую с именем столбца и выходящих из всех вершин, в индексах которых содержится какой-либо номер из шифра этого состояния (строки). Если таких дуг нет, то записывается номер 0. В строке с шифром 0 во всех столбцах содержатся нули. (Состоянию с шифром 0 соответствует “тупиковое” состояние.)

7) Порождение новых строк. После заполнения очередной строки в столбцах, именованных входными символами, записаны шифры состояний. Если в строке появились новые шифры, не встречавшиеся ранее, то они порождают новые строки.

8) Выходные значения. Выходным значением-индикатором помечается то состояние, в шифре которого есть номер, совпадающий с номером из индекса заключительной вершины.

Пример 7.3.-1. Воспользуемся ранее рассмотренным примером. Спроектировать автомат, который устанавливает на выходе 1, если в последних 3-х тактах на двухразрядном входе автомата перед появлением кода 11 (в 4-м такте) появился только один раз код 00.

Введем следующие обозначения для сигналов на входе автомата:

00 → O
11 → I
(01,10) → X

Построим источник дефинитного языка (рис.1) и детерминизируем его.

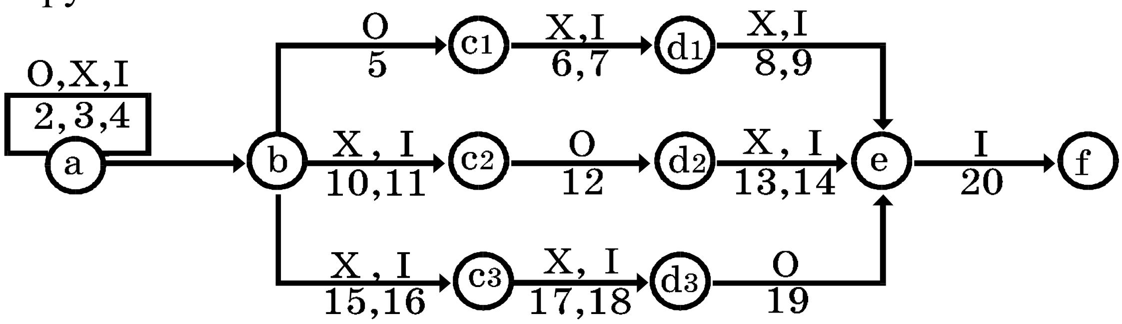

Рис.24. Источник дефинитного языка

Таблица источника:

имя	индекс	O	X	I
a	1,2,3,4	2	3	4
b	1,2,3,4	5	10,15	11,16
c1	5	--	6	7
c2	10,11	12	--	--
c3	15,16	--	17	18
d1	6,7	--	8	9
d2	12	--	13	14
d3	17,18	19	--	--
e	8,9,13,14,19	--	--	20
f	20	--	--	--

Таблица автомата:

шифр		O	X	I	
1		2,5	3,10,15	4,11,16	A
2,5		2,5	3,6,10,15	4, 7,11,16	B
3,10,15		2,5,12	3,10,15,17	4,11,16,18	C
4,11,16		2,5,12	3,10,15,17	4,11,16,18	C
3,10,15,6		2,5,12	3,8,10,15,17	4,9,11,16,18	D
4,11,16,7		2,5,12	3,8,10,15,17	4,9,11,16,18	D
2,5,12		2,5	3,6,10,13,15	4,7,11,14,16	E
3,10,15,17		2,5,12,19	3,10,15,17	4,11,16,18	F
4,11,16,18		2,5,12,19	3,10,15,17	4,11,16,18	F
3,10,15,17,8		2,5,12,19	3,10,15,17	4,11,16,18,20	G
4,11,16,18,9		2,5,12,19	3,10,15,17	4,11,16,18,20	G

3,10,15,6,13		2,5,12	3,8,10,15,17	4,9,11,16,18,20	H
4,11,16,7,14		2,5,12	3,8,10,15,17	4,9,11,16,18,20	H
2,5,12,19		2,5	3,8,10,15,17	4,7,11, 14,16,20	J
4,11,16,18,20	1	2,5,12,19	3,10,15,17	4,11,16,18	FI
4,11,16,7,14,20	1	2,5,12	3,8,10,15,17	4,7,11,14,16,20	HI
4,11,16,18,9,20	1	2,5,12,19	3,10,15,17	4,11,16,18,20	GI

Удаляя явно эквивалентные состояния, получаем автоматную таблицу, в последнем столбце которой приведены обозначения состояний из примера 7.2.-1.

Таблица 7.3.-1

	O	X	I	
A	B	C	C	Λ
B	B	D	D	O
C	E	F	F	H
D	E	G	G	OH
E	B	H	H	HO
F	J	F	F	HH
G	J	F	FI	OHH
H	E	G	GI	HOH
J	B	H	HI	HHO
FI,1	J	F	F	OHHI
HI,1	E	G	GI	HHOI
GI,1	J	F	FI	HOHI

7.4. Булевы операции над языками

Поскольку языки (события) являются множествами, для них естественно определены булевы операции дополнения, объединения, пересечения. Применение этих операций к автоматным событиям замкнуто.

7.4.1. *Дополнение*. Если R язык над алфавитом $\boldsymbol{A}$, то дополнение $\overline{R}$ является языком из множества $\boldsymbol{A}^*$ всех слов в алфавите $\boldsymbol{A}$, не принадлежащих языку R. Если R автоматное событие, то $\overline{R}$ также автоматное событие и может быть представлено автоматом с заменой всех финальных состояний на нефинальные и наоборот. Или автоматом с выходным алфавитом $B=\{0,1\}$, индикатором $I=\{1\}$ и инверсным выходом.

7.4.2. *Объединение*. Пусть R_1 и R_2 автоматные события, тогда объединение $R = R_1 \cup R_2$ также является автоматным событием. Если R_1 и R_2 представлены источниками, то источник, представляющий R, есть простое объединение этих источников такое, что множества начальных, финальных и промежуточных вершин являются объединением соответствующих вершин исходных источников с сохранением всех дуг без изменений.

Если R_1 и R_2 представлены инициальными автоматами с выходом B={0,1}, индикатором I={1}, то синхронная параллельная композиция этих автоматов с общим входом и дизъюнктивным объединением выходов (по ***ИЛИ***) будет автоматом, представляющим искомое событие.

Если автомат представляет систему событий $\{R_1, R_2\}$и каждое из событий представлено отдельными (своими) подмножествами финальных состояний $\{F_1, F_2\}$, тогда событие R будет представлено финальными состояниями $F_1 \cup F_2$.

7.4.3. *Пересечение*. Если R_1 и R_2 автоматные события, то пересечение этих множеств $R = R_1 \cap R_2$ также автоматное событие.

Если R_1 и R_2 представлены инициальными автоматами с выходом B={0,1}, индикатором I={1}, то синхронная параллельная композиция этих автоматов с общим входом и конъюнктивным объединением выходов (по ***И***) будет автоматом, представляющим искомое событие.

Если R_1 и R_2 представлены источниками G_1 и G_2 с множествами заключительных вершин Q_{K1} и Q_{K2}, то автомат, реализующий событие $R = R_1 \cap R_2$ должен иметь финальные состояния, состоящие из тех и только тех состояний, которые поставлены в соответствие замкнутым подмножествам вершин источника, являющегося простым объединением G_1 и G_2, имеющих непустое пересечение с подмножеством заключительных вершин $Q_{K1} \cap Q_{K2}$.

7.4.4. *Разность* двух множеств $R = R_1 \backslash R_2 = R_1 \cap \overline{R}_2$. Поэтому, если R_1 и R_2 автоматные события, то $R = R_1 \backslash R_2$. также автоматное событие.

7.5. Регулярные выражения

7.5.1. *Катенация* (*конкатенация, произведение*) языков L_1 и L_2 определяется как язык L_1L_2 = { αβ, где α слово из L_1, β слово из

L_2 }.

Катенация автоматных языков является автоматным языком. Пусть $L = L_1L_2$, а G_1 и G_2 соответственно источники, представляющие эти языки. Тогда источник G представляющий язык L, строится следующим образом: начальные вершины источника G совпадают с начальными вершинами источника G_1, заключительные вершины источника G совпадают с заключительными вершинами источника G_2, все заключительные вершины G_1 соединяются со всеми начальными вершинами G_2 (можно это сделать через одну новую вершину).

Катенативное замыкание (*итерация*) языка L является множество L^*, которому принадлежат все слова, являющиеся катенацией любого конечного числа слов из L, а также пустое слово. Таким образом итерация даёт возможность строить бесконечные множества из конечных.

Итерация автоматного языка L является автоматным языком L^*. Если G источник, представляющий язык L, то источник J, представляющий язык L^*, строится следующим образом: все заключительные вершины соединяются со всеми начальными вершинами дугами без меток. Добавляются одна начальная и одна заключительная вершины, которые соединяются с объединёнными вершинами дугами без меток. В языке появляется пустое слово.

7.5.2. Три операции объединение, катенация, итерация – называются *регулярными*.

Атомарным (*элементарным*) событием в алфавите $\boldsymbol{A}=\{a_1, a_2,\ldots, a_m\}$ называется одно из m+1 одноэлементного события λ, $a_1, a_2,\ldots, a_m$, где λ – пустое слово. Всякое событие, которое можно получить из атомарных применением любого конечного числа операций объединения (дизъюнкции), катенации, итерации, называется *регулярным событием*, или *регулярным множеством слов*, а всякое его задание через атомарные события и три указанные операции – *регулярным выражением*.

Например:

L_1=(a **v** b)*c(a **v** b)*c – событие (слово), содержащее символ «c» ровно 2 раза.

L_2=(a **v** b)*c(a **v** b)*c(a **v** b)* – все слова, содержащие символ «с» ровно 2 раза.

L_3=((a **v** b)*c(a **v** b)*c(a **v** b)*)* – все слова, содержащие символ «с» чётное число раз.

7.5.3. Теорема Клини утверждает, что семейства автоматных языков и регулярных множеств совпадают. Теорема Клини распадается на две теоремы теорему синтеза и теорему анализа.

То, что по регулярному выражению может быть построен автомат, представляющий регулярное множество (теорема синтеза), показано выше см. 7.4.2., 7.5.1.

Доказательство теоремы анализа «каждое автоматное событие является регулярным множеством» опускаем.

7.5.4. Разумеется, регулярные множества замкнуты относительно основных теоретико-множественных операций дополнения, объединения, пересечения. Если заданы два регулярных выражения, то для получения объединения достаточно связать эти два выражения символом дизъюнкции. В случае же необходимости получить какое-либо регулярное выражение, описывающее пересечение двух регулярных множеств или дополнение регулярного множества, такого простого алгоритма нет.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гилл А. Введение в теорию конечных автоматов. –М.: Наука, 1966 –272с.
2. Гилл А. Линейные последовательностные машины. –М.: Наука, 1974 –288с.
3. Кузин Л.Т. Основы кибернетики в двух томах. Т.2 Основы кибернетических моделей. –М.: Энергия, 1979 –583с.
4. Кудрявцев В.Б., Алешин С.В., Подколзин А.С. Введение в теорию автоматов. –М.: Наука, 1985 –320с.
5. Фридман А., Менон П. Теория и проектирование переключательных схем. – М.: Мир, 1978 –580с.