aba80b84033c9ef196388e9ea0434f90a83a40954ba270e84bcbe64f02d1d8d0

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ»
(КНИТУ-КАИ)
Чистопольский филиал «Восток»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРАКТИЧЕСКИМ РАБОТАМ

по дисциплине

ТЕОРИЯ АВТОМАТОВ

Индекс по учебному плану: **Б1.В.08**

Направление подготовки: **09.03.01 Информатика и вычислительная техника**

Квалификация: **Бакалавр**

Профиль подготовки: **Автоматизированные системы обработки информации и управления**

Типы задач профессиональной деятельности: **проектный,**
производственно-технологический

Рекомендовано УМК ЧФ КНИТУ-КАИ

Чистополь
2023 г.

Занятие 1. Минимизация переключательных функций (ПФ).

Совершенные ДНФ и КНФ используются лишь для первоначального представления ПФ. Дело в том, что эти формы записи в ПФ не удобны для построения КС, поскольку при их реализации получаются схемы, содержащие элементы, которые можно исключить, если исходить из других форм ПФ. Поэтому при синтезе комбинационных схем (КС) встает задача упрощения записи выражения для ПФ. Эта задача называется минимизацией ПФ.

В процессе минимизации СДНФ получается последовательно в начале сокращенная ДНФ, далее тупиковая и минимальная.

Существуют различные методы минимизации ПФ из которых чаще всего используются методы Квайна, Мак-Класски, Блейка-Порецкого, диаграмм Вейча, карт Карно. В принципе все эти методы являются равновидностями метода Квайна.

Метод Квайна

Метод Квайна основан на преобразовании СДНФ с помощью операций неполного склеивания и поглощения.

<u>Операция склеивания</u>: $xy \vee x\bar{y} = x$.

<u>Операция поглощения</u>: $x \vee xy = x$.

<u>Операция неполного склеивания</u>: $xy \vee x\bar{y} = x \vee xy \vee x\bar{y}$.

Теорема Квайна. Если в совершенной ДНФ ПФ провести все операции неполного склеивания, а затем все операции поглощения, то в результате получится сокращенная ДНФ, т.е. дизъюнкция всех ее простых импликант.

В соответствии с теоремой Квайна в исходном состоянии ПФ должна быть представлена в СДНФ. Если ПФ задана в произвольной ДНФ, то необходимо применять операцию развертывания. Эта операция заключается в умножении некоторых дизъюнктивных членов на выражение типа: $x \vee \bar{x} = 1$, что, естественно, не меняет значение этих членов.

Пример.

$$f(A,B,C,D) = \underset{1}{\underline{\bar{A}B\bar{C}D}} \vee \underset{2}{\underline{\bar{A}B\bar{C}\bar{D}}} \vee \underset{3}{\underline{AB\bar{C}\bar{D}}} \vee \underset{4}{\underline{A\bar{B}\bar{C}\bar{D}}} \vee \underset{5}{\underline{A\bar{B}\bar{C}D}} \vee \underset{6}{\underline{\bar{A}\bar{B}\bar{C}\bar{D}}}$$

Результат удобно записывать в следующем виде:

$1^{*} - 2^{*} = \bar{A}B\bar{C}$ (по D), $2 - 6^{*} = \bar{A}\bar{C}\bar{D}$ (по B), $4 - 5^{*} = A\bar{B}\bar{C}$,

$2 - 3^{*} = B\bar{C}\bar{D}$ (по A), $3 - 4^{*} = A\bar{C}\bar{D}$ (по B), $4 - 6 = \bar{B}\bar{C}\bar{D}$.

Звездочками помечаем дизъюнктивные члены, которые участвовали в операции склеивания и которые будут поглощены. Чтобы быстрее находить дизъюнктивные члены, которые могут склеиваться, заметим, что склеиваться могут дизъюнктивные члены, у которых число переменных с отрицанием отличается на единицу.

В дальнейшем конституенты единицы уже не будут склеиваться ни с одним из полученных членов. В результате получим:

$$f(A,B,C,D)=\underline{\bar{A}B\bar{C}}\vee\underline{B\bar{C}\bar{D}}\vee\underline{A\bar{C}\bar{D}}\vee\underline{\bar{B}\bar{C}\bar{D}}\vee\underline{A\bar{B}\bar{C}}\vee\underline{\bar{A}\bar{C}\bar{D}}$$

1 2 3 4 5 6

Можно ли еще склеивать, т.е. все ли простые импликанты?

$$2^*-4^* = \bar{C}\bar{D}\ (\text{по } B),\quad 3^*-6^* = \bar{C}\bar{D}\ (\text{по } A).$$

Тогда получим сокращенную ДНФ: $f(A,B,C,D)=\bar{C}\bar{D}\vee\bar{A}B\bar{C}\vee A\bar{B}\bar{C}$.

Давайте запишем алгоритм для получения сокращенной ДНФ. Пусть ПФ содержит nаргументов.

1. В СДНФ провести все возможные операции склеивания конституент 1. В результате образуются произведения, содержащие (n–1) букву
2. Поскольку конституенты 1 в дальнейшем не будут склеиваться ни с одним из полученных членов, проводят все операции поглощения.
3. Провести все возможные операции склеивания дизъюнктивных членов с (n–1) буквами и затем все операции поглощения членной с (n–1) буквой и т.д.

Рассмотрим еще пример. $f(A,B,C)=\bar{A}C\vee\bar{B}C\vee A\bar{B}\vee A\bar{C}\vee B\bar{C}$.

Применяя операцию развертывания, получим:

$$f(A,B,C)=\bar{A}\bar{B}C\vee\bar{A}BC\vee A\bar{B}C\vee\bar{A}\bar{B}C\vee A\bar{B}C\vee A\bar{B}\bar{C}\vee A\bar{B}\bar{C}\vee AB\bar{C}\vee\bar{A}B\bar{C}\vee AB\bar{C}=$$

$$=\underline{\bar{A}\bar{B}C}\vee\underline{\bar{A}BC}\vee\underline{A\bar{B}C}\vee\underline{A\bar{B}\bar{C}}\vee\underline{AB\bar{C}}\vee\underline{\bar{A}B\bar{C}}.$$

1 2 3 4 5 6

$$1^*-2^*=\bar{A}C\ (\text{по } B),\qquad 2-6^*=\bar{A}B\ (\text{по } C),\qquad 4-5^*=A\bar{C}\ (\text{по } B),$$

$$1-3^*=\bar{B}C\ (\text{по } A),\qquad 3-4^*=A\bar{B}\ (\text{по } C),\qquad 5-6=B\bar{C}\ (\text{по } A).$$

В результате получим:

$$f(A,B,C)=\bar{A}C\vee\bar{B}C\vee\bar{A}B\vee A\bar{B}\vee A\bar{C}\vee B\bar{C}.$$

Сравнивая выражение для $f(A,B,C)$ до операции развертывания и сокращенную ДНФ видим, дизъюнктивный член $\bar{A}B$ является **лишним.** Это объясняется тем, что сокращенная ДНФ содержит **все простые импликанты.** Ясно, что некоторые импликанты можно исключить из сокращенной ДНФ, не изменяя значений ПФ.

Методы получения тупиковых и минимальных ДНФ

Очередной этап упрощения или минимизации ПФ заключается в отыскании **тупиковых,** а затем **минимальных** форм.

Определение. Дизъюнкция простых импликант, ни одну из которых исключить нельзя, называется **тупиковой** ДНФ заданной ПФ.

Некоторые ПФ имеют несколько тупиковых форм.

Определение. Тупиковые формы ПФ, содержащие наименьшее количество букв, называются **минимальными**.

Рассмотрим 2 метода отыскания тупиковых форм.

1. **Метод испытания дизъюнктивных членов.**

Правило испытания. Для того, чтобы испытать некоторый дизъюнктивный член, необходимо исключить его из сокращенной ДНФ и подставить в оставшееся выражение такие значения переменных, которые обращают исключенный дизъюнктивный член в 1. Если при этом оставшееся выражение будет тождественно равно 1, то исключенный дизъюнктивный член является лишним.

Пример. $f(A,B,C)=\overline{A}C\vee\overline{B}C\vee\overline{A}B\vee A\overline{B}\vee A\overline{C}\vee B\overline{C}$

Испытываем $\overline{A}C$=1, A=0, C=1. $f(A,B,C)=\overline{B}\vee B$=1. $\overline{A}C$ – лишний.

Его исключаем из выражения для ПФ и затем вновь проводим испытание дизъюнктивных членов ит.д.

$f(A,B,C)=\overline{B}C\vee\overline{A}B\vee A\overline{B}\vee A\overline{C}\vee B\overline{C}$.

$\overline{B}C$ = 1, B = 0, C = 1. $f(A,B,C)=0\vee A$ = A, этот член исключить нельзя.

$\overline{A}B$ = 1, A= 0, B = 1. $f(A,B,C)=0\vee\overline{C}=\overline{C}$, этот член исключить нельзя.

$A\overline{B}$ = 1, A = 1, B = 0. $f(A,B,C)=C\vee\overline{C}=1$, этот член можно исключить.

$A\overline{C}$ = 1, A = 1, C = 0. $f(A,B,C)=B\vee\overline{B}=1$, этот член можно исключить.

$B\overline{C}$ = 1, B = 1, C = 0. $f(A,B,C)=\overline{B}\vee B=1$, этот член можно исключить.

Итак, $\overline{A}C$ исключили из выражения для ПФ. Далее испытаем $B\overline{C}$.

$f(A,B,C)=\overline{B}C\vee\overline{A}B\vee A\overline{B}\vee A\overline{C}$.

$\overline{B}C$ = 1, B = 0, C = 1. $f(A,B,C)=0\vee A$ = A, этот член исключить нельзя.

$\overline{A}B$ = 1, A= 0, B = 1. $f(A,B,C)=0\vee 0$= 0, этот член исключить нельзя.

$A\overline{B}$ = 1, A = 1, B = 0. $f(A,B,C)=C\vee\overline{C}=1$, этот член можно исключить.

$A\overline{C}$ = 1, A = 1, C = 0. $f(A,B,C)=0\vee\overline{B}=\overline{B}$, этот член исключить нельзя.

Получим: $f_1(A,B,C)=\overline{B}C\vee\overline{A}B\vee A\overline{C}$. Испытывая вновь дизъюнктивные члены, увидим, что ни один из этих членов исключить нельзя. В результате нашли одну из тупиковых ДНФ.

Если начать испытания дизъюнктивных членов с члена $\overline{B}C$, то получим:

$$f_2(A,B,C)=\overline{A}C\vee A\overline{B}\vee B\overline{C}.$$

В общем случае это громоздкий способ. Более удобно применять графический метод.

Метод импликантных матриц

Для построения импликантной матрицы строят таблицу, в горизонтальные и вертикальные входы которой записывают все простые импликанты и все конституенты 1 соответственно. В таблице отмечают крестиками те конституенты 1, которые поглощаются соответствующими импликантами.

Правило. Чтобы получить минимальную ДНФ заданной ПФ, достаточно найти минимальное число импликант, которые совместно накрывают крестиками все колонки импликантной матрицы, т.е. все конституенты 1.

В результате получим:

f_1 (A, B, C) = $\overline{A}C\vee A\overline{B}\vee B\overline{C}$,

f_2 (A, B, C) = $\overline{B}C\vee\overline{A}B\vee A\overline{C}$,

f_3 (A, B, C) = $\overline{A}C\vee\overline{B}C\vee A\overline{C}\vee B\overline{C}$. Как видим, ПФ f_3 (A, B, C) тупиковая, но не минимальная.

Пр. импл. \ Конст. "1"	$\overline{A}\,\overline{B}C$	$\overline{A}BC$	$A\overline{B}C$	$A\overline{B}\,\overline{C}$	$AB\overline{C}$	$\overline{A}B\overline{C}$
	1	2	3	4	5	6
a) $\overline{A}C$	x	x				
в) $\overline{B}C$	x		x			
c) $\overline{A}B$		x				x
d) $A\overline{B}$			x	x		
e) $A\overline{C}$				x	x	
f) $B\overline{C}$					x	x

Выбор минимального покрытия по методу Петрика

Из импликантной матрицы может следовать достаточно большое число вариантов покрытий, что при большой импликантной матрице затрудняет задачу выбора минимального покрытия.

Петрик предложил формальный метод выбора минимального покрытия, основанный на получении булевого выражения – конъюнкции дизъюнктивных членов. Раскрывая скобки, определяем все множество возможных покрытий и из них определяем минимальные.

Пример. $f(a,b,c,d,e,f) = (a \vee b)(a \vee c)(b \vee d)(d \vee e)(e \vee f)(c \vee f) =$

$= (a \vee ab \vee ac \vee bc)(d \vee bd \vee ed \vee be)(ec \vee cf \vee ef \vee f) = (a \vee bc)(d \vee be)(f \vee ec) =$

$= adf \vee bcdf \vee abef \vee bcef \vee acde \vee bcde \vee abce \vee bce$.

В результате получили все множество покрытий.

f_1 (A, B, C) = $\overline{A}C \vee A\overline{B} \vee B\overline{C}$, f_2 (A, B, C) = $\overline{B}C \vee \overline{A}B \vee A\overline{C}$.

Остальные формы тупиковые, но не минимальные.

Метод минимизации Блейка-Порецкого.

Недостатком метода Квайна является то, что для его применения необходимо представить функцию в СДНФ. Процесс такого представления для функции с большим числом аргументов зачастую является весьма громоздкой задачей, т.е. было бы желательно найти возможность построения сокращенной ДНФ, не по СДНФ, а по произвольной ДНФ данной функции. Идея построения сокращенной ДНФ по произвольным ДНФ была предложена в работах А. Блейка и П.С. Порецкого. Суть метода состоит в том, что в произвольной ДНФ осуществляют все операции **обобщенного склеивания:**

Если в ДНФ для данной функции $f(A, B, C)$ входят две конъюнкции вида AC и $B\overline{C}$, то имеет место следующее равенство:

$$AC \vee B\overline{C} = AC \vee B\overline{C} \vee AB.$$

Запишем функцию $f(A,B,C)$в следующем виде:

$$AC \vee B\overline{C} = AC \vee ABC \vee B\overline{C} \vee AB\overline{C} = AC \vee B\overline{C} \vee AB.$$

Из этого вытекает метод построения сокращенной ДНФ. Этот метод заключается в том, если в ПФ есть конъюнкции с переменными C и $\overline{C}$,то не изменяя исходную функцию необходимо пополнить новыми членами вида АВ. После этого надо провести поглощения и вновь повторить пополнение ДНФ. Этот процесс необходимо производить до тех пор, пока не будут возникать новые конъюнкции вида АВ. По окончании преобразований получаем сокращенную ДНФ.

В основе метода Квайна так же лежит операция склеивания, но операция неполного склеивания, является частным случаем операции обобщенного склеивания при А = В.

$$AC \vee A\overline{C} = A \vee AC \vee A\overline{C}.$$

Рассмотрим примеры:

1. Найти сокращенную ДНФ для функции трех аргументов:

$f(x,y,z)=xy\bar{z} \vee xz \vee \bar{x}y=xy\bar{z} \vee xz \vee \bar{x}y \vee xy \vee y\bar{z} \vee yz=$

$=xz \vee \bar{x}y \vee xy \vee y\bar{z} \vee yz \vee y=xz \vee y$

2. Пусть дана функция трех аргументов:

$f(x1,x2,x3)=x1\overline{x2}x3 \vee \overline{x1}x2x3 \vee \overline{x1}\,\overline{x2}\,\overline{x3} \vee x2\overline{x3} \vee x1\overline{x3}=$

$=x1\overline{x2}x3 \vee \overline{x1}x2x3 \vee \overline{x1}\,\overline{x2}\,\overline{x3} \vee x2\overline{x3} \vee x1\overline{x3} \vee x1\overline{x2} \vee$

$\vee \overline{x1}x2 \vee \overline{x1}\,\overline{x3} \vee \overline{x2}\,\overline{x3}= x2\overline{x3} \vee x1\overline{x3} \vee x1\overline{x2} \vee$

$\vee \overline{x1}x2 \vee \overline{x1}\,\overline{x3} \vee \overline{x2}\,\overline{x3} \vee \overline{x3}=x1\overline{x2} \vee \overline{x1}x2 \vee \overline{x3}$

Метод минимизации Мак-Класки

В методе Квайна есть одно существенное неудобство. Оно связано с необходимостью полного попарного сравнения всех членов СДНФ ПФ на этапе нахождения всех простых импликант (при выполнении всевозможных операций неполного склеивания). При большом числе переменных применение метода Квайна становится затруднительным. Мак-Класки в 1956г. предложил модернизацию метода Квайна, дающую уменьшение числа сравнений.

Идея метода Мак-Класки заключается в следующем. Конституенты 1 в СДНФ представляются не в буквенном виде, а виде условных чисел – номеров соответствующих конституент.

Пример.

$$f(A,B,C,D) = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} \vee \overline{A}\,\overline{B}\,\overline{C}D \vee \overline{A}\,\overline{B}C\overline{D} \vee \overline{A}BCD \vee A\overline{B}\,\overline{C}D \vee A\overline{B}C\overline{D} \vee ABC\overline{D} \vee ABCD =$$

$$= 0000 \vee 0001 \vee 0010 \vee 0111 \vee 1001 \vee 1010 \vee 1110 \vee 1111 =$$

$$= 0 \vee 1 \vee 2 \vee 7 \vee 9 \vee 10 \vee 14 \vee 15 = \vee\ (0, 1, 2, 7, 9, 10, 14, 15).$$

При этом оказывается, что переменные имеют вес: ABCD $\rightarrow$ 8421.

Давайте посмотрим, какие конституенты 1 склеиваются. Мы говорили, что могут склеиваться те, у которых число отрицаний отличается на 1.

$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D} \vee \overline{A}\,\overline{B}\,\overline{C}D =$ 0000$\vee$0001 = 0$\vee$1= $\overline{A}\,\overline{B}\,\overline{C}$; 0$\vee$1= (0,1), (1), т.е. склеиваются конституенты 0 и 1, а в скобках указывается разность, какая переменная исключается (А).

Еще пример: 0100$\vee$0110 = 4$\vee$6 = (4, 6), (2) – склеиваются по В.

Вводится понятие **индекса.** Индексом целого числа называется количество 1 в двоичном представлении этого числа; 7 = 0111 – индекс равен 3, 0000 – индекс равен 0 и т.д. Если конституенты 1 склеиваются, то их индексы на 1, в скобках должно быть число (их разности), равное целой степени 2.

Правило. Для того чтобы два числа *m* и *n* являлись номерами двух склеивающихся между собой конституент 1, необходимо и достаточно,

чтобы их индексы отличались точно на 1, чтобы сами числа отличались друг от друга на степень 2 и чтобы число с большим индексом было больше числа с меньшим индексом.
Так конституенты 1 с номерами 4 и 3 = 0100 и 0011 не склеиваются, хотя их разности равны 1.

Все конституенты 1 разбиваются на группы в соответствии с индексом и располагаются в порядке их возрастания. Группы отделяются друг от друга чертой. Склеиваться могут конституенты только соседних групп.
Пример. $f(A,B,C,D) = \vee$ (0, 1, 3, 4, 5, 6, 10, 11, 12, 14, 15)
Прежде всего группируем номера конституент в порядке роста их индексов. В нулевую группу попадает номер 0. В 1 группу – конституенты с индексом 1 (1 и 4), 2 (3=0011, 5=0101, 6=0110, 10-1010, и 12=1100), 3 (11=1011 и 14=1110), 4 (15=1111). В результате имеем следующее разбиение конституент функции $f(A,B,C,D)$ на группы:

0	0
1	1 4
2	3 5 6 10 12
	11 14
4	15

В силу теоремы, склеивание возможно между номерами конституент соседних групп, т.е. 0-1, 1-2, 2-3 и 3-4. Никакие другие склеивания невозможны.

Необходимо также, чтобы число, стоящее в следующей группе, было больше соответствующего числа в предыдущей группе, причем больше на целую степень 2.

Результаты склеивания записываются парами, рядом записываются разности номеров склеивающихся конституент. Например, 0 склеивается с 1, записывается пара (0,1) и рядом их разность: 1– 0 = 1. Запись выглядит так: (0, 1), (1). Около склеенных номеров ставятся звездочки, показывающие, что эти члены участвовали хотя бы в одном склеивании. Такие члены, как известно, в сокращенную ДНФ уже не войдут. После завершения склеивания групп отделяем горизонтальной чертой полученные выражения. Дальше осуществляется та же процедура, но теперь склеиваются между собой полученные пары. Под парой подразумевается трехбуквенное выражение. Для склеивания пар применяются те же правила, но добавляется еще одно условие: разности у склеиваемых пар должны быть одинаковы. При этом оказывается, что переменные в выражении одинаковы.

0	0	*
1	1	*
	4	*
2	3	*
	5	*
	6	*
	10	*
	12	*
3	11	*
	14	*
4	15	*

(0,1), (1) *
(0,4), (4) *

(1,3), (2) (a)
(1,5), (4) *
(4,5), (1) *
(4,6) , (2) *
(4,12), (8) *

(3,11), (8) (в)
(6,14), (8) *
(10,11), (1) *
(10,14), (4) *
(12,14), (2) *

(11,15), (4) *
(14,15), (1) *

(0,1,4,5), (1,4) (c)
~~(0,1,4,5), (4,1)~~

(4,6,12,14), (2,8) (d)
~~(4,6,12,14), (8,2)~~

(10,11,14,15) (1,4) (e)
~~(10,11,14,15) (4,1)~~

Возьмем, например, пары (0, 1),(1) и (1, 3), (2). 0 и 1 склеиваются, 1 и 3 тоже, но разности у них не одинаковы – поэтому склеивание невозможно (у первой пары нет переменной D, у второй пары нет переменной С – они, конечно, и не могут склеиваться).

В результате склеивания пишем уже четверки (0, 1, 4, 5) и две разности (1, 4), т.е. в результирующем выражении нет 2-х переменных В и D.

После завершения этих склеиваний получают, если возможно, восьмерки, при этом должны совпадать уже разности в скобках у четверок и т.д.

Обозначим пары и четверки, которые не участвовали в склеивании, латинскими буквами. Тогда $f(A,B,C,D) = a \vee b \vee c \vee d \vee e$. При получении выражения для импликант *а*, *в*, *с* и т.д., поступают следующим образом: берут любую из конституент 1, участвующих в операции склеивания, и исключают переменные, указанные в разностях.

Например, для импликанты *a* получим: 0001=$\bar{A}\bar{B}\bar{C}D$. Разность 2 говорит о том, из выражения необходимо исключить переменную с весом 2, т.е. С. Тогда $a = \bar{A}\bar{B}D$.

Аналогично получим:

для *в*: 0011=$\bar{A}\bar{B}CD$, $в = \bar{B}CD$, для *с*: 0001=$\bar{A}\bar{B}\bar{C}D$, $c = \bar{A}\bar{C}$, для *d*: 0100 = =$\bar{A}B\bar{C}\bar{D}$, $d = B\bar{D}$, для *e*: 1111=AC, $e = AC$. В результате получим:

$$f(A,B,C,D) = \bar{A}\bar{B}D \vee \bar{B}CD \vee \bar{A}\bar{C} \vee B\bar{D} \vee AC.$$

Занятие 2. Диаграммы Карно-Вейча для полностью определённых ПФ. Минимизация не полностью определённых ПФ. Построение комбинационных схем.

Диаграмма Вейча является фактически таблицей истинности ПФ, которая представляется не в виде столбцов, а в виде специальных карт. Каждой клетке диаграммы соответствует определенный набор значений аргументов. Поэтому диаграммы можно рассматривать как графическое представление совокупности всех конституент единицы. При этом диаграмма строится таким образом, что склеивающиеся между собой конституенты единицы оказываются расположенными в соседних клетках, т.к. отличаются значением только одной переменной. Приведем примеры построения диаграммы Вейча и карт Карно для ПФ от разного числа аргументов.

1) переключательные функции двух аргументов:

N	A	B	f
0	0	0	1
1	0	1	1
2	1	0	1
3	1	1	0

A \ B	0	1
0	1	1
1	1	0

A \ B	$\bar{B}$	B
$\bar{A}$	$\bar{A}\bar{B}$	$\bar{A}B$
A	$A\bar{B}$	AB

$$f(A,B)=\bar{A}\,\bar{B} \vee \bar{A}B \vee A\bar{B}=\bar{A}(\bar{B}\vee B)\vee\bar{B}(\bar{A}\vee A)=\bar{A}\vee\bar{B}$$

Диаграмма Вейча:

A \ B	$\bar{B}$	B
$\bar{A}$	1	1
A	1	0

Для представления функции с помощью диаграмм Вейча или карт Карно следует записать единицы в клетках таблицы, соответствующие наборам, на которых функция равна единице, а нули в остальные клетки. При склеивании соседние конституенты будем обводить овалом.

2) переключательные функции трех переменных.

N	A	B	C	f
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

A \ BC	00	01	11	10
0	1	0	1	1
1	1	1	0	1

$$f(A,B,C)=\bar{A}B \vee A\bar{B} \vee \bar{C}$$

В этой карте соседними клетками являются также крайние клетки каждой строки, т.к. расположенные в них конституенты отличаются друг от друга значением только одной переменной. Такую диаграмму следует рассматривать, как свернутую в цилиндр.

Минимизация ПФ с помощью диаграмм Вейча и карт Карно сводится к такому объединению соседних единиц в группы, при котором каждая группа содержит максимальное число единиц, а количество таких групп минимально. При этом все единицы диаграммы Вейча данной функции накрываются наименьшим числом наиболее коротких произведений (импликант).

3) **переключательные функции четырех переменных**.

N	A	B	C	D	f1	f2
0	0	0	0	0	1	1
1	0	0	0	1	1	1
2	0	0	1	0	0	0
3	0	0	1	1	0	0
4	0	1	0	0	1	0
5	0	1	0	1	1	0
6	0	1	1	0	0	1
7	0	1	1	1	0	1
8	1	0	0	0	0	1
9	1	0	0	1	1	1
10	1	0	1	0	1	1
11	1	0	1	1	1	1
12	1	1	0	0	0	0
13	1	1	0	1	1	0
14	1	1	1	0	1	0
15	1	1	1	1	1	0

AB \ CD	00	01	11	10
00	1	1	0	0
01	1	1	0	0
11	0	1	1	1
10	0	1	1	1

$f1=\bar{A}\bar{C} \vee \bar{C}D \vee AC$ - минимальная ДНФ

Получим минимальную КНФ:

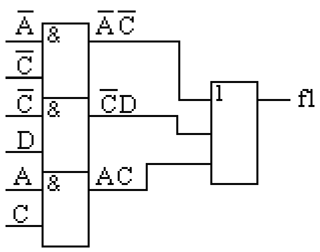
$\overline{f1}=\bar{A}C \vee A\bar{C}\bar{D}$ - минимальная ДНФ функции f1 с отрицанием.

С помощью формулы Де-Моргана перейдем к функции f1 без отрицания:

$f1=\overline{\overline{f1}}=\overline{\bar{A}C \vee A\bar{C}\bar{D}}=\overline{\bar{A}C} \,\&\, \overline{A\bar{C}\bar{D}}=(A \vee \bar{C})\&(\bar{A} \vee C \vee D)$ - минимальная КНФ функции f1

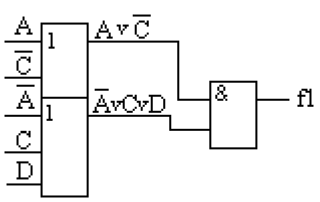
4) нарисуем схему в булевом базисе:

ДНФ



$f1=\bar{A}\bar{C} \vee \bar{C}D \vee AC$

КНФ



$f1=(A \vee \bar{C})\&(\bar{A} \vee C \vee D)$

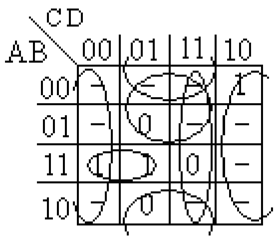
Построим схему реализующую функции f1 в базисе И-НЕ (штрих Шеффера). Для этого осуществим переход от булевого базиса к базису И-НЕ.

Минимизация не полностью определённых ПФ.

В ЭВМ часто применяются комбинационные схемы (КС), закон функционирования которых определен не полностью. В таких схемах некоторые комбинации сигналов на входы никогда не подаются. Эти комбинации входных сигналов называются **запрещенными.** Для запрещенных входных комбинаций выходные сигналы не определены, т.е. могут принимать любые значения (0 или 1). Поэтому при синтезе КС с не полностью заданным законом функционирования можно произвольно задать значение выходных сигналов для запрещенной комбинации входных сигналов, поскольку нормальная работа схемы при этом не нарушается. Обычно выходным сигналам на запрещенных комбинациях придают такие значения, при которых можно построить наиболее простую схему. Работа схем с запрещенными комбинациями входных сигналов описывается не полностью определенными ПФ, т.е. функциями, значения которых определены не на всех наборах аргументов. Поэтому минимизация не полностью определенных ПФ с помощью карт Карно сводится к такому доопределению ПФ, при котором получаются группы с максимальным числом соседних единиц в каждой группе, а число таких групп минимально. При этом ПФ будет содержать минимум букв.

Пример:

Найти минимальную ДНФ и минимальную КНФ не полностью определенной ПФ от четырех переменных. Функция равна единице на наборах: f=v(2,12,13) и равна нулю на наборах: f=v(5,9,15). На остальных наборах функция не определена.



$f=\bar{D} \vee AB\bar{C}$ - минимальная ДНФ.

$\bar{f}=\bar{A}D \vee CD \vee \bar{B}D$ - минимальная ДНФ с функцией $\bar{f}$

$f=\bar{\bar{f}}=\overline{\bar{A}D \vee CD \vee \bar{B}D}=(A \vee \bar{D})\&(\bar{C} \vee \bar{D})\&(B \vee \bar{D})$ - минимальная КНФ.

Занятие 3. Конечные автоматы. Способы их задания.
Элементарные автоматы.

Для задания конечного автомата **S** необходимо задавать совокупность из пяти объектов: **S(A, X, Y, δ, λ)**,

где **A** = {$a_0, a_1, a_2, \ldots, a_n$} – множество внутренних состояний автомата,

$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ – множество входных сигналов (входной алфавит), $\mathbf{x}_i$ буква входного алфавита,

$\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$ – множество выходных сигналов (выходной алфавит),

δ - функция переходов, определяющая состояние автомата $\mathbf{a}(\mathbf{t}+1)$, в котором автомат будет находится в момент времени $(\mathbf{t}+1)$, в зависимости от состояния автомата $\mathbf{a}(\mathbf{t})$ и входного сигнала $\mathbf{x}(\mathbf{t})$ в момент времени $\mathbf{t}$, т.е. $\mathbf{a}(\mathbf{t}+1) = \delta\ [\mathbf{a}(\mathbf{t}), \mathbf{x}(\mathbf{t})]$,

λ - функция выходов, определяющая значение выходного сигнала $\mathbf{y}(\mathbf{t})$ в зависимости от состояния автомата $\mathbf{a}(\mathbf{t})$ и входного сигнала $\mathbf{x}(\mathbf{t})$ в момент времени $\mathbf{t}$, т.е. $\mathbf{y}(\mathbf{t}) = \lambda[\mathbf{a}(\mathbf{t}), \mathbf{x}(\mathbf{t})]$.

Автомат работает следующим образом, в каждый момент времени $\mathbf{t}$ он находится в определенном состоянии $\mathbf{a}(\mathbf{t})$ из множества $\mathbf{A}$ возможных состояний. Причем в начальный момент времени $\mathbf{t} = \mathbf{0}$, он всегда находится в состоянии $\mathbf{a}(\mathbf{t} = 0) = \mathbf{a}_0$. В момент времени $\mathbf{t}$ автомат воспринимает входной сигнал $\mathbf{x}(\mathbf{t})$, выдает выходной сигналу $\mathbf{y}(\mathbf{t}) = \lambda[\mathbf{a}(\mathbf{t}), \mathbf{x}(\mathbf{t})]$ и переходит в следующее состояние $\mathbf{a}(\mathbf{t}+1)=\delta[\mathbf{a}(\mathbf{t}), \mathbf{x}(\mathbf{t})]$. Другими словами абстрактный автомат каждой паре символов $\mathbf{a}(\mathbf{t})$ и $\mathbf{x}(\mathbf{t})$ ставит в однозначное соответствие пару символов $\mathbf{a}(\mathbf{t}+1)$ и $\mathbf{y}(t)$. Такие автоматы называют ***детерминированными***. На преобразование информации в детерминированных автоматах наложены условия.

Кроме детерминированных автоматов существуют **вероятностные** или **стохастические автоматы**, в которых переход из одного состояния в другое под воздействием случайных или детерминированных входных сигналов происходит случайно. Работа таких автоматов описывается уже матрицей переходов δ, элементами которой являются вероятности переходов из одного состояния в другое.

Задание автомата

Чтобы задать конечный автомат $\mathbf{S}$ необходимо описать все элементы множества $\mathbf{S} = \{\mathbf{A}, \mathbf{X}, \mathbf{Y}, \delta, \lambda\}$. То есть необходимо описать входной, выходной алфавиты и алфавит состояний, а также функции переходов δ и выходов λ. При этом среди множества $\mathbf{A} = \{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_n\}$ необходимо выделить начальное состояния $\mathbf{a}_0$, в котором автомат находится в момент времени $\mathbf{t} = 0$. Существует несколько способов задания работы автомата, но наиболее часто используются табличный и графический.

Табличный способ

При табличном способе задания автомат Мили описывается двумя таблицами: таблицей переходов и таблицей выходов.

Таблица переходов

$x_j \backslash a_i$	a_0	…	a_n
x_1	$\delta(a_0,x_1)$	…	$\delta(a_n,x_1)$
…	…	…	…
x_m	$\delta(a_0,x_m)$	…	$\delta(a_n,x_m)$

Таблица выходов

$x_j \backslash a_i$	a_0	…	a_n
x_1	$\lambda(a_0,x_1)$	…	$\lambda(a_n, x_1)$
…	…	…	…
x_m	$\lambda(a_0,x_m)$	…	$\lambda(a_n,x_m)$

Строки этих таблиц соответствуют входным сигналам $\mathbf{x}(\mathbf{t})$, а столбцы – состояниям. На пересечении столбца a_i и строки $\mathbf{x}_j$ в таблице переходов ставится состояние $\mathbf{a}_s = \boldsymbol{\delta}[\ \mathbf{a}_i,\ \mathbf{x}_j]$, в которые автомат перейдет из состояния $\mathbf{a}_i$ под воздействием сигнала $\mathbf{x}_j$; а в таблице выходов – соответствующий этому переходу выходной сигнал $\mathbf{y}_g = \boldsymbol{\lambda}[\ \mathbf{a}_i,\mathbf{x}_j]$. Иногда автомат Мили задают совмещенной таблицей переходов и выходов, она в некоторых случаях более удобна.

Совмещенная таблица переходов и выходов автомата Мили.

$x_j \backslash a_i$	a_0	…	a_n
x_1	$\delta(a_0,x_1)$/ $\lambda(a_0,x_1)$	…	$\delta(a_n,x_1)$/ $\lambda(a_n,x_1)$
…	…	…	…
x_m	$\delta(a_0,x_m)$/ $\lambda(a_0,x_m)$	…	$\delta(a_n,x_m)$/ $\lambda(a_n,x_m)$

Задание таблиц переходов и выходов полностью описывает работу конечного автомата, поскольку задаются не только сами функции переходов и выходов, но также и все три алфавита: входной, выходной и алфавит состояний. Так как в автомате Мура выходной сигнал однозначно определяется состоянием автомата, то для его задания требуется только одна таблица, которая называется отмеченной таблицей переходов автомата Мура. Отмеченная таблица переходов автомата Мура:

y_g	$\lambda(a_0)$	…	$\lambda(a_n)$
$x_j\backslash a_c$	a_0	…	a_n
x_1	$\delta(a_0,x_1)$	…	$\delta(a_n,x_1)$
…	…	…	…
x_m	$\delta(a_0,x_m)$	…	$\delta(a_n,x_m)$

В этой таблице каждому столбцу приписан, кроме состояния $\mathbf{a}_i$, еще и выходной сигнал $\mathbf{y}(\mathbf{t}) = \boldsymbol{\lambda}(\mathbf{a}(t))$, соответствующий этому состоянию. Таблица переходов автомата Мура называется отмеченной потому, что каждое состояние отмечено выходным сигналом.

Примеры табличного задания автоматов Мили и Мура:

Автомат Мура

yg	y_2	y_1	y_1	y_3	y_2
$x_j\backslash x_j$	a_0	a_1	a_2	a_3	a_4
x_1	a_2	a_1	a_3	a_4	a_2
x_2	a_3	a_4	a_4	a_0	a_1

Автомат Мили

$x_j\backslash a_i$	a_0	a_1	a_2	a_3
x_1	a_1/y_1	a_2/y_3	a_3/y_2	a_0/y_1
x_2	a_0/y_2	a_0/y_1	a_3/y_1	a_2/y_3

По этим таблицам можно найти реакцию автомата на любое входное слово. Например:

Для автомата Мура:

$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_2}$	$\mathbf{x_2}$	$\mathbf{x_1}$	…
$\mathbf{a_0}$	$\mathbf{a_2}$	$\mathbf{a_4}$	$\mathbf{a_1}$	$\mathbf{a_4}$	
$\mathbf{y_2}$	$\mathbf{y_1}$	$\mathbf{y_2}$	$\mathbf{y_1}$	$\mathbf{y_2}$	

Для автомата Мили:

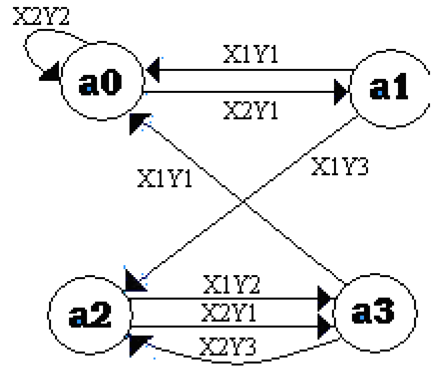
$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_2}$	$\mathbf{x_2}$	$\mathbf{x_1}$	…
$\mathbf{a_0}$	$\mathbf{a_1}$	$\mathbf{a_0}$	$\mathbf{a_0}$	$\mathbf{a_0}$	$\mathbf{a_1}$
$\mathbf{y_1}$	$\mathbf{y_1}$	$\mathbf{y_2}$	$\mathbf{y_2}$	$\mathbf{y_1}$	

Графический способ

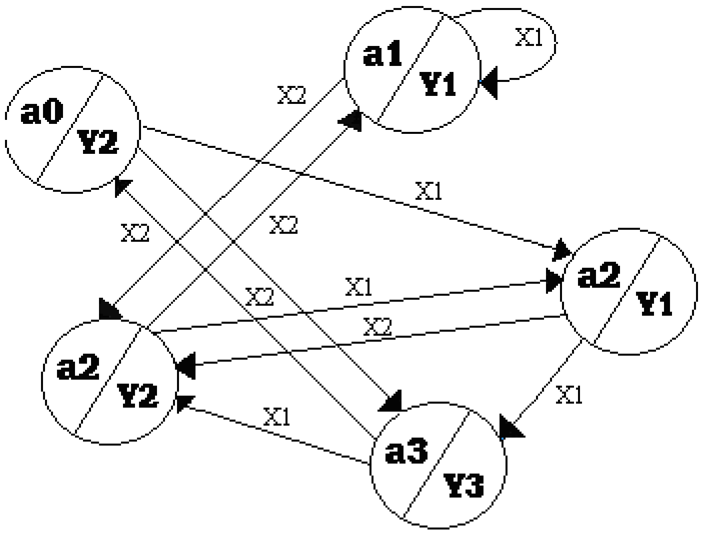
При графическом способе задание автомата осуществляется с помощью графа. Этот способ основан на использовании ориентированных связных графов. Вершины графов соответствуют состояниям автомата, а дуги –

переходам между ними. Две вершины графа $\mathbf{a}_i$ и $\mathbf{a}_s$ соединяются дугой, направленной от $\mathbf{a}_i$ к $\mathbf{a}_s$, если в автомате имеется переход из $\mathbf{a}_i$ в $\mathbf{a}_s$, то есть $\mathbf{a}_s = \delta(\mathbf{a}_i, \mathbf{x}_j)$. В автомате Мили дуга отмечается входным сигналом $\mathbf{x}_j$, вызвавшим переход, и выходным сигналом $\mathbf{y}_g$, который возникает при переходе. Внутри кружочка, обозначающего вершину графа, записывается состояние.

Граф для автомата Мили



Граф для автомата Мура

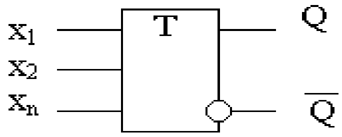


Элементарные автоматы

Элементарные автоматы, имеющие следующие особенности:

1. Они являются автоматами Мура с двумя внутренними состояниями;
2. Автомат имеет два различных выходных сигнала, соответствующих двум его внутренним состояниям. В дальнейшем состояния автомата и его выходные сигналы будем обозначать одной буквой **Q.**

Элементарные автоматы могут иметь, в общем случае, несколько физических входов:



В качестве элементарных автоматов в вычислительной технике используются триггеры. Рассмотрим некоторые из них.

***T*-триггер.** Это автомат Мура с двумя устойчивыми состояниями и одним входом **T**, который изменяет свое состояние всякий раз, когда сигнал **T**=1 . Таблица переходов **T**- триггера:

y_g	0	1
$x_j \backslash a_i$	0	1
T=0	0	1
T=1	1	0

Из таблицы переходов видно, что **T**-триггер обладает полной системой переходов и выходов, поскольку для каждой пары состояний (0-0, 0-1, 1-0, 1-1) имеется входной сигнал, обеспечивающий переход из одного состояния в другое. Кроме того, каждое состояние автомата отмечено отличным от других выходным сигналом. На практике более удобно вместо отмеченных таблиц переходов пользоваться так называемыми матрицами переходов элементарных автоматов:

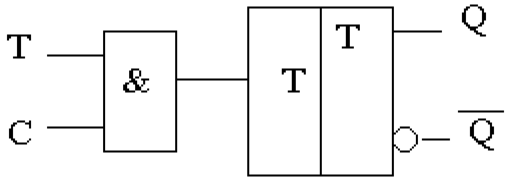
T	**Q(t)**	**Q(t+1)**
0	0	0
1	0	1
1	1	0
0	1	1

Матрица переходов определяет значения сигналов на входах элементарного автомата, обеспечивающие каждый их четырех возможных переходов. Здесь **Q**(**t**) и **Q**(**t**+1) – состояния автомата в моменты времени **t** и **t**+1 соответственно. Для записи закона функционирования **T**-триггера в аналитическом виде составим диаграмму Вейча по матрице перехода:

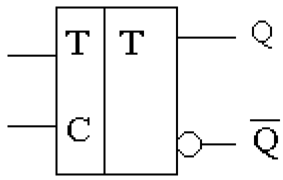
T\Q(t)	**0**	**1**
0	0	1
1	1	0

Из диаграммы имеем: $Q(t+1) = T(t)\overline{Q}(t) \vee \overline{T}(t)Q(t)$ или $Q(t+1) = T(t) \oplus Q(t)$.

Поскольку эта формула совпадает с переключательной функции сложение по модулю два, то **T**-триггер часто называют триггером со счетным входом **T**, а входной сигнал, поступающий на вход **T**, счетным сигналом. На практике кроме асинхронного используют так же и синхронный **T**-триггер, который меняет свои состояния только при **T** = 1 и **C** = 1. Вход **C** называют входом синхронизации. Поясняющая работу комбинационная схема и обозначение синхронного **T**-триггера представлены на рисунке:



Обозначение синхронного Т-триггера на функциональных схемах:



RS-триггер

RS-триггером называют автомат Мура с двумя устойчивыми состояниями, имеющий два входа **R** и **S** такие, что при **S**=1 и **R**=0 триггер принимает состояния 1, а при **R**=1 и **S**=0 – состояние 0.

В соответствие с состоянием, принимаемым триггером, вход **S** называет единичным входом, а вход **R** нулевым. **Матрица переходов RS-триггера:**

R	S	Q(t)	Q(t+1)
b_1	0	0	0
0	1	0	1
1	0	1	0
0	b_2	1	1

Комбинация сигналов **R**=1 и **S**=1 является запрещенной и поэтому переход в триггере при таких значениях входных сигналов не определен. Переход триггера из 0 в 0 возможен при двух комбинациях входных сигналов: **R**=0, **S**=0 и **R**=1, **S**=0. Поэтому в первой строке матрицы переходов **RS** триггера в столбце **R** поставлена переменная $\mathbf{b}_1$, которая может принимать два значения $0 \vee 1$. Аналогично, переход из состояния 1 в 1 также возможен при двух комбинациях входных сигналов: **R**=0, **S**=0 и **R**=0, **S**=1. Поскольку при таком переходе значения сигнала на входе S безразлично, то в нижней строке матрицы переходов в столбце S записана переменная $\mathbf{b}_2$. По матрице переходов можно построить граф **RS**-триггера.

Автоматы, которые могут переходить из одного состояния в другое под действием нескольких комбинаций входных сигналов, называются автоматами с избыточной системой переходов. Избыточность можно использовать в процессе синтеза для упрощения схемы, придавая переменным $\mathbf{b}_1$ и $\mathbf{b}_2$ такие значения, которые позволяют минимизировать число элементов. Поэтому, если схемы двух элементарных автоматов равноценны по сложности, то предпочтение отдают автомату, имеющему большую избыточность системы переходов. Запишем закон функционирования **RS**-триггера в аналитическом виде, для чего составим по матрице переходов диаграмму Вейча:

	SQ(t)			
T	00	01	10	11
0	0	1	1	1
1	0	0	-	-

Пустые клеточки соответствуют запрещенной комбинации входных сигналов. При минимизации эти клеточки можно заполнить произвольным образом, в нашем случае лучше единичным значением. Тогда имеем:

$$Q(t+1) = S \vee RQ(t).$$

JK-триггер. JK-триггером называют автомат Мура с двумя устойчивыми состояниями и двумя входами **J** и **K**, который при условии **J** * **K** = 1 меняет свое состояние на противоположное. В остальных случаях он функционируют в соответствии с таблицей истинности **RS**-триггера, при этом вход **J** эквивалентен входу **S**, а вход **K** - входу **R**. Этот триггер уже не имеет запрещенной комбинации входных сигналов и его таблица истинности имеет вид:

J	**K**	**Q(t)**	**Q(t+1)**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

По этой таблице можно построить диаграмму Вейча и матрицу переходов:

	KQ(t)			
J	00	01	10	11
0	0	1	0	0
1	1	1	0	1

Матрица переходов JK-триггера

J	**K**	**Q(t)**	**Q(t+1)**
0	b_1	0	0
1	b_2	0	1
b_3	1	1	0
b_4	0	1	1

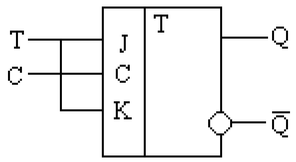
$$Q(t+1) = J\overline{Q}(t) \vee \overline{K}Q(t)$$

Универсальный триггер

Триггер **JK** типа относится к разряду универсальных триггеров, поскольку на его основе путем несложной внешней коммутации можно построить **RS-**, **D**- и **T**- триггера.

RS-триггер получается из **JK**-триггера простым наложением ограничения на комбинацию входных сигналов **J**=**K**=1, так как эта комбинация является запрещенной для **RS** триггера.

Счетный триггер на основе **JK-**триггера получается путем объединения входов **J** и **K**.

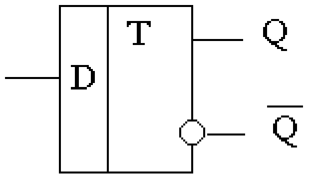


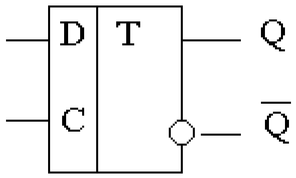
D-триггер

D-триггером (триггером задержки) называют элементарный автомат мура с двумя устойчивыми состояниями и одним входом **D** таким, что **Q**(**t**+1) = **D**(**t**). Название **D**-триггера происходит от слова “delay” – задержка. Из определения следует, что состояние триггера в момент времени **t**+1 повторяет значение входного сигнала **D**(**t**) в момент времени **t.** Матрица переходов для **D**-триггера:

D	Q(t)	Q(t+1)
0	0	0
1	0	1
0	1	0
1	1	1

Обозначения асинхронного и синхронного **D**-триггеров.

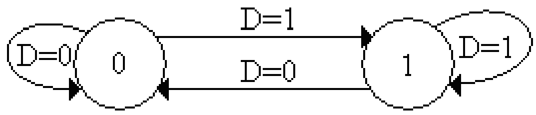




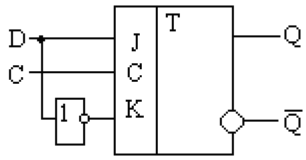
В синхронном **D**-триггере при **C**=0 триггер свое состояние не меняет, а при **C**=1 работает так же, как и асинхронный, то есть

$$\mathbf{Q}(t+1)=\mathbf{D}(t)*\mathbf{C}(t) \vee \mathbf{Q}(t)*\overline{\mathbf{C}(t)}$$

Граф D-триггера



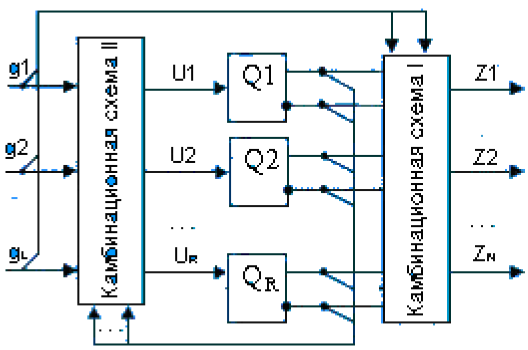
D-триггер строится на основе **JK**-триггера:



Занятие 4. Структурный синтез цифровых автоматов и их схемная реализация.

В структурной теории автомат представляют в виде композиции двух частей: запоминающей части, состоящей из элементов памяти и комбинационной части, состоящей из логических элементов.

Структурная схема



Комбинационная схема, строится на логических элементах, образующих функционально полную систему, а память – на элементарных автоматах, обладающих полной системой переходов и выходов.

Каждое состояние абстрактного автомата $\mathbf{a}_i$, где i={0, n}, кодируется в структурных автоматах набором состояний элементов памяти $\mathbf{Q}_r$, **r**={1, **R}**. Поскольку в качестве элементов памяти используются триггера, то каждое состояние можно закодировать двоичным числом $\mathbf{a_i = Q_1^{a1}Q_2^{a2}... Qr^{ar}}$.

Общее число необходимых элементов памяти можно определить из соотношения: $2^R \geq n+1$. Здесь (**n**+1) – число состояний. Логарифмируя неравенство получим $R \geq]\log_2(n+1)[$. Здесь] **C** [означает, что необходимо взять ближайшее целое число, большее или равное **C**.

В отличие от абстрактного автомата, имеющего 1 входной и 1выходной каналы, на которые поступают сигналы во входном $\mathbf{X}=\{\mathbf{x}_1, \mathbf{x}_2,..., \mathbf{x}_m\}$ и выходном $\mathbf{Y}=\{\mathbf{y}_1, \mathbf{y}_2,..., \mathbf{y}_k\}$ алфавитах, структурный автомат имеет **L** входных и **N** выходных каналов. Каждый входной $\mathbf{x}_j$ и выходной $\mathbf{y}_j$ сигналы абстрактного автомата могут быть закодированы двоичным набором состояний входных и выходных каналов структурного автомата.

$$\mathbf{x}_i = \mathbf{o}_1^{\mathbf{a1}}\,\mathbf{o}_2^{\mathbf{a2}}...\,\mathbf{o}_L^{\mathbf{aL}}$$
$$\mathbf{y_g} = \mathbf{Z}_1^{\mathbf{a1}}\mathbf{Z}_2^{\mathbf{a2}}...\,\mathbf{Z}_N^{\mathbf{aN}}$$

Здесь $\mathbf{o}_f$ и $\mathbf{Z}_h$– состояния входных и выходных каналов соответственно.

Очевидно, что число выходов **L** и **N** можно определить по формулам $L \geq]\log_2 m[$; $N \geq]\log_2 k[$, аналогичным формуле для определения **R**.

Изменение состояния элементов памяти происходит под действием сигналов $\mathbf{U}=(\mathbf{U}_1,\mathbf{U}_2,...,\mathbf{U}_r)$, поступающих на их входы. Эти сигналы формируются комбинационной схемой II и называются функций возбуждения элементарных автоматов. На вход комбинационной схемы II, кроме входного сигнала $\mathbf{x}_j$, по цепи обратной связи поступают сигналы $\mathbf{Q}=(\mathbf{Q}_1, \mathbf{Q}_2, ..., \mathbf{Q}_R)$, называемые функцией обратной связи от памяти автомата к комбинационной схеме. Комбинационная схема I служит для формирования выходного сигнала **y**g, причем в случае автомата Мили на вход этой схемы поступает входной сигнал $\mathbf{x}_j$, а в случае автомата Мура – сигнал $\mathbf{x}_j$ не поступает, так как $\mathbf{y}_g$ не зависит от $\mathbf{x}_j$.

Табличный метод структурного синтеза

Структурный синтез конечных автоматов заключается в выборе типов элементарных автоматов, в составлении функции возбуждения каждого элементарного автомата и функций кодированных выходов заданного автомата. На этапе структурного синтеза выбираем также способ кодирования состояний и выходных сигналов заданного автомата через состояния и выходные сигналы элементарных автоматов, в результате чего составляют кодированные таблицы переходов и выходов. Функции возбуждения элементарных автоматов и функции выходов получаются на основе кодированной таблицы переходов и выходов. Рассмотрим примеры синтеза, которые позволяют сформулировать общий алгоритм структурного синтеза конечных автоматов.

Пусть необходимо синтезировать автомата Мили, заданный совмещенной таблицей переходов и выходов:

$x_j \backslash a_i$	a_0	a_1	a_2
x_1	a_1/y_1	a_1/y_2	a_1/y_2
x_2	a_2/y_3	a_2/y_3	a_0/y_1

В качестве элементарных автоматов будем использовать JK-триггера, а в качестве логических элементов – элементы И, ИЛИ, НЕ. Итак, имеем $\mathbf{A}=\{\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2\}$; $\mathbf{X}=\{\mathbf{x}_1, \mathbf{x}_2\}$; $\mathbf{Y}=\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$. Здесь n=2, $\mathbf{n}+1=3$; $\mathbf{m}=2$, $\mathbf{k}=3$.

1.Перейдем от абстрактного автомата к структурному, для чего определим количество элементов памяти **R** и число входных **L** и выходных **N** каналов.

$$\mathbf{R} =]\log_2(\mathbf{n}+1)[= 2$$
$$\mathbf{L} =]\log_2 \mathbf{m}[= 1$$
$$\mathbf{N} =]\log_2 \mathbf{k}[= 2$$

Таким образом, необходимо иметь два элементарных автомата $\mathbf{Q}_1$ и $\mathbf{Q}_2$ (так как $\mathbf{R}=2$), один входной канал O_1 и два выходных канала $\mathbf{Z}_1$ и $\mathbf{Z}_2$.

2.Закодируем состояния автомата, входные и выходные сигналы:

Таблицы кодирования

состояний автомата

a_j	$\mathbf{Q}_1$	$\mathbf{Q}_2$
a_0	0	0
a_1	0	1
a_2	1	0

входных сигналов

$\mathbf{x_j}$	$\mathbf{O_1}$
$\mathbf{x_1}$	**0**
$\mathbf{x_2}$	**1**

выходных сигналов

y_g	z_1	z_2
y_1	0	0
y_2	0	1
y_3	1	0

Здесь и в дальнейшем будем использовать естественное кодирование, когда наборы значений двоичных переменных расписываются в порядке возрастания их номеров. Поскольку автомат имеет три состояния, то комбинация состояний элементарных автоматов 11 не используется и является запрещенной (автомат в это состояние никогда не попадет). С учетом кодирования составим совмещенную таблицу переходов и выходов автомата:

$x_j \backslash a_i$	**00**	**01**	**10**
0	01/00	01/01	01/01
1	10/10	10/10	00/00

3. Строим кодированные таблицы переходов и выходов. Кодированная таблица переходов определяет зависимость состояний элементарных автоматов в момент времени (**t**+1) от значения входного сигнала и внутренних состояний автоматов в предшествующий момент времени **t**, т.е.: $Qi(t+1)=fi[Q_1(t), Q_2(t),\ldots,Q_R(t),O_1(t),\ldots, O_L(t)]$.

В кодированной таблице выходов выходные сигналы $\mathbf{Z_l(t)}$ определяются в зависимости от значения входных сигналов и внутренних состояний в момент времени **t**, т.е.: $Z_l(t)=fi[Q_1(t),Q_2(t),\ldots,Q_R(t),O_1(t),\ldots,O_L(t)]$.

Кодированная таблица переходов и выходов имеет следующий вид:

(t)					(t+1)	
o_1	Q_1	Q_2	Z_1	Z_2	Q_1	Q_2
0	0	0	0	0	0	1
0	0	1	0	1	0	1
0	1	0	0	1	0	1
0	1	1	-	-	-	-
1	0	0	1	0	1	0
1	0	1	1	0	1	0
1	1	0	0	0	0	0
1	1	1	-	-	-	-

4.Основная задача, решаемая в процессе структурного синтеза – построение таблицы функций возбуждения элементарных автоматов. При построении этой таблицы используется матрица переходов **JK**-триггера.

J	K	Q(t)	Q(t+1)
0	b_1	0	0
1	b_2	0	1
b_3	1	1	0
b_4	0	1	1

С помощью матрицы переходов заполняются столбцы таблицы функций возбуждения. В строках этой таблицы записываются значения **J** и **K**, обеспечивающие нужный переход.

Таблица функций возбуждения

(t)							(t+1)	
o_1	Q_1	Q_2	J_1	K_1	J_2	K_2	Q_1	Q_2
0	0	0	0	b	1	b	0	1
0	0	1	0	b	b	0	0	1
0	1	0	b	1	1	b	0	1
0	1	1	-	-	-	-	-	-
1	0	0	1	b	0	b	1	0
1	0	1	1	b	b	1	1	0
1	1	0	b	1	0	b	0	0
1	1	1	-	-	-	-	-	-

В результате мы получим систему переключательных функций **Z**$_1$(t), **Z**$_2$(t), **J**$_1$(t), **K**$_1$(t), **J**$_2$(t) и **K**$_2$(t) заданных в виде таблиц их истинности.

5.Следующий этап – синтез комбинационной части конечных автоматов. Обычно полученные переключательные функции минимизируют и представляют в булевом базисе, а переход к заданному базису осуществляют после.

В нашем случае мы имеем 6 переключательных функций трёх аргументов, для каждой из которых построим диаграмму Вейча.

Диаграммы Вейча

J_1

	$\bar{Q}_1$		Q_1	
$\bar{\sigma}_1$	0	0	-	b
σ_1	1	1	-	b
	$\bar{Q}_2$	Q_2		$\bar{Q}_2$

K_1

	$\bar{Q}_1$		Q_1	
$\bar{\sigma}_1$	b	b	-	1
σ_1	b	b	-	1
	$\bar{Q}_1$	Q_2		$\bar{Q}_2$

$J_1 = \sigma_1$ $K_1 = 1$

		$\overline{Q}_1$		Q_1	
J_2	$\overline{\sigma}_1$	1	b	-	1
	σ_1	0	b	-	0
		$\overline{Q}_2$	Q_2		$\overline{Q}_2$

		$\overline{Q}_1$		Q_1	
K_2	$\overline{\sigma}_1$	b	0	-	b
	σ_1	b	1	-	b
		$\overline{Q}_2$	Q_2		$\overline{Q}_2$

$J_2 = \overline{\sigma}_1 \quad K_2 = \sigma_1$

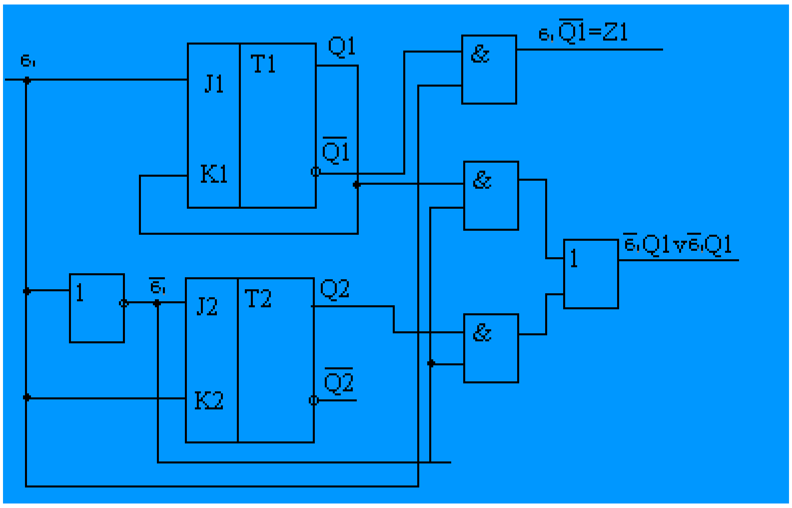
		$\overline{Q}_1$		Q_1	
Z_1	$\overline{\sigma}_1$	0	0	-	0
	σ_1	1	1	-	0
		$\overline{Q}_2$	Q_2		$\overline{Q}_2$

		$\overline{Q}_1$		Q_1	
Z_2	$\overline{\sigma}_1$	0	1	-	1
	σ_1	0	0	-	0
		$\overline{Q}_2$	Q_2		$\overline{Q}_2$

$Z_1 = \sigma_1 \overline{Q}_1$

$Z_2 = \overline{\sigma}_1 Q_1 \vee \overline{\sigma}_1 Q_2$

Обычно полученную систему ПФ минимизируют совместно. Однако совместная минимизация всех ПФ представляет собой достаточно трудоемкую и длительную операцию, применимую, в общем случае, при использовании машины. В результате минимизации мы получим следующую схему конечного автомата:



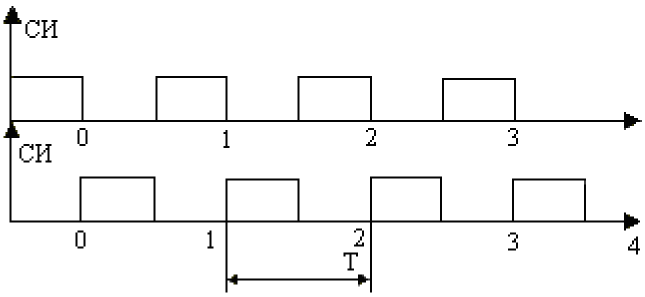
Функциональные схемы, получаемые в результате структурного синтеза, в дальнейшем на этапе инженерной доработки подвергаются изменениям.

Эти изменения связаны с тем, что добавляются специальные цепи, учитывающие технические особенностями конечных автоматов.

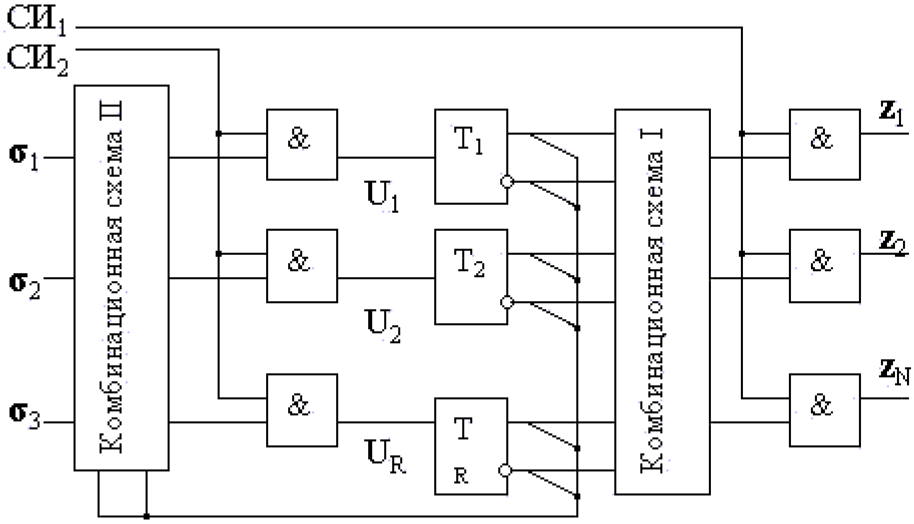
Технические особенности конечных автоматов

В схемах ЭВМ все сигналы изменяются и воспринимаются, как правило, в дискретные моменты времени, обозначаемые числами натурального ряда **t**=0, 1,…. Для отметки моментов дискретного времени ЭВМ содержит специальный блок, вырабатывающий синхронизирующие импульсы (СИ), следующие через равные интервалы времени **T**. Этот интервал времени **T** определяет такт работы устройства.

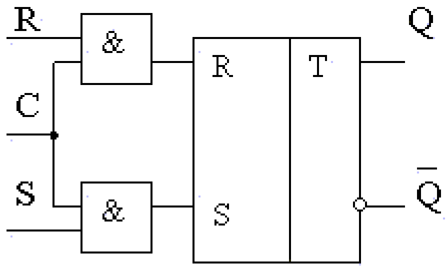
Поэтому ***первая техническая особенность*** связана с необходимостью синхронизации работы конечного автомата, причем синхронизации подлежат не только выходные сигналы, но и функции возбуждения. В связи с этим в автомат обычно вводят две серии синхроимпульсов $СИ_1$ и $СИ_2$, сдвинутых на половину периода друг против друга. Две серии синхроимпульсов $СИ_1$ и $СИ_2$

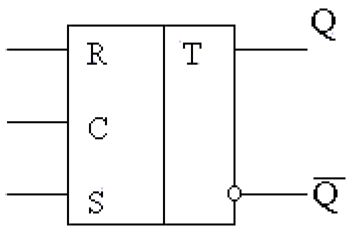


Под действием $СИ_1$, формируются выходные сигналы $\mathbf{Z}_l(t) = \mathbf{g}[\mathbf{a}(t),\mathbf{x}(t)]$, а под действием $СИ_2$ автомат переводится в новое состояние $\mathbf{a}(t+1)$.

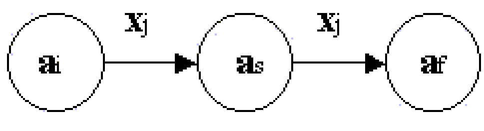


Здесь **U** – сигналы возбуждения триггера. Согласно приведенной схеме на входах каждого из триггеров стоят двухвходовые элементы И. На практике триггера часто выполняются в синхронном варианте (синхронные триггера), когда упомянутые элементы И включают в схему триггера. Например, схему синхронного триггера **RS**-типа можно рассматривать как состоящую из асинхронного **RS**-триггера, ко входам **R** и **S** которого подключены двухвходовые элементы И. На эти элементы кроме входных сигналов поступает синхронизирующий сигнал, обозначаемый буквой **C**. Очевидно, синхронные триггера будут сохранять свои состояния при **C**=0, а переходы в них возможны при **C**=1. Если **C**=1, то переходы в синхронном триггере будут осуществляться также, как в асинхронном. Применение синхронных триггеров в качестве элементов память конечного автомата облегчает организацию синхронизации таких автоматов. Синхронные триггера





Вторая техническая особенность конечного автомата связана с возможностью возникновения неустойчивых состояний и так называемых «гонок» в автомате. Понятие устойчивости заключается в следующем. Пусть в графе автомата мы имеем такой участок:

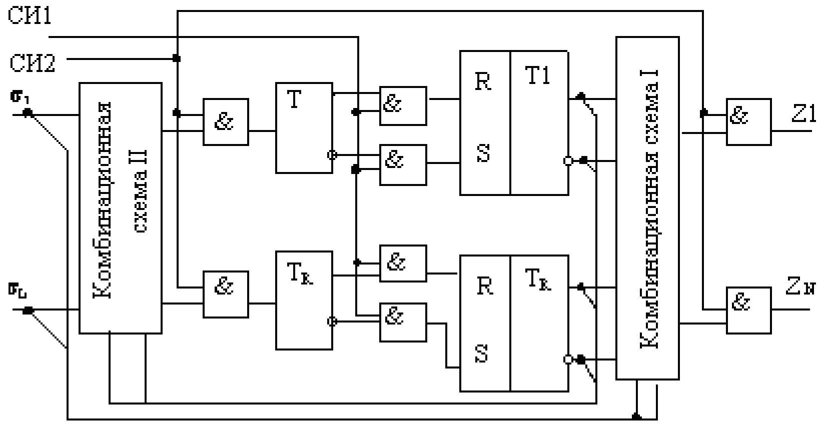


Здесь оба перехода ($\mathbf{a}_i$ --> $\mathbf{a}_s$) и ($\mathbf{a}_s$ --> $\mathbf{a}_f$) выполняются под действием одного и того же входного сигнала $\mathbf{x}_j$. Если длительность синхронизирующего сигнала $СИ_2$ больше времени перехода автомата из состояния $\mathbf{a}_i$ в состояние $\mathbf{a}_s$, то сразу после перехода автомата в $\mathbf{a}_s$ может начаться переход в следующие состояния $\mathbf{a}_f$ под действием того же входного сигнала $\mathbf{x}_j$. Таким образом автомат может перескочить состояние $\mathbf{a}_s$ и к моменту времени **t**+1 оказаться не в $\mathbf{a}_s$, как это требуется по графу, а в $\mathbf{a}_f$. Состояние $\mathbf{a}_s$ в данном случае будет неустойчивым.

Другой неприятный момент заключается в том, что при работе автомата могут возникать так называемые «гонки» (состязания). Дело в том, что триггера в схеме имеет различные времена срабатывания, а также различные времена задержек сигналов обратной связи, которые поступают с выходов триггеров на их входы через комбинационную схему II. По этим причинам, если при переходе автомата из состояния $\mathbf{a}_i$ в $\mathbf{a}_s$ должны измениться состояния нескольких триггеров, то между выходными сигналами этих триггеров начинаются гонки. Тот триггер, который выиграет гонки, то есть изменит свое состояние раньше других триггеров, может через цепь обратной связи изменить сигналы возбуждения на входах других триггеров до того момента, как они изменят свои состояния. Это, очевидно, может вызвать переход автомата совсем не в то состояние, которое нужно графу. Например. Пусть $\mathbf{a}_i$=101, $\mathbf{a}_s$=010. Тогда при переходе из $\mathbf{a}_i$ в $\mathbf{a}_s$ под действием входного сигнала $\mathbf{x}_j$ меняются состояния всех триггеров. Допустим, что первый триггер изменил свое состояние раньше других. В этом случае автомат окажется в некотором промежуточном состоянии $\mathbf{a}_h$=001, и если из этого состояния есть переход под действием сигнала $\mathbf{x}_j$ в $\mathbf{a}_l$=011, то автомат в момент времени $\mathbf{t}$+1может оказаться в $\mathbf{a}_l$, а не в $\mathbf{a}_s$.

Для устранения описанного эффекта гонок и неустойчивых состояний часто используют двойную память в автомате, когда каждый триггер в схеме дублируется. Структурная схема автомата выглядит при этом следующим образом:

Структурная схема автомата

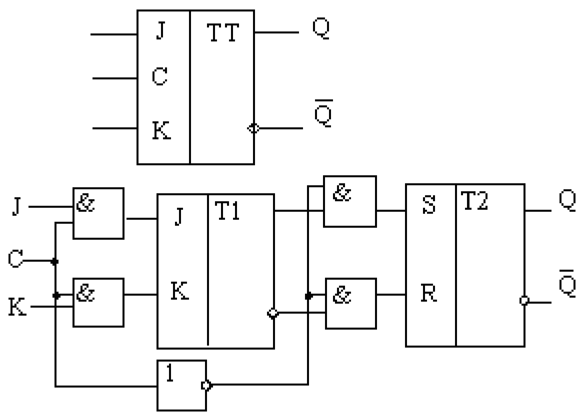


Здесь под действием синхронизирующего сигнала СИ$_1$ формируются выходные сигналы $\mathbf{Z}_l$(t) и переключаются триггера первого ряда. Под действием СИ$_2$ состояния триггеров первого ряда переписываются в соответствующие триггера второго ряда. Поскольку СИ$_2$ сдвинуты относительно СИ$_1$, а сигнал обратной связи о состоянии автомата снимается с триггеров второго ряда, то в момент поступления входного сигнала, то есть в СИ$_1$, состояние автомата не изменяется и продолжает оставаться прежним до $СИ_2$. Поэтому в такой схеме полностью обеспечивается устойчивость состояний и устраняется влияние гонок. Действительно, гонки сигналов с выходов триггеров второго ряда возможны в момент $СИ_2$, то есть. в момент переключения этих триггеров. Но в момент $СИ_2$=1, $СИ_1$=0 и следовательно эти гонки никак не могут повлиять на состояния триггеров первого ряда, которые переключаются в момент $СИ_1$=1. Также не будет и неустойчивых состояний, поскольку автомат не может проскочить за один такт через одно состояние и перейти в следующее, ибо в момент перехода триггеров первого ряда в новое состояние, то есть в $СИ_1$, состояние триггеров второго ряда не меняется ($СИ_2$=0) и, следовательно, не могут измениться и сигналы возбуждения триггеров первого ряда, которые зависят от состояния триггеров второго ряда. Поэтому автомат не может проскочить состояние.

С целью упрощения построения схем автоматов, имеющих двойную память, промышленность выпускает специальные двухступенчатые триггера. Рассмотрим работу такого триггера на примере двухступенчатого **JK**-триггера.

Особенностью двухступенчатого триггера является то, что он меняет свое состояние в момент окончания синхронизирующего сигнала **C**. В результате этого во время действия сигнала **C** выходные сигналы триггера не меняются, а происходит запись информации в триггер $\mathbf{T}_1$. В момент **C**=0 состояние триггера $\mathbf{T}_1$ переписывается в $\mathbf{T}_2$.

Двухступенчатый триггер



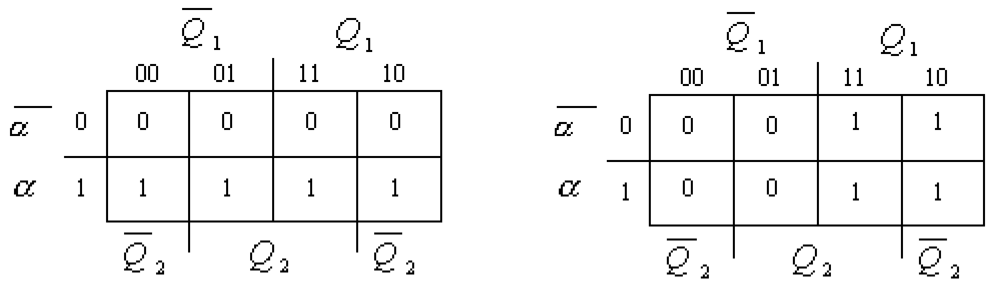
Занятие 5. Синтез регистров и счетчиков

Для приема и выдачи чисел, представленных в последовательном коде и используются регистры последовательного действия, основу которых составляют регистры сдвига. Регистр сдвига осуществляет операцию сдвига записанного в него двоичного числа влево или вправо на один или несколько разрядов при подаче специального управляющего сигнала «сдвиг». Рассмотрим процесс синтеза двухразрядного сдвигающего регистра на *D*-триггерах. Регистр должен работать следующим образом: в момент прихода синхронизирующего сигнала «*C*» код сдвигается в регистре вправо на один разряд. При этом разряды числа, сдвигаемые вправо, теряются, а в освобождающиеся слева разряды вводятся разряды числа, поступившего в последовательном коде на его вход. Имеем следующую кодированную таблицу переходов и функции возбуждения (таблица выходов не строится, ибо выходами регистра являются выходы самих триггеров).

Кодированная таблица переходов и функций возбуждения

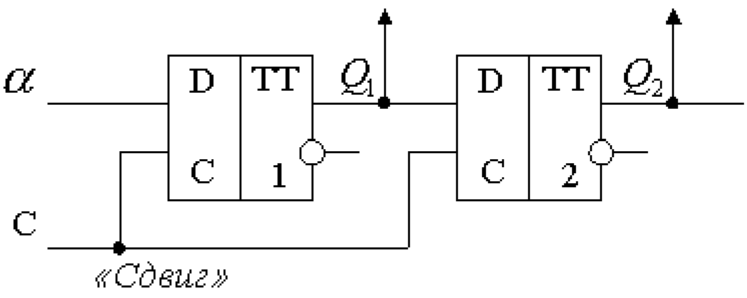
α	$Q_1(t)$	$Q_2(t)$	$Q_1(t+1)$	$Q_2(t+1)$	D_1	D_2
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	1	0	1
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	1	0	1	0
1	1	0	1	1	1	1
1	1	1	1	1	1	1

Используя диаграммы Вейча, получаем следующие минимальные дизъюнктивные нормальные формы функций возбуждения триггеров:



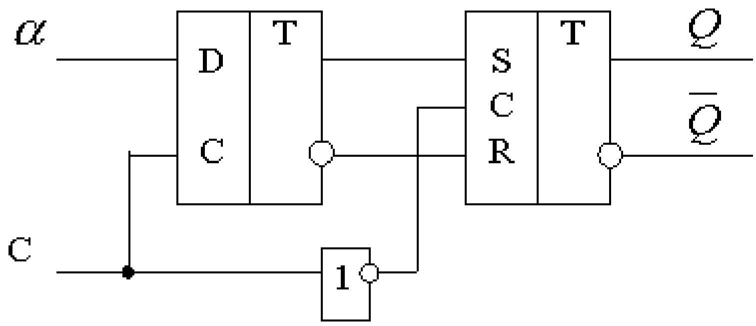
$D_1 = \alpha$ $\qquad$ $D_2 = Q_1$

Отсюда получаем следующую схему сдвигающего регистра:



С целью устранения гонок и неустойчивых состояний используются двухступенчатые *D*-триггера.

Двухступенчатый D-триггер



Аналогично строится и n-разрядный регистр сдвига, который содержит n последовательно соединенных *D*-триггеров, причем вход первого триггера является входом регистра.

По приведенной методике можно построить регистр сдвига информации влево или вправо и на другой элементной базе, например на *RS* или *JK* триггерах. Заметим, что в случае сдвига информации, хранящейся в регистре, и отсутствии входного сигнала, в освобождающиеся разряды регистра вводятся нули. Например, регистр сдвига вправо на один разряд на синхронных *JK*- триггерах имеет вид.

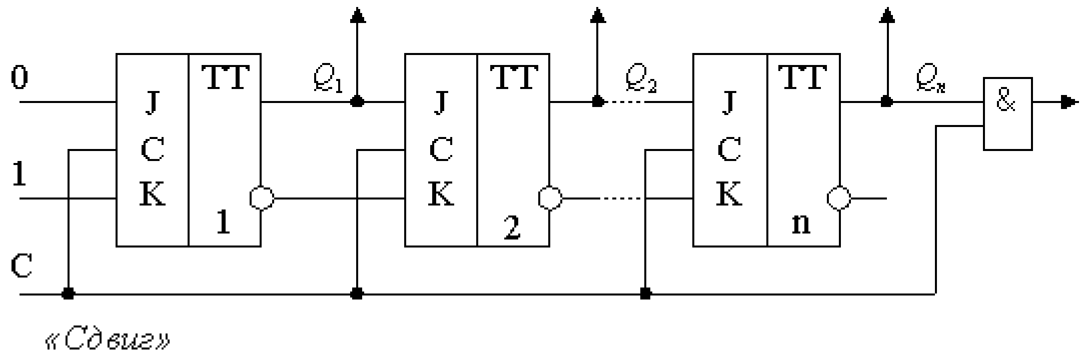
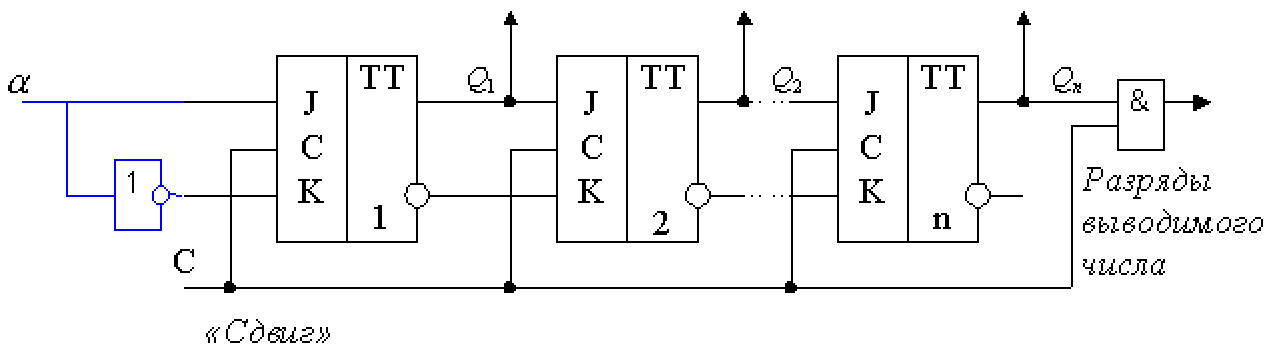
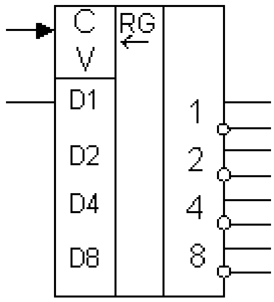


Схема регистра, дополненная схемой ввода информации



Заполнение регистра в этом случае будет происходить в течение n тактов, после чего число, находящееся в регистре, может быть прочитано в параллельном коде. Цепи ввода и вывода числа в такой регистр в параллельном коде, такие же, как и у параллельного регистра. Регистр сдвига на функциональных схемах обозначается следующим образом:



Для указания направления сдвига используется стрелка: → сдвиг в сторону старших разрядов, ← сдвиг в сторону младших разрядов.

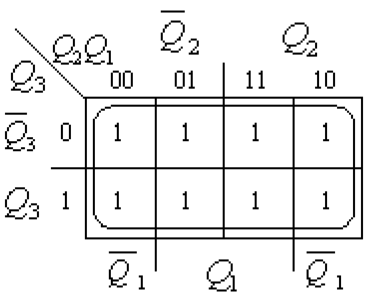
Синтез счетчиков

Синтезировать 3-х разрядный счетчик с коэффициентом пересчета $K_{ст} = 2^n = 8$. Синтез такого счетчика можно провести на основании кодированной таблицы переходов трехразрядного счетчика и таблицы функций возбуждения:

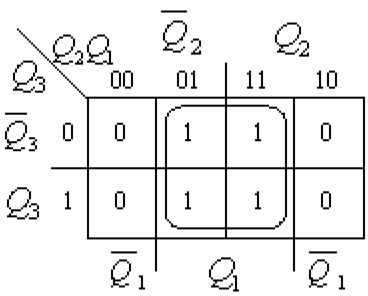
α	$Q_3(t)$	$Q_2(t)$	$Q_1(t)$	$Q_3(t+1)$	$Q_2(t+1)$	$Q_1(t+1)$	T_3	T_2	T_1
1	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0	1	1
1	0	1	0	0	1	1	0	0	1
1	0	1	1	1	0	0	1	1	1
1	1	0	0	1	0	1	0	0	1
1	1	0	1	1	1	0	0	1	1
1	1	1	0	1	1	1	0	0	1
1	1	1	1	0	0	0	1	1	1

На основании этой таблицы построим диаграммы Вейча для сигналов возбуждения триггеров, рассматривая их как функции состояний триггеров $Q_1(t)$, $Q_2(t)$, $Q_3(t)$. При $\alpha = 1$. Имеем:

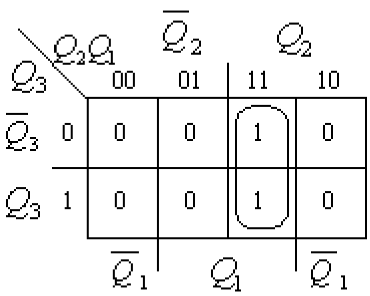
Для T_1



Для T_2



Для T_3

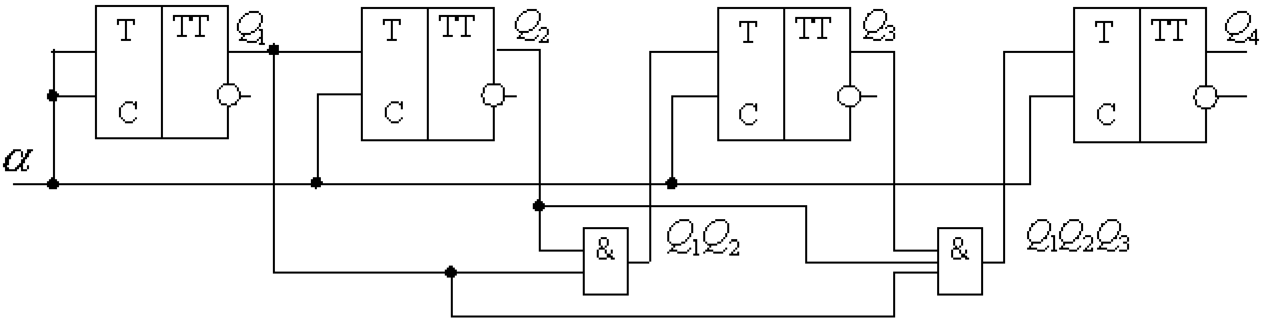


Из диаграмм получаем следующие выражения для сигналов возбуждения триггеров $T_1=\alpha$; $T_2=Q_1\alpha$; $T_3=Q_1Q_2\alpha$. Если бы мы синтезировали *n*-разрядный счетчик, то получили бы следующие выражения для сигналов возбуждения триггеров:

$T_1=\alpha$; $T_2=Q_1\alpha$; $T_3=Q_1Q_2\alpha$; $T_4=Q_1Q_2Q_3\alpha$; … $T_n=Q_1Q_2Q_3…Q_{n-1}\alpha$

Счетчик, построенный в соответствии с этими уравнениями носит название счетчика с одновременным или параллельным переносом.

Т.к. во всех функциях возбуждения присутствует входной сигнал α, то такой счетчик целесообразно строить на синхронных *T*-триггерах, подавая счетные сигналы на входы синхронизации всех триггеров. В результате получаем следующую схему:



Синтез счетчиков с коэффициентом пересчета $k\neq2^n$.

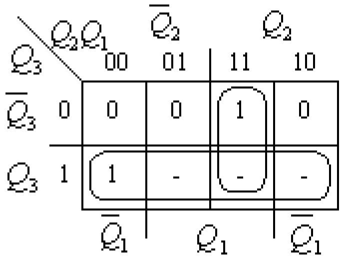
Для многих устройств ЭВМ необходимы счетчики с $k\neq2^n$. Такие счетчики называют еще пересчетными схемами.

Синтезируем счетчик с коэффициентом пересчета равным 5 (на *T*-триггерах). Такой счетчик имеет пять состояний (от 0 до 4). Количество триггеров определяется согласно формуле $R =]Log_2 k[$, где *k*=5 – число состояний. Отсюда $R =]Log_2 5[= 3$. Выходной сигнал *Z*=1 должен формироваться на каждый пятый входной сигнал. В результате получается кодированная таблица переходов, выходов и сигналов возбуждения.

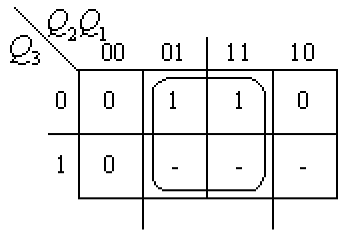
Таблица

α	$Q_3(t)$	$Q_2(t)$	$Q_1(t)$	$Q_3(t+1)$	$Q_2(t+1)$	$Q_1(t+1)$	T_3	T_2	T_1	Z
1	0	0	0	0	0	1	0	0	1	0
1	0	0	1	0	1	0	0	1	1	0
1	0	1	0	0	1	1	0	0	1	0
1	0	1	1	1	0	0	1	1	1	0
1	1	0	0	0	0	0	1	0	0	1

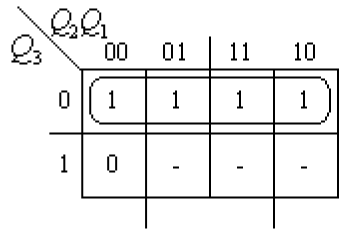
При α=1 строим следующие диаграммы Вейча



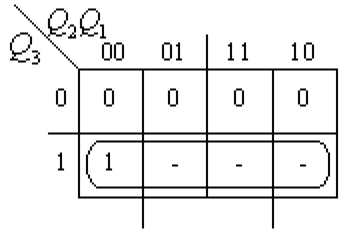
Для T_3



Для T_2



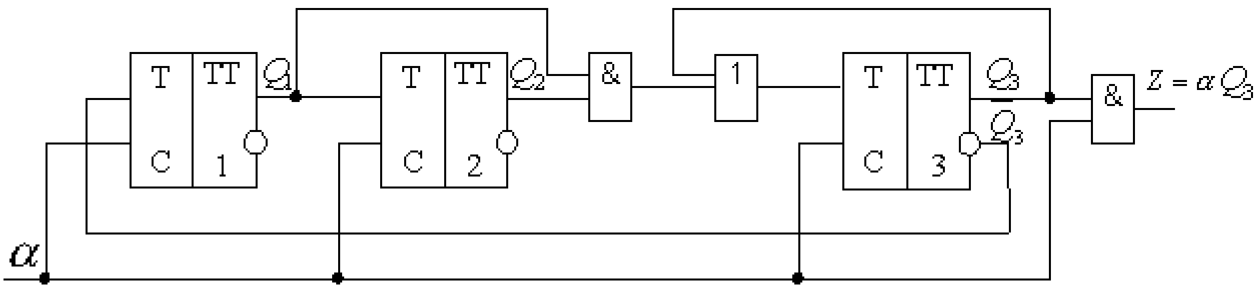
Для T_1



Для Z

Из диаграмм Вейча получаем следующие выражения для трех функций возбуждения и функции выходов: $T_3=\alpha\,(Q_3 \cup Q_1Q_2)$; $T_2=Q_1\alpha$; $T_1=\alpha\bar{Q}_3$; $Z=\alpha Q_3$

Это синхронный пятеричный счетчик:

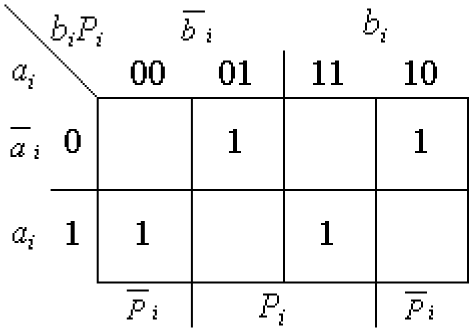


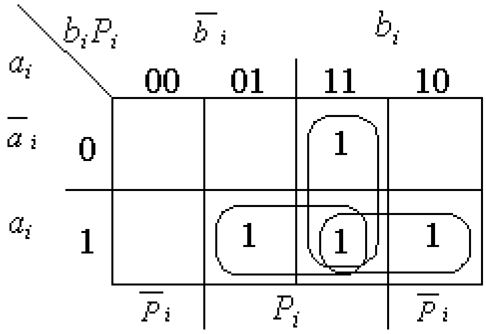
Занятие 6. Синтез сумматоров. Синтез одноразрядного двоичного сумматора.

Закон функционирования такого сумматора при сложении трех цифр определяется следующей таблицей.

a_i	$в_i$	P_i	P_{i+1}	S_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

На основании таблицы строятся диаграммы Вейча и получаются минимальные ДНФ функций S_i и P_{i+1}



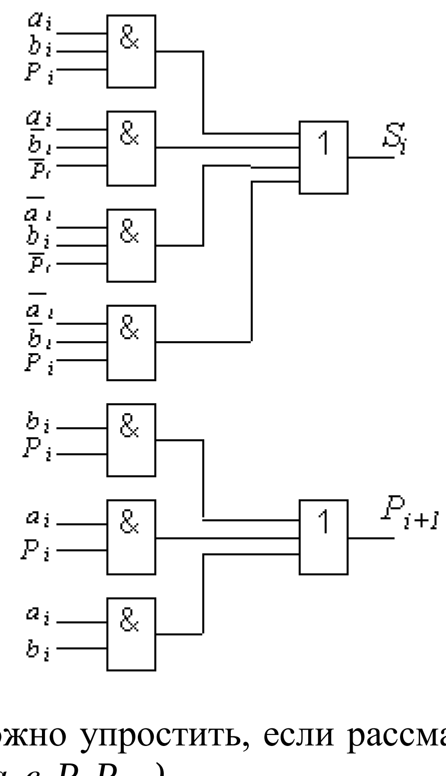


$$S_i = a_i b_i P_i \cup a_i \bar{b}_i \bar{P}_i \cup \bar{a}_i b_i \bar{P}_i \cup \bar{a}_i \bar{b}_i P_i$$

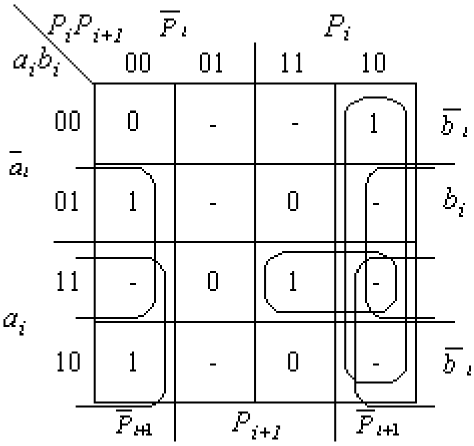
$$P_{i+1} = в_i P_i \cup a_i P_i \cup a_i в_i = a_i в_i \cup (a_i \cup в_i) P_i$$

По полученным уравнениям можно построить двухуровневую схему одноразрядного комбинационного сумматора.

Двухуровневая схема одноразрядного комбинационного сумматора



Полученную схему можно упростить, если рассматривать S_i как функцию 4-х переменных $S_i=S_i(a_i,в_i,P_i,P_{i+1})$.



Отсюда имеем:

$$S_i = b_i\overline{P}_{i+1} \cup a_i\overline{P}_{i+1} \cup P_i\overline{P}_{i+1} \cup a_ib_iP_i = (a_i \cup b_i \cup P_i)\overline{P}_{i+1} \cup a_ib_iP_i$$

$$\overline{P}_{i+1} = a_ib_i \cup (a_i \cup b_i)P_i$$

Схема сумматора:

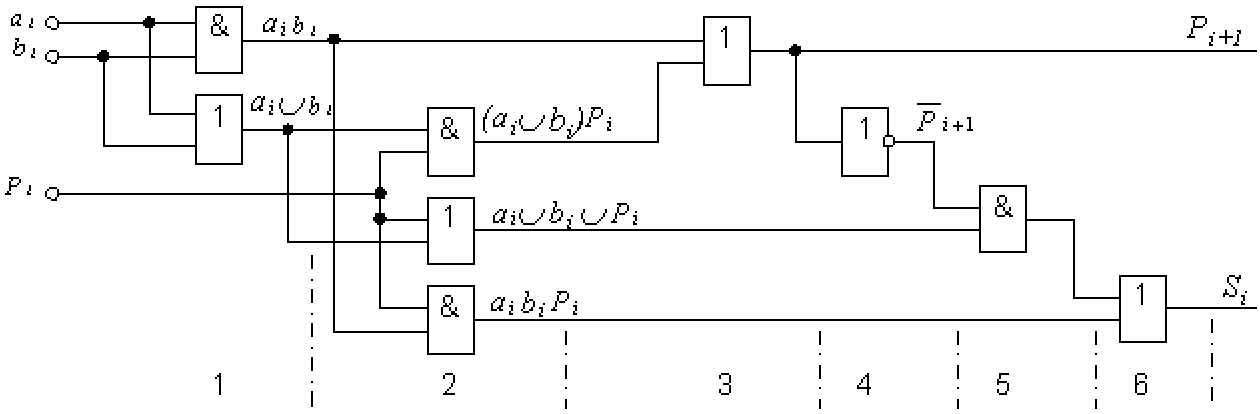


Схема имеет число уровней здесь r=6, сложность равна 17.

Занятие 7. Абстрактный синтез цифровых автоматов. Алгебра событий. Составление и разметка регулярных выражений. Построение отмеченной таблицы переходов автомата Мура.

Представление событий в автоматах

В основе рассматриваемого способа задания автоматов, лежит понятие ***событий***, представимых в автоматах.

Пусть $\mathbf{Y}\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$ – выходной алфавит конечного автомата $\mathbf{S}$ с фиксированным начальным состоянием $\mathbf{a}_0$. Тогда каждой букве $\mathbf{y}_j$, выходного алфавита можно поставить в соответствие множество входных слов $\mathbf{S}_j(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)$, которые вызывают появление на выходе автомата буквы y_j. Определенное таким образом множество слов $\mathbf{S}_j(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)$ называют событием, представленным в автомате выходным сигналом $\mathbf{y}_j$.

Поэтому для задания конечного автомата, имеющего выходной алфавит $\mathbf{Y}\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$, достаточно разбить множество всех возможных входных слов на K событий $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_k$, представленных в автомате выходными сигналами $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k$ соответственно. Для частичного автомата необходимо, кроме того, задать множество $\mathbf{S}_з$ запрещенных слов. Таким образом, конечный автомат может быть задан таблицей, устанавливающей соответствия между событиями и буквами выходного алфавита. Зная набор событий $\mathbf{S}_j$, можно, не пользуясь таблицами переходов и выходов, найти реакцию автомата на любое входное слово, для чего достаточно определить в множество каких слов входного алфавита оно входит (то есть какому событию принадлежит).

Для описания автоматов на языке регулярных событий вводят ряд операций над событиями, то есть строят алгебру событий. Мы рассмотрим алгебру событий, введенную Клини и усовершенствованную академиком Глушковым В. М.

Соответствие событий буквам выходного алфавита

Событие	Буква
$S_1(x_1, x_2,\ldots, x_m)$	y_1
$S_2(x_1, x_2,\ldots, x_m)$	y_2
…	…
$S_k(x_1, x_2,\ldots, x_m)$	y_k
$S_з(x_1, x_2,\ldots, x_m)$	-

Операции в алгебре событий

Алгебра событий включает три операции:

- Дизъюнкцию (объединение) событий;
- Произведение событий;
- Итерацию событий.

Дизъюнкцией событий называют событие $\mathbf{S} = \mathbf{S}_1 \text{v } \mathbf{S}_2 \text{v} \ldots \text{v} \mathbf{S}_k$, состоящее из всех слов, входящих в события$\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_k$.

Пример. Событие $\mathbf{S}_1$ содержит слова $\mathbf{x}_1$, $\mathbf{x}_2\mathbf{x}_1$, $\mathbf{x}_1\mathbf{x}_1$, т.е. $\mathbf{S}_1 = (\mathbf{x}_1, \mathbf{x}_2\mathbf{x}_1, \mathbf{x}_1\mathbf{x}_1)$, а$\mathbf{S}_2 = (\mathbf{x}_2, \mathbf{x}_1\mathbf{x}_2)$. Тогда $\mathbf{S} = \mathbf{S}_1 \text{ v } \mathbf{S}_2 = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_1\mathbf{x}_1, \mathbf{x}_1\mathbf{x}_2\mathbf{x}_2\mathbf{x}_1)$.

Произведением событий называется событие $\mathbf{S} = \mathbf{S}_1* \ \mathbf{S}_2* \ \ldots,* \ \mathbf{S}_k$, состоящее из всех слов, полученных приписыванием к каждому слову события **S**1 каждого слова события $\mathbf{S}_2$, затем слова события $\mathbf{S}_3$ и т.д.

Пример. $\mathbf{S}_1$и $\mathbf{S}_2$ те же. Тогда $\mathbf{S} = \mathbf{S}_1* \ \mathbf{S}_2 = (\mathbf{x}_1\mathbf{x}_2, \mathbf{x}_1\mathbf{x}_1\mathbf{x}_2, \mathbf{x}_2\mathbf{x}_1\mathbf{x}_2, \mathbf{x}_2\mathbf{x}_1\mathbf{x}_1\mathbf{x}_2, \mathbf{x}_1\mathbf{x}_1\mathbf{x}_2, \mathbf{x}_1\mathbf{x}_1\mathbf{x}_1\mathbf{x}_2)$. Произведение событий не коммутативно, то есть слова, входящие в события $\mathbf{S}_1*\mathbf{S}_2$ и $\mathbf{S}_2\mathbf{*S}_1$ различны: то есть $\mathbf{S}_1\mathbf{*S}_2? \ \mathbf{S}_2\mathbf{*S}_1$. Поскольку произведение не коммутативно, следует различать операции «умножение справа» и «умножение слева». Например, относительно произведения событий $\mathbf{S}_1\mathbf{*S}_2$ можно сказать, что событие $\mathbf{S}_2$ умножено на событие $\mathbf{S}_1$справа, а событие $\mathbf{S}_1$на $\mathbf{S}_2$ слева.

Третьей операцией, применяемой в алгебре событий, является одноместная операция итерация, которая применима только к одному событию. Для обозначения итерации вводят фигурные скобки, которые называются итерационными.

Итерацией события **S** называется событие{**S**}, состоящее из пустого слова ***e*** и всех слов вида **S**, **SS**, **SSS** и т.д. до бесконечности. Т.е. {**S**} = ***e*** v **S** v **SS** v **SSS** v …

Пример.

$\mathbf{S} = (\mathbf{x}_2, \mathbf{x}_1\mathbf{x}_2)\{\mathbf{S}\} = (\boldsymbol{e}, \mathbf{x}_2, \mathbf{x}_2\ \mathbf{x}_2, \mathbf{x}_2\mathbf{x}_2\mathbf{x}_2, \ldots, \mathbf{x}_1\mathbf{x}_2, \mathbf{x}_1\mathbf{x}_2\mathbf{x}_1\mathbf{x}_2, \ldots, \mathbf{x}_2\mathbf{x}_1\mathbf{x}_2, \mathbf{x}_1\mathbf{x}_2\mathbf{x}_2, \ldots)$

При синтезе конечных автоматов важнейшую роль играют регулярные события. Очевидно любое событие, состоящее из конечного множества слов, является регулярным. Действительно, такие события можно представить в виде дизъюнкции всех входящих в него слов, образованных из букв заданного алфавита с помощью операции умножения. События, состоящие из бесконечного числа слов, могут быть как регулярными, так и не регулярными.

Теорема: Любые регулярные выражения и только они представимы в конечных автоматах.

Из этой теоремы следует, что любой алгоритм преобразования информации, который можно записать в виде регулярного выражения, реализуется конечным автоматом. С другой стороны, любые конечные автоматы реализуют только те алгоритмы, которые могут быть записаны в виде регулярных выражений.

Рассмотрим, как можно совершить переход от описательной формы задания алгоритмов работы конечных автоматов к представлению этих алгоритмов в виде регулярных выражений. С целью упрощения такого перехода вводят основные события, из которых с помощью операций дизъюнкции, умножения и итерации можно составить более сложные события, соответствующие заданному алгоритму работы автомата. За основные события принимают такие события, которые более часто встречаются в инженерной практике при синтезе схем ЭВМ. Пусть дан конечный алфавит $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)$.

Регулярное событие – это любое событие, которое можно получить из букв данного алфавита с помощью конечного числа операций дизъюнкции, произведения и итерации.

Регулярное выражение – любое выражение, составленное с помощью операций дизъюнкции, произведения и итерации.

Система основных событий

В эту систему мы включим те из наиболее часто встречающихся событий, которые используются при записи регулярных выражений на практических занятиях и курсовой работе.

Пусть дан алфавит $\mathbf{X}\{ \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m \}$.

1. Событие, состоящее из всех слов входного алфавита (всеобщее событие).

$\mathbf{F = \{x_1 \vee x_2 \vee \ldots \vee x_m\}}$

2. Событие, содержащее все слова, оканчивающиеся буквой $\mathbf{x}_i$.

$\mathbf{S = \{ x_1 \vee x_2 \vee \ldots \vee x_i \vee \ldots \vee x_m \} xi = Fx_i}$

3. Событие, содержащее все слова, оканчивающиеся отрезком слова $\boldsymbol{l}_1$$\mathbf{S = F\ l_1}$

4. Событие, содержащее все слова, начинающиеся с отрезка слова $\boldsymbol{l}_1$ и оканчивающиеся на $\boldsymbol{l}_2$:$\mathbf{S = l_1\ F\ l_2}$

5. Событие, содержащее только однобуквенные слова входного алфавита

$\mathbf{S = x_1 \vee x_2 \vee \ldots \vee x_m}$

6. Событие, содержащее только двухбуквенные слова входного алфавита

$\mathbf{S = (x_1 \vee x_2 \vee \ldots \vee x_m)(x_1 \vee x_2 \vee \ldots \vee x_m)}$

7. Событие, содержащее все слова длиной r

$\mathbf{S = (x_1 \vee x_2 \vee \ldots \vee x_m)(x_1 \vee x_2 \vee \ldots \vee x_m) \ldots (x_1 \vee x_2 \vee \ldots \vee x_m)}$ - r членов

8. Событие, содержащее все слова, длина которых кратна r

$\mathbf{S = \{(x_1 \vee x_2 \vee \ldots \vee x_m)(x_1 \vee x_2 \vee \ldots \vee x_m)\ldots(x_1 \vee x_2 \vee \ldots \vee x_m)\}}$ - r членов

9. Событие, состоящее из всех слов алфавита $\mathbf{X}\{ \mathbf{x}_1, \mathbf{x}_2\}$, не содержащих комбинации букв $\mathbf{x}_1\mathbf{x}_1$ и оканчивающихся буквой $\mathbf{x}_2$

$\mathbf{S = \{ x_2 \vee x_1 x_2\}}$

10. Событие, состоящее из всех слов алфавита $\mathbf{X}\{ \mathbf{x}_1, \mathbf{x}_2\}$, не содержащих серии из **r** букв $\mathbf{x}_1$ и оканчивающихся буквой $\mathbf{x}_2$

$\mathbf{S = \{ x_2 \vee x_1 x_2 \vee x_1x_1 x_2 \vee \ldots \vee x_1x_1\ldots x_1 x_2\}}$ - (r-1) членов.

Рассмотрим пример составления регулярного выражения, определяющего закон функционирования конечного автомата.

Пример.Записать в виде регулярного выражения алгоритм работы автомата, сравнивающего два двоичных числа, представленных в последовательном коде. Количество разрядов числа – произвольно.

-------------------------	КА	------------>
$\mathbf{X}\{\mathbf{x}_{00},\mathbf{x}_{01},\mathbf{x}_{10},\mathbf{x}_{11},\mathbf{x}_s\}$		$\mathbf{Y}\{\mathbf{y}_1,\mathbf{y}_2,\mathbf{y}_3\}$

Окончание чисел фиксируется подачей на вход автомата сигнала $\mathbf{x}_s$. Если число, поданное на первый вход автомата, меньше числа, поданного на второй вход, то КА выдает сигнал $\mathbf{y}_1$, если больше – то $\mathbf{y}_2$, если оба числа равны – то $\mathbf{y}_3$. Числа подаются на входы автомата младшими разрядами вперед. На входы автомата сравнения одновременно может поступить одна из четырех комбинаций сигналов 00,01,10,11, которые закодируем следующим образом $\mathbf{x}_{00}$=00, $\mathbf{x}_{01}$=01, $\mathbf{x}_{10}$=10, $\mathbf{x}_{11}$=11. При этом будем считать, что первая цифра каждой комбинации относится к первому входу, а вторая – ко второму входу. Таким образом, входной алфавит автомата включает пять букв $\mathbf{X}\{\mathbf{x}_{00}, \mathbf{x}_{01}, \mathbf{x}_{10}, \mathbf{x}_{11}, \mathbf{x}_s \}$, а выходной три буквы $\mathbf{Y}\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$.

Два двоичных числа равны, если равны цифры в любых одинаковых разрядах. Поэтому событие, заключающееся в поступлении на вход автомата равных чисел, состоит из всех возможных слов, содержащих буквы $\mathbf{x}_{00}$ и $\mathbf{x}_{11}$. Т.е. $\mathbf{S}_3 = \{ \mathbf{x}_{00} \vee \mathbf{x}_{11}\} \mathbf{x}_s$.

События, представленные в автомате сигналами $\mathbf{y}_1$ и $\mathbf{y}_2$ можно записать в виде:

$\mathbf{S}_1=\{\mathbf{x}_{00}\vee\mathbf{x}_{01}\vee\mathbf{x}_{10}\vee\mathbf{x}_{11}\}\mathbf{x}_{01}\{\mathbf{x}_{00}\vee\mathbf{x}_{11}\}\mathbf{x}_s$

$\mathbf{S}_2=\{\mathbf{x}_{00}\vee\mathbf{x}_{01}\vee\mathbf{x}_{10}\vee\mathbf{x}_{11}\}\mathbf{x}\mathbf{x}_{10}\{\mathbf{x}_{00}\vee\mathbf{x}_{11}\}\mathbf{x}_s$.

События $\mathbf{S}_1$, $\mathbf{S}_2$ и $\mathbf{S}_3$ не охватывают всего множества слов, которые могут быть записаны в алфавите $\mathbf{X}\{\ \mathbf{x}_{00}, \mathbf{x}_{01}, \mathbf{x}_{10}, \mathbf{x}_{11}, \mathbf{x}_s\}$, т.к. в эти события входят только слова, оканчивающиеся буквой $\mathbf{x}_s$. Слова, не входящие в $\mathbf{S}_1$, $\mathbf{S}_2$ и $\mathbf{S}_3$, должны быть представлены в автомате пустой буквой $\boldsymbol{e}$: $\mathbf{S}_4 = \mathbf{S}_1 \vee \mathbf{S}_2 \vee \mathbf{S}_3$.Очевидно, что записанные выражения можно упростить, если входные сигналы 00 и 11 закодировать одной буквой, например $\mathbf{x}_r$. Такое кодирование возможно, т.к. КА одинаково реагирует на эти комбинации.

$\mathbf{S}_1=\{\mathbf{x}_r \mathrm{v} \mathbf{x}_{01} \mathrm{v} \mathbf{x}_{10}\}\mathbf{x}_{01}\{\mathbf{x}_r\}\mathbf{x}_s$
$\mathbf{S}_2=\{\mathbf{x}_r \mathrm{v} \mathbf{x}_{01} \mathrm{v} \mathbf{x}_{10}\}\mathbf{x}_{10}\{\mathbf{x}_r\}\mathbf{x}_s$
$\mathbf{S}_3=\{\mathbf{x}_r\}\mathbf{x}\mathrm{s}$ $\mathbf{S}_4=\mathbf{S}_1 \mathrm{v} \mathbf{S}_2 \mathrm{v} \mathbf{S}_3$

Заметим, что одно и тоже регулярное событие может быть представлено различными регулярными выражениями. Поэтому встает задача отыскания таких регулярных выражений, которые позволяют представлять события наиболее простыми формулами. Рассмотрим несколько основных соотношений, которые используются при преобразовании регулярных выражений.

1. $S_1 \vee S_2 = S_2 \vee S_1$ - закон коммутативности
2. $(S_1 \mathrm{v} S_2) \mathrm{v} S_3 = S_1 \mathrm{v} (S_2 \mathrm{v} S_3) = S_1 \mathrm{v} S_2 \mathrm{v} S_3$ - законы ассоциативности
3. $S_1*(S_2*S_3) = (S_1*S_2)*S_3$ - законы ассоциативности
4. $S_1(S_2 \mathrm{v}\ S_3) = S_1S_2 \vee S_1S_3$ - закон дистрибутивности
5. $\{\{S\}\} = \{S\}$
6. $\{S\}*\{S\} = \{S\}$
7. $\{\{S_1\} \vee \{S_2\} = \{\ S_1 \vee S_2\}$
8. $\{e\} = e$
9. $eS = Se = S$

Методы абстрактного синтеза

Задача абстрактного синтеза заключается в составлении таблиц переходов и выходов автоматов по заданным условиям его функционирования, представленным в форме регулярных выражений.

Абстрактный синтез обычно выполняется в два этапа:

1. Первый этап заключается в получении таблиц переходов и выходов в некоторой исходной форме. Построенный по этим таблицам автомат обычно содержит «лишние» внутренние состояния.
2. На втором этапе производится минимизация количества внутренних состояний заданного автомата.

Синтезируемый автомат может быть задан либо как автомат Мура, либо как автомат Мили. Поскольку для автомата Мура всегда можно построить эквивалентный автомат Мили, то достаточно рассмотреть алгоритм синтеза автомата Мура, который проще автомата Мили.

Алгоритм синтеза автомата Мура

Рассмотрим пример абстрактного синтеза автомата для случая, когда регулярные отношения составлены без применения операции итерации. Составим отмеченную таблицу переходов автомата, имеющего входной алфавит $\mathbf{X}\{\mathbf{x}_1, \mathbf{x}_2\}$ и реализующий следующий алгоритм.

$$\mathbf{S}_1 = \mathbf{x}_1\mathbf{x}_2 \vee \mathbf{x}_1\mathbf{x}_1\mathbf{x}_1$$

$$\mathbf{S}_2 = \mathbf{x}_1\mathbf{x}_2\mathbf{x}_2 \vee \mathbf{x}_2\mathbf{x}_2$$

$$\mathbf{S}_{\text{запр.}} = \mathbf{x}_1 \vee \mathbf{x}_2\mathbf{x}_2\mathbf{x}_2$$

При синтезе условимся начальное состояние автомата обозначать цифрой 0, а остальные состояния – десятичными числами 1, 2, 3 и т.д. Очевидно, самый простой, хотя и не экономный по числу используемых внутренних состояний автомата, алгоритм синтеза заключается в следующем. Фиксируется начальное состояние и для входного слова, содержащего **r** букв, назначается **r** внутренних состояний. Переходы в автомате определяются так, что первая буква входного слова переводит автомат из начального состояния 0 в состояние 1, вторая буква из 1 в 2 и т.д. Аналогичные последовательности внутренних состояний назначаются для всех остальных слов. Затем все конечные состояния, в которые автомат попадает после подачи слов, входящих в событие $\mathbf{S}_i$, отмечаются выходными сигналами $\mathbf{y}_i$. Чтобы система переходов автомата была определенной, для всех слов, имеющих одинаковые начальные отрезки, следует назначать одну и ту же последовательность состояний. Например, для регулярного события $\mathbf{S}_1$ первая буква $\mathbf{x}_1$ переводит автомат из начального состояния 0 в состояние 1, вторая буква $\mathbf{x}_2$ – из 1 в 2.

$$\begin{array}{lccccccccccccc}
\mathbf{S_1} = & | & \mathbf{x}_1 & | & \mathbf{x}_2 & | & \vee & | & \mathbf{x}_1 & | & \mathbf{x}_1 & | & \mathbf{x}_1 & | \\
 & 0 & & 1 & & 2 & & 0 & & 1 & & 3 & & 4 \\
\mathbf{S_2} = & | & \mathbf{x}_1 & | & \mathbf{x}_2 & | & \mathbf{x}_2 & | & \vee & | & \mathbf{x}_2 & | & \mathbf{x}_2 & | \\
 & 0 & & 1 & & 2 & & 5 & & 0 & & 6 & & 7
\end{array}$$

Поскольку первая буква второго слова $\mathbf{x}_1\mathbf{x}_1\mathbf{x}_1$, входящего в $\mathbf{S}_1$ также есть $\mathbf{x}_1$, то она переводит автомат из начального состояния 0 в 1. Вторая буква $\mathbf{x}_1$ переводит автомат из 1 в 3, третья – из 3 в 4.

Первые две буквы слова $\mathbf{x}_1\mathbf{x}_2\mathbf{x}_2$, входящего в $\mathbf{S}_2$, совпадают с первым словом события $\mathbf{S}_1$. Поэтому первые две буквы этого слова должны последовательно переводить автомат из 0 в 1, и из 1 в 2. Дальнейшие состояния обозначим числами 5, 6 и 7. Получившаяся в результате форма записи определяет разметку мест регулярных выражений.

Местами регулярного выражения называют промежутки между двумя буквами, между буквой и знаком дизъюнкции, а так же между буквой и скобкой. Кроме того, вводят начальное место, обозначаемое цифрой 0 и конечные места, отождествляемые с концом каждого слова. Для запрещенного события $\mathbf{S}_{\text{запр}}$ последовательность событий можно не назначать.

Для размеченных регулярных выражений составляется отмеченная таблица переходов.

Отмеченная таблица переходов

	e	e	y_1	e	y_1	y_2	e	y_2	e
$x_j \backslash a_i$	0	1	2	3	4	5	6	7	*
x_1	1	3	*	4	*	*	*	*	*
x_2	6	2	5	*	*	*	7	*	*

Чтобы система переходов автомата была определена при подаче любого входного слова, кроме состояний 0 ? 7, вводится еще одно состояние, которое обозначается звездочкой *. В это состояние автомат переходит при подаче входных слов, которые не входят в события $\mathbf{S}_1$ и $\mathbf{S}_2$. Выходным сигналом $\mathbf{y}_1$ отмечены состояния 2 и 4, $\mathbf{y}_2$ – состояния 5 и 7. Остальные состояния отмечены пустой буквой ***e***.

Алгоритм синтеза усложняется, если регулярные выражения содержат итерационные скобки. При разметке регулярных выражений различают <u>***основные***</u> и <u>***предосновные места.***</u>

Очевидно, некоторые места могут быть одновременно основными и предосновными.

Все основные места отмечаются различными десятичными числами, при этом всем начальным местам приписывается индекс 0. Затем каждое предосновные место отмечается совокупностью индексов основных мест. В эту совокупность входят индексы внутренних состояний, находясь в которых автомат может принять букву, стоящую справа от предосновного места.

Разметка регулярных выражений. Разметка регулярных выражений проводится ***по правилам подчинения мест.***

Общие правила подчинения мест регулярного выражения

1. Индекс места перед любыми скобками распространяется на начальные места всех дизъюнктивных членов, записанных в этих скобках.
2. Индекс конечного места, любого дизъюнктивного члена, заключенного в любые скобки, распространяется на место, непосредственно следующее за этими скобками.
3. Индекс места перед итерационными скобками распространяется на место, непосредственно следующее за этими скобками.
4. Индекс конечного места любого дизъюнктивного члена, заключенного в итерационные скобки, распространяется на начальные места всех дизъюнктивных членов, заключенных в эти итерационные скобки.
5. Индексы мест, слева и справа от которых стоят буквы, никуда не распространяются.
6. В автоматах многократного действия индекс конечного места всего выражения распространяется на те же места, на которые распространяется индекс начального места. Это правило справедливо только в тех случаях, когда событие представлено регулярным выражением так, что оно не содержит многократно повторяющихся слов, входящих в заданное событие. И тогда организация автомата многократного действия осуществляется путем разметки.

Смысл приведенных правил подчинения мест сводится к следующему: основному месту с индексом **i** подчиняется место **j**, если автомат, находящийся в состоянии **i**, может принять букву входного алфавита, записанную непосредственно справа от места **j**. Рассмотрим эти правила на примере синтеза автомата, описываемого следующим

регулярным выражением:

```
S= | { | X2 | v | X1 | X2 | v | X1 | X1 | X2 |} | X1 | X1 | X1| { | X1 | } | X2 |
     0 |   1    |       2    3  |    4    5    6  |    7      8  9 |   10    |   11
-----------------------------------------------------------------------------------
       0        0       2       0    4    5       0    7      8    9         9
       1        1               1                 1               10        10
       3        3               3                 3
       6        6               6                 6
      11       11              11                11
```

В этом автомате сигнал $\mathbf{y}_1$ выдается после поступления подряд 3-ех букв $\mathbf{x}_1$, а $\mathbf{y}_2$ – после $\mathbf{x}_2$, следующей за серией их трех и более букв $\mathbf{x}_1$. В остальных случаях выдается буква ***e***.

Индексы основных мест записываются непосредственно под регулярными выражениями, а индексы предосновные мест располагаются ниже индексов основных мест, под горизонтальной чертой. Выражение имеет 12 основных мест (от 0 до 11).

Проведем разметку предосновных мест. В начале определим, какие буквы может принять автомат, если он находится в состоянии 0. Поскольку на вход автомата может поступить любое из трех слов, записанных в итерационных скобках, то индекс 0 распространяется на каждое из трех предосновных мест, расположенных в начале этих слов. Учитывая, что событие, соответствующее выражению, записанному в итерационных скобках, содержит пустое слово ***e***, индекс 0 распространяется на предосновное место, расположенное сразу за скобками. Это означает, что в частном случае ни одно из трех слов, заключенных в итерационные скобки, на вход автомата не поступит и тогда первой буквой, которую принимает автомат, является буква $\mathbf{x}_1$, стоящая непосредственно за итерационными скобками. Таким образом все эти предосновные места подчинены месту с индексом 0.

Теперь найдем предосновные места, на которые распространяется индекс 1. Если автомат находится в состоянии 1, то он может принять букву $\mathbf{x}_2$, расположенную слева от места 1, так как эта буква находится в итерационных скобках и, следовательно, неоднократно может повторятся во входном слове автомата. Кроме того, в состоянии 1 автомат может принять начальные буквы других слов, расположенных в итерационных скобках, и букву $\mathbf{x}_1$, непосредственно следующую за этими скобками. Таким образом, месту с индексом 1 в данном случае подчиняются те же предосновные места, что и месту с индексом 0.Если автомат находится в состоянии 2, то он может принять только букву $\mathbf{x}_2$, расположенную справа от места с индексом 2. Поэтому индекс распространяется на единственное предосновное место, являющееся одновременно основным местом 2. Аналогично можно найти подчиненные места других основных мест.

По окончании слова, входящего в событие **S**, автомат переходит в состояние 11, после чего на вход автомата может поступить второе слово, этого события **S**, так как мы считаем, что автомат является автоматом многократного действия. Автоматами многократного действия называются такие автоматы, которые могут неоднократно принимать слова, входящие в события, представленные в автомате. В таких автоматах индекс конечного места распространяется на те же предосновные места, на которые распространяется индекс начального места, т.е. по окончании очередного слова, на вход автомата этого слова может поступить вновь.

По размеченному регулярному выражению теперь можно составить таблицу переходов автомата. Однако перед построением таблицы целесообразно уменьшить число индексов основных мест, а следовательно и число внутренних состояний автомата.

На этом первом этапе минимизации внутренних состояний можно пользоваться следующим правилом:

- Если несколько предосновных мест отмечено одинаковой совокупностью индексов и справа от этих мест записаны одинаковые буквы, то основные места, расположенные справа от этих букв можно отметить одинаковыми индексами.

В полученном нами выражении основные места 2, 4 и 7можно отметить общим индексом, так как слева от каждого из этих мест записана буква x_1, а предосновные места, предшествующие этой букве, имеют одинаковую совокупность индексов (0, 1, 3, 6, 11). Теперь с учетом этого проведем **новую разметку:**

```
 S= | { |X2| v |  X1 | X2 | v | X1 | X1 | X2 |} | X1 | X1 | X1| { | X1 | } | X2 |
     0       1          2     3         2     4     5        2     6    7         8         9
--------------------------------------------------------------------------------------------
        0        0      2         0     2     4        0     2     6        7         7
        1        1                1                    1                    8         8
        3        3                3                    3
        5        5                5                    5
        9        9                9                    9
```

Проделанную процедуру можно повторить вновь, так как в полученном выражении есть два места (4 и 6), перед которыми стоит одинаковая буква x_1, имеющая предосновное место, отмеченное одинаковым индексом 2. После этого получим **окончательную разметку:**

```
 S= | { |X2| v |  X1 | X2 | v | X1 | X1 | X2 |} | X1 | X1 | X1| { | X1 | } | X2 |
     0       1          2     3         2     4     5        2     4    6         7         8
--------------------------------------------------------------------------------------------
        0        0      2         0     2     4        0     2     4        6         6
        1        1                1                    1                    7         7
        3        3                3                    3
        5        5                5                    5
        8        8                8                    8
```

На этом первый этап минимизации закончен.

Второй этап минимизации

Составление отмеченной таблицы переходов. Составим теперь отмеченную таблицу переходов автомата. Определим вначале внутренние состояния, в которые переходит автомат из состояния 0 при подаче на его вход сигнала $\mathbf{x}_1$. Для этого найдем все предосновные места, содержащие индекс 0, справа от которых записана буква $\mathbf{x}_1$. Таких мест в выражении три. Все основные места, расположенные за этой буквой $\mathbf{x}_1$, отмечены индексом 2. Следовательно, автомат из состояния 0 под действием сигнала $\mathbf{x}_1$ переходит в состояние 2. Аналогично, сигнал $\mathbf{x}_2$ переводит автомат из состояния 0 в состояние 1, так как за предосновным, содержащим индекс 0, после буквы $\mathbf{x}_2$ расположено основное место с индексом 1. Таким же образом определяются переходы автомата их других внутренних состояний. Сигнал $\mathbf{y}_1$ выдается после поступления подряд трех букв $\mathbf{x}_1$, то есть в состоянии 6, а сигнал $\mathbf{y}_2$ – после $\mathbf{x}_2$, следующей за серией из трех и более букв, то есть в состоянии 8. В остальных случаях выдается пустая буква ***e***. Отсюда получаем следующую отмеченную таблицу переходов:

y_g	e	e	e	e	e	e	y_1	e	$\mathbf{y}_2$
$x_j\backslash a_i$	0	1	2	3	4	5	6	7	8
x_1	2	2	4	2	6	2	7	7	2
x_2	1	1	3	1	5	1	8	8	1

Из построенной таблицы видно, что из состояний 0 ,1 ,3 и 5 автомат сигналами $\mathbf{x}_1$ и $\mathbf{x}_2$ переводится в одинаковые состояния (2 и 1). Кроме того, все перечисленные состояния отмечены одинаковыми выходными сигналами. Поэтому состояния 0, 1, 3 и 5 можно объединить в одно состояние, обозначив его как $\mathbf{a}_0$. Введем также обозначения: 2 – $\mathbf{a}_1$; 4 – $\mathbf{a}_2$; 6 – $\mathbf{a}_3$; 7 – $\mathbf{a}_4$; 8 – $\mathbf{a}_5$. Тогда получим упрощенную таблицу переходов автомата. В этой таблице из состояний $\mathbf{a}_3$ и $\mathbf{a}_4$ под действием входных сигналов $\mathbf{x}_1$ и $\mathbf{x}_2$ автомат переходит в одинаковые состояния $\mathbf{a}_4$ и $\mathbf{a}_5$. Но объединять эти состояния нельзя, так как они отмечены разными выходными сигналами. По этой же причине нельзя объединять состояния $\mathbf{a}_0$ и $\mathbf{a}_5$. Объединение состояний и составляет второй этап минимизации, причем объединяются только такие состояния, которые отмечены одинаковыми выходными сигналами, и из которых под действием одинаковых входных сигналов происходит переход в одинаковые состояния. Очевидно, у таких состояний должны совпадать столбцы таблицы переходов.

y_g	e	e	e	y_1	e	y_2
$x_j\backslash a_i$	a_0	a_1	a_2	a_3	a_4	a_5
x_1	a_1	a_2	a_3	a_4	a_4	a_1
x_2	a_0	a_0	a_0	a_5	a_5	a_0

Второй пример абстрактного синтеза

Рассмотрим еще один пример абстрактного синтеза автомата. Найдем таблицу переходов автомата сравнения чисел, условия работы которого заданы регулярными выражениями:

Регулярные выражения

$$\mathbf{S}_3=|\{|\mathbf{x}_r|\}|\mathbf{x}_s|;$$

0	1	2
0	0	
1	1	
2	2	
8	8	
11	11	

$$\mathbf{S}_1=|\{|\mathbf{x}_r|\mathrm{v}|\mathbf{x}_{01}|\mathrm{v}|\mathbf{x}_{10}|\}|\mathbf{x}_{01}|\{|\mathbf{x}_r|\}|\mathbf{x}_s|;$$

0	3	4	5	6	7	8
0	0	0	0	6	6	
2	2	2	2	7	7	
3	3	3	3			
4	4	4	4			
5	5	5	5			
8	8	8	8			
11	11	11	11			

$$\mathbf{S}_2=|\{|\mathbf{x}_r|\mathrm{v}|\mathbf{x}_{01}|\mathrm{v}|\mathbf{x}_{10}|\}|\mathbf{x}_{10}|\{|\mathbf{x}_r|\}|\mathbf{x}_s|$$

0	3	4	5	9	10	11
0	0	0	0	9	9	
2	2	2	2	10	10	
3	3	3	3			
4	4	4	4			
5	5	5	5			
8	8	8	8			
11	11	11	11			

Регулярные выражения событий $\mathbf{S}_1$ и $\mathbf{S}_2$ содержат одинаковые сомножители в итерационных скобках, перед которыми расположено место с индексом 0. Поэтому в обоих выражениях основные места внутри итерационных скобок отмечены одинаковыми индексами (3, 4 и 5). Индекс конечного места каждого выражения распространяется на начальные места всех регулярных выражений, так как в автоматах многократного действия за словом любого события, например $\mathbf{S}_1$, может быть подано слово любого другого события, то есть $\mathbf{S}_1$ v $\mathbf{S}_2$ v $\mathbf{S}_3$. В размеченных выражениях можно объединить места с индексами 4, 6 и 5, 9:

$$\mathbf{S}_1=|\{|\mathbf{x}_r|\mathrm{v}|\mathbf{x}_{01}|\mathrm{v}|\mathbf{x}_{10}|\}|\mathbf{x}_{01}|\{|\mathbf{x}_r|\}|\mathbf{x}_s|;$$

0	3	4	5	4	6	7
0	0	0	0	4	4	
2	2	2	2	6	6	
3	3	3	3			
4	4	4	4			
5	5	5	5			
7	7	7	7			
9	9	9	9			

$$\mathbf{S}_2=|\{|\mathbf{x}_r|\mathrm{v}|\mathbf{x}_{01}|\mathrm{v}|\mathbf{x}_{10}|\}|\mathbf{x}_{10}|\{|\mathbf{x}_r|\}|\mathbf{x}_s|$$

0	3	4	5	5	8	9
0	0	0	0	5	5	
2	2	2	2	8	8	
3	3	3	3			
4	4	4	4			
5	5	5	5			
7	7	7	7			
9	9	9	9			

$\mathbf{S}_3=|\{\,|\mathbf{x}_r|\,\}\,|\mathbf{x}_s|;$

0	1	2
0	0	
1	1	
2	2	
7	7	
9	9	

По размеченному выражению составим отмеченную таблицу переходов.

y_g	*e*	*e*	y_3	*e*	*e*	*e*	*e*	y_1	*e*	y_2	*e*	*e*	*e*	*e*
$x_j \backslash a_i$	0	1	2	3	4	5	6	7	8	9	1 v 3	3 v 6	3v 8	*
x_r	1 v 3	1	1v 3	3	3 v 6	3 v 8	6	1 v 3	8	1 v 3	1 v 3	3 v 6	3v 8	*
x_{01}	4	*	4	4	4	4	*	3	*	4	4	4	4	*
x_{10}	5	*	5	5	5	5	*	5	*	5	5	5	5	*
x_s	2	2	2	*	7	9	7	2	9	2	2	7	9	*

При составлении таблицы следует учитывать, что для разных регулярных выражений автомат под действием одних и тех же входных сигналов переходит в разные состояния. Эти внутренние состояния будем отмечать множеством индексов основных мест. Например. В событии $\mathbf{S}_3$ переход из состояния 0 в состояние 1 происходит под действием сигнала $\mathbf{x}_r$ а в $\mathbf{S}_1$ под действием этого же сигнала из состояния 0 автомат переходит в состояние 3. Поэтому, внутреннее состояние, в которое автомат переходит под действием x_r из состояния 0, будем обозначать множеством из двух индексов 1 v 3. Аналогично получается переход из состояний 2, 7 и 9 под действием $\mathbf{x}_r$, а также переход из состояния 4 и 5 в состояния 3 v 6 и 3 v 8 соответственно под действием $\mathbf{x}_r$. При заполнении таблицы получается свободные клетки там, где переходы в автомате не определенны. Такие клетки будем отмечать звездочкой *****, которую следует рассматривать как индекс некоторого внутреннего состояния. Таблица переходов составляется не только для состояний, отмеченных индексами основных мест регулярного выражения, но и для состояний, отмеченных множеством индексов. Для заполнения колонок для таких состояний достаточно образовать дизъюнкцию таких индексов, которые расположены в колонках, отмеченных индексами, входящими в множества. Например. Для заполнения колонки 1 v 3 образуем дизъюнкцию индексов расположенных в колонках 1 и 3. Поскольку состояния 1, 3, 6 и 8 отмечены пустой буквой ***e***, то и состояния 1 v 3, 3 v 6, 3 v 8 также отмечаются буквой ***e***.

После построения таблицы проведем второй этап минимизации числа внутренних состояний автомата. Можно объединить состояния, отмеченные индексами: 0 и 1v3, 4 и 3v6, 5 и 3v8. При этом состояния отмеченные звездочкой обозначим через 10, а состояния 1v3 – 0, 3v6 – 4, 3v8 – 5.

y_g	e	e	y_3	e	e	e	e	y_1	e	y_2	e
$x_j \backslash a_i$	0	1	2	3	4	5	6	7	8	9	10
x_r	0	1	0	3	4	5	6	0	8	0	10
X_{01}	4	10	4	4	4	4	10	4	10	4	10
X_{10}	5	10	5	5	5	5	10	5	10	5	10
x_s	2	2	2	10	7	9	7	2	9	2	10

По построенной таблице проведем третий этап минимизации, исключив такие состояния, в которые автомат из нулевого состояния никогда перейти не может. В нашем случае такими состояниями являются 1, 3, 6, 8 и 10. Обозначив оставшиеся состояния 0 – $\mathbf{b}_0$, 2 – $\mathbf{b}_1$, 4 – $\mathbf{b}_2$, 5 – $\mathbf{b}_3$, 7 – $\mathbf{b}_4$, 9 – $\mathbf{b}_5$, получим окончательную таблицу переходов заданного автомата.

y_g	e	$\mathbf{y}_3$	e	e	$\mathbf{y}_1$	$\mathbf{y}_2$
$x_j \backslash a_i$	b_0	b_1	b_2	b_3	b_4	b_5
x_r	b_0	b_0	b_2	b_3	b_0	b_0
x_{01}	b_2	b_2	b_2	b_2	b_2	b_2
x_{10}	b_3	b_3	b_3	b_3	b_3	b_3
x_s	b_1	b_1	b_4	b_5	b_1	b_1

Четвёртый этап минимизации

В некоторых случаях после получения отмеченной таблицы переходов автомата возможен четвертый этап минимизации. Правда этот этап не всегда приводит к уменьшению числа состояний и часто является проверочным. Алгоритм этого этапа рассмотрим на примере.

Пусть есть автомат, заданный следующей отмеченной таблицей переходов:

y_g	$\mathbf{y}_1$	$\mathbf{y}_1$	$\mathbf{y}_1$	$\mathbf{y}_2$	$\mathbf{y}_1$	$\mathbf{y}_2$	$\mathbf{y}_2$	$\mathbf{y}_2$
$x_j \backslash a_i$	0	1	2	3	4	5	6	7
x_1	2	5	2	2	4	4	5	2
x_2	3	1	7	3	5	5	1	7
X_3	1	6	1	1	1	1	6	1

Алгоритм минимизации заключается в следующем:

1. Все внутренние состояния разбиваются на группы по числу выходных сигналов. В нашем случае есть два выходных сигнала $\mathbf{y}_1$ и $\mathbf{y}_2$ и, следовательно, будет две группы, которые мы обозначим буквами **a** и **b**.

$$\overbrace{\begin{matrix}0 & 1 & 2 & 4\\ aba & bab & aba & aba\end{matrix}}^{a} \qquad \overbrace{\begin{matrix}3 & 5 & 6 & 7\\ aba & aba & bab & aba\end{matrix}}^{b}$$

2. По таблице переходов автомата определяют, к каким группам принадлежат внутренние состояния, в которые автомат переходит из данного состояния под воздействием каждой буквы входного алфавита. Эти состояния запишем в виде последовательности букв под каждым из состояний автомата. Например, из состояния 0 автомат переходит в состояния 2, 3 и 1, которые принадлежат соответственно к следующим группам **a**, **b** и **a**. Эта последовательность букв (**aba**) и записывается под состоянием 0.
3. Проводят новое разделение внутренних состояний на группы, объединяя в каждой группе состояния, отмеченные одинаковой последовательностью букв. В нашем случае каждая из двух групп распадается на две группы, по числу различных последовательностей букв:

$$\overbrace{\begin{matrix}0 & 2 & 4\\ \mathbf{acb} & \mathbf{acb} & \mathbf{acb}\end{matrix}}^{\mathbf{a}} \qquad \overbrace{\begin{matrix}1\\ \mathbf{cbd}\end{matrix}}^{\mathbf{b}}$$

$$\overbrace{\begin{matrix}3 & 5 & 7\\ \mathbf{acb} & \mathbf{acb} & \mathbf{acb}\end{matrix}}^{\mathbf{c}} \qquad \overbrace{\begin{matrix}6\\ \mathbf{cbd}\end{matrix}}^{\mathbf{d}}$$

4. Пользуясь таблицей переходов автомата, вновь отмечают каждое состояние последовательностью букв. Разделение состояний на новые группы продолжают до тех пор, пока новые группы состояний

появляться не будут. В нашем случае, минимизация заканчивается на втором шаге, так как все состояния, входящие в группы **a** и **c** отмечены одинаковыми последовательностями букв, а группа **b** и **d** содержат только по одному состоянию.

Все состояния, входящие в каждую из этих групп, можно заменить одним состоянием той же группы. Взяв в качестве представителей групп состояния 0, 1, 3 и 6 и обозначив их символами $\mathbf{a}_0$, $\mathbf{a}_1$, $\mathbf{a}_2$ и $\mathbf{a}_3$ соответственно, получим следующую таблицу переходов с минимальным числом внутренних состояний.

$\mathbf{y}_g$	$\mathbf{y}_1$	$\mathbf{y}_1$	$\mathbf{y}_2$	$\mathbf{y}_2$
$x_j \backslash a_i$	a_0	a_1	a_2	a_3
$\mathbf{x}_1$	a_0	a_2	a_0	a_2
$\mathbf{x}_2$	a_2	a_1	a_2	a_1
$\mathbf{x}_3$	a_1	a_3	a_1	a_3

Для построения автомата Мили, воспользуемся рассмотренным ранее алгоритмом, для чего в каждую клетку >совмещенной таблицы переходов и выходов запишем значения выходного сигнала, которым отмечено, находящееся здесь состояние.

$\mathbf{x_j \backslash a_i}$	$\mathbf{a_0}$	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{a_3}$
$\mathbf{x_1}$	$\mathbf{a_0/y_1}$	$\mathbf{a_2/y_2}$	$\mathbf{a_0/y_1}$	$\mathbf{a_2/y_2}$
$\mathbf{x_2}$	$\mathbf{a_2/y_2}$	$\mathbf{a_1/y_1}$	$\mathbf{a_2/y_2}$	$\mathbf{a_1/y_1}$
$\mathbf{x_3}$	$\mathbf{a_1/y_1}$	$\mathbf{a_3/y_2}$	$\mathbf{a_1/y_1}$	$\mathbf{a_3/y_2}$

В полученной таблице колонки, помеченные состояниями $\mathbf{a}_0$ и $\mathbf{a}_2$, $\mathbf{a}_1$ и $\mathbf{a}_3$ идентичны, что позволяет при минимизации исключить состояния $\mathbf{a}_2$ и $\mathbf{a}_3$. В результате получаем таблицу переходов и выходов автомата Мили имеющего два состояния $\mathbf{a}_0$ и $\mathbf{a}_1$.

$\mathbf{x_j \backslash a_i}$	$\mathbf{a_0}$	$\mathbf{a_1}$
$\mathbf{x_1}$	$\mathbf{a_0/y_1}$	$\mathbf{a_0/y_2}$
$\mathbf{x_2}$	$\mathbf{a_0/y_2}$	$\mathbf{a_1/y_1}$
$\mathbf{x_3}$	$\mathbf{a_1/y_1}$	$\mathbf{a_1/y_2}$

Занятие 8. Эквивалентные автоматы. Алгоритм перехода от произвольного автомата Мили к эквивалентному ему автомату Мура и обратно.

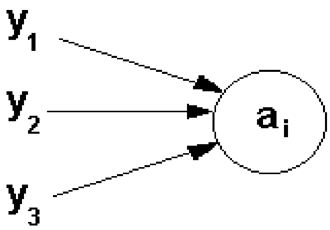
Два автомата S_a и S_b с одинаковыми входными и выходными алфавитами называются ***эквивалентными***.

Оказывается, что для любого автомата Мили существует эквивалентный ему автомат Мура, и, обратно, для любого автомата Мура существует эквивалентный ему автомат Мили.

Рассмотрим алгоритм перехода от произвольного конечного автомата Мили к эквивалентному ему автомату Мура.

Пусть дан конечный автомат Мили $S_a=\{A_a,X_a,Y_a,\delta_a,\lambda_a\}$, имеющий множество состояний $A_a=\{a_0,a_1,\ldots,a_i,\ldots,a_n\}$, множество входных и выходных сигналов $X_a=\{x_1,x_2,\ldots,x_j,\ldots,x_m\}$ и $Y=\{y_1,y_2,\ldots,y_g,\ldots,y_k\}$, а также функции переходов $\delta_a(a,x)$ и выходов $\lambda_a(a,x)$.

Требуется построить эквивалентный ему автомат Мура. $S_b=\{A_b,X_b,Y_b,\delta_b,\lambda_b\}$, у которого $X_b=X_a$, $Y_b=Y_a$, так как множества входных и выходных сигналов у эквивалентных автоматов должны совпадать. Для определения множества состояний A_b автомата Мура образуем всевозможные пары вида (a_i,y_g), где y_g – выходной сигнал, приписанный к дуге входящей в состояние a_i. Например, для вершины a_i имеем пары (a_i,y_1), (a_i,y_2), (a_i,y_3).

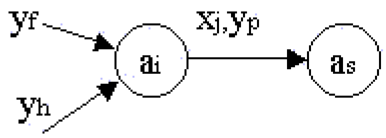


Если такие пары мы образуем для всех вершин, то получим множество пар, которое является множеством состояний автомата Мура:

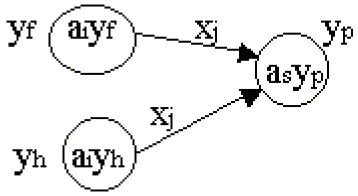
$A_b=\{(a_0,y_1), (a_0,y_2),\ldots,(a_n,y_k)\}=\{b_1,b_2,\ldots,b_n\}$, где $b_l=(a_i,y_g)$.

Функции выходов λ_b и переходов δ_b определим следующим образом. Каждому состоянию автомата Мура, представляющему собой пару вида (a_i,y_g) поставим в соответствие выходной сигнал y_g, то есть функция выходов равна $y_g=\lambda_b[(a_i,y_g)] = \lambda_b[b_l]$. Если в автомате Мили S_a был переход $\delta_a(a_i,x_j)=a_s$ и при этом выдавался выходной сигнал $\lambda_a(a_i,x_j)=y_p$, то в эквивалентном автомате Мура будет переход из множества состояний (a_i,y_g), где g принадлежит G, G – множество номеров выходных сигналов, приписанных к входящей a_i дуге, в состояние (a_s,y_p) под действием входного сигнала x_j.

Автомат Мили (фрагмент)



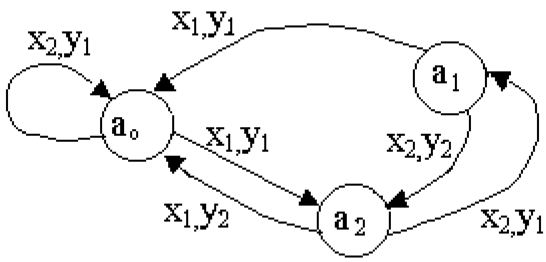
Автомат Мура эквивалентный автомату Мили



Автомат Мили имеет два состояния, а автомат Мура три : (a_i,y_f), (a_i,y_h), (a_i,y_p). Если автомат Мили был в состоянии a_i и пришел входной сигнал x_j, то должен выработаться выходной сигнал y_p. Поэтому в автомате Мура из состояний, порождаемых a_i, то есть из состояний (a_i,y_f) и (a_i,y_h) при поступлении x_j переход должен идти в состояние, отмеченное выходным сигналом y_p, то есть в (a_s, y_p). В качестве начального состояния автомата Мура можно взять любое состояние из множества (a_0, y_r).

Преобразование автомата Мили в автомат Мура

Рассмотрим пример: Пусть необходимо преобразовать автомат Мили, в автомат Мура. Граф автомата Мили:



В автомате Мили $\mathbf{X}_a = \{\mathbf{x}_1, \mathbf{x}_2\}$, $\mathbf{Y}_a = \{\mathbf{y}_1,\mathbf{y}_2\}$, $\mathbf{A}_a = \{\mathbf{a}_0, \mathbf{a}_1,\mathbf{a}_2\}$.

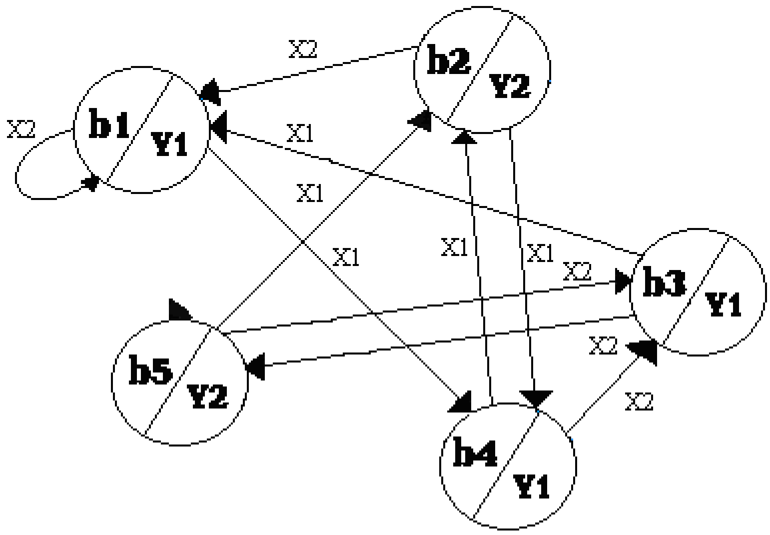
В эквивалентном автомате Мура$\mathbf{X}_b = \mathbf{X}_a = \{\mathbf{x}_1, \mathbf{x}_2\}$, $\mathbf{Y}_b = \mathbf{Y}_a = \{\mathbf{y}_1, \mathbf{y}_2\}$.

Построим множество состояний $\mathbf{A}_b$ автомата Мура, для чего найдем множества пар, порождаемых каждым состоянием автомата $\mathbf{S}_a$.

Состояние	Порождаемые пары
$\mathbf{a}_0$	$\{(\mathbf{a}_0, \mathbf{y}_1), (\mathbf{a}_0, \mathbf{y}_2)\}=\{\mathbf{b}_1, \mathbf{b}_2\}$
$\mathbf{a}_1$	$\{(\mathbf{a}_1, \mathbf{y}_1)\}=\{\mathbf{b}_3\}$
$\mathbf{a}_2$	$\{(\mathbf{a}_2, \mathbf{y}_1), (\mathbf{a}_2, \mathbf{y}_2)\}=\{\mathbf{b}_4, \mathbf{b}_5\}$

Отсюда имеем множества $\mathbf{A}_b$ состояний автомата Мура $\mathbf{A}_b = \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4, \mathbf{b}_5\}$. Для нахождения функции выходов $\boldsymbol{\lambda}_b$ с каждым состоянием, представляющим собой пару вида ($\mathbf{a}_i$, $\mathbf{y}_g$), отождествим выходной сигнал, являющийся вторым элементом этой пары. В результате имеем: $\lambda_b(\mathbf{b}_1) = \lambda_b(\mathbf{b}_3) = \lambda_b(\mathbf{b}_4) = \mathbf{y}_1$; $\lambda_b(\mathbf{b}_2) = \lambda_b(\mathbf{b}_5) = \mathbf{y}_2$.

Построим функцию переходов δ_b. Так как в автомате $\mathbf{S}_a$ из состояния $\mathbf{a}_0$ есть переход под действием сигнала $\mathbf{x}_1$ в состояние $\mathbf{a}_2$ с выдачей $\mathbf{y}_1$,то из множества состояний $\{\mathbf{b}_1, \mathbf{b}_2\}$, порождаемых $\mathbf{a}_0$, в автомате $\mathbf{S}_b$ должен быть переход в состояние ($\mathbf{a}_2$, $\mathbf{y}_1$) = $\mathbf{b}_4$ под действием сигнала $\mathbf{x}_1$.Аналогично, из $\{\mathbf{b}_1, \mathbf{b}_2\}$ под действием $\mathbf{x}_2$ должен быть переход в ($\mathbf{a}_0$, $\mathbf{y}_1$) = $\mathbf{b}_1$. Из ($\mathbf{a}_1$, $\mathbf{y}_1$) = $\mathbf{b}_3$ под действием $\mathbf{x}_1$ переход в ($\mathbf{a}_0$, $\mathbf{y}_1$) = $\mathbf{b}_1$, а под действием $\mathbf{x}_2$ – в ($\mathbf{a}_2$, $\mathbf{y}_2$) = $\mathbf{b}_5$. Наконец из состояний $\{(\mathbf{a}_2, \mathbf{y}_1), (\mathbf{a}_2, \mathbf{y}_2)\} = \{\mathbf{b}_4, \mathbf{b}_5\}$ под действием $\mathbf{x}_1$ в ($\mathbf{a}_0$, $\mathbf{y}_2$) = $\mathbf{b}_2$, а под действием $\mathbf{x}_2$ – в ($\mathbf{a}_1$, $\mathbf{y}_1$) = $\mathbf{b}_3$. В результате имеем граф и таблицу переходов эквивалентного автомата Мура:



y_g	y_1	y_2	y_1	y_1	y_2
$x_j \backslash b_i$	b_1	b_2	b_3	b_4	b_5
x_1	b_4	b_4	b_1	b_2	b_2
x_2	b_1	b_1	b_5	b_3	b_3

В качестве начального состояния автомата $\mathbf{S}_b$ можно взять любое из состояний $\mathbf{b}_1$ или $\mathbf{b}_2$, так как оба порождены состоянием a_0 автомата $\mathbf{S}_a$.

Переход от автомата Мура к автомату Мили

Обратная задача, то есть переход от автомата Мура к автомату Мили решается чрезвычайно просто. Пусть дан автомат Мура $\mathbf{S}_b = \{\mathbf{A}_b, \mathbf{X}_b, \mathbf{Y}_b, \delta_b, \lambda_b\}$. Необходимо построить эквивалентный ему автомат Мили $\mathbf{S}_a = \{\mathbf{A}_a, \mathbf{X}_a, \mathbf{Y}_a, \delta_a, \lambda_a\}$.

По определению эквивалентности имеем $\mathbf{X}_a = \mathbf{X}_b$; $\mathbf{Y}_a = \mathbf{Y}_b$. Кроме того, $\mathbf{A}_a = \mathbf{A}_b$, $\boldsymbol{\delta}_a = \boldsymbol{\delta}_b$. Остается только построить функцию выходов. Если в автомате Мура $\boldsymbol{\delta}_b(\mathbf{a}_i, \mathbf{x}_j) = \mathbf{a}_s$, а$\boldsymbol{\lambda}_b(\mathbf{a}_s) = y_g$, то в автомате Мили $\boldsymbol{\lambda}_a(\mathbf{a}_i, \mathbf{x}_j) = \mathbf{y}_g$. Другими словами $\boldsymbol{\lambda}_a(\mathbf{a}_i, \mathbf{x}_j) = \boldsymbol{\lambda}_b(\boldsymbol{\delta}_b(\mathbf{a}_i, \mathbf{x}_j))$. Таким образом таблица переходов автоматов Мили и Мура совпадают. А таблица выходов эквивалентного автомата Мили строится так, что в каждую клетку таблицы записывается выходной сигнал, которым отмечено состояние, расположенное в данной клетке.

Пример. Пусть дан автомат Мура:

$x_j \backslash y_i$	y_1	y_1	y_3	y_2	y_3
$\mathbf{x}_j \backslash a_i$	a_0	a_1	a_2	a_3	a_4
x_1	a_1	a_4	a_4	a_2	a_2
x_2	a_3	a_1	a_1	a_0	a_0

Тогда эквивалентный ему автомат Мили имеет следующую совмещенную таблицу переходов и выходов.

$\mathbf{x}_j \backslash a_i$	a_0	a_1	a_2	a_3	a_4
x_1	a_1/y_1	a_4/y_3	a_4/y_3	a_2/y_3	a_2/y_3
x_2	a_3/y_2	a_1/y_1	a_1/y_1	a_0/y_1	a_0/y_1

Занятие 9. Вероятностные автоматы

Детерминированные автоматы **S** мы задавали совокупностью из пяти объектов: $\mathbf{S(A, X, Y, \delta, \lambda)}$, где $\mathbf{A} = \{\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_M\}$– множество внутренних состояний автомата, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x_F}\}$ – множество входных сигналов (входной алфавит), $\mathbf{x}_i$ – буква входного алфавита, $\mathbf{Y = \{y_1, y_2, \ldots, y_G\}}$ – множество выходных сигналов (выходной алфавит),

δ - функция переходов, обеспечивающая однозначный переход автомата в состояние $\mathbf{a_s}$ из состояния $\mathbf{a_m}$ под действием входного сигнала $\mathbf{x_f}$, то есть $\mathbf{a_s} = \delta\,[\mathbf{a_m}, \mathbf{x_f}]$,

λ - функция выходов, определяющая однозначное значение выходного сигнала $\mathbf{y_g}$ в зависимости от состояния автомата $\mathbf{a_m}$ и входного сигнала $\mathbf{x_f}$, т.е. $\mathbf{y_g} = \lambda[\mathbf{a_m}, \mathbf{x_f}]$.

В работе Поспелова Д.А. «Вероятностные автоматы» рассмотрена более общую модель автомата, а именно: зная состояние автомата $\mathbf{a_m}$ и входной сигнал $\mathbf{x_f}$, мы не можем с вероятностью, равной 1 сказать, в каком состоянии окажется автомат в следующий момент времени, а также какой выходной сигнал он в этом случае вырабатывает. Однако мы можем указать вероятности наступления соответствующего события, а именно: зная состояние $\mathbf{a_m}$ и входной сигнал $\mathbf{x_f}$, мы можем указать вероятности перехода автомата в состояния $\{\mathbf{a_0, a_1, \ldots, a_m, \ldots, a_M}\}$, а также вероятности появления выходных сигналов $\{\mathbf{y_1, y_2, \ldots, y_g, \ldots, y_G}\}$, т.е. задаем закон распределения вероятностей. Законы распределения задаются в виде следующих таблиц:

a_m,x_f	a_0	a_1	…	a_m	…	a_M
a_0,x_1	P_{010}	P_{011}	…	P_{01m}	…	P_{01M}
a_0,x_2	P_{020}	P_{021}	…	P_{02m}	…	P_{02M}
…	…	…	…	…	…	…
a_m,x_F	P_{0F0}	P_{0F1}	…	P_{0Fm}	…	P_{0FM}
a_1,x_1	P_{110}	P_{111}	…	P_{11m}	…	P_{11M}
…	…	…	…	…	…	…
a_m,x_f	P_{mf0}	P_{mf1}	…	P_{mfm}	…	P_{mfM}
…	…	…	…	…	…	…
a_M,x_F	P_{MF0}	P_{MF1}	…	P_{MFm}	…	P_{MFM}

a_m,x_f	y_1	y_2	…	y_y	…	a_M
a_0,x_1	q_{011}	q_{012}	…	q_{01y}	…	q_{01G}
a_0,x_2	q_{021}	q_{022}	…	q_{02y}	…	q_{02G}
…	…	…	…	…	…	…
a_m,x_F	q_{0F1}	q_{0F2}	…	q_{0Fy}	…	q_{0FG}
a_1,x_1	q_{111}	q_{112}	…	q_{11y}	…	q_{11G}
…	…	…	…	…	…	…
a_m,x_f	q_{mf1}	q_{mf2}	…	q_{mfy}	…	q_{mfG}
…	…	…	…	…	…	…
a_M,x_F	q_{MF1}	q_{MF2}	…	q_{MFy}	…	q_{MFG}

Т.е. в каждом случае имеем закон распределения, заданный в виде гистограмм. Очевидно, т.к. автомат обязательно перейдет в одно из состояний, то $\sum_{i=0}^{M} p_{mfi} = 1, \quad \sum_{j=0}^{G} q_{mfj} = 1,$ где $0 \le p_{mfi} \le 1,\ 0 \le q_{mfj} \le 1.$

Автоматы, в которых зная состояние автомата $\mathbf{a_m}$ и входной сигнал $\mathbf{x_f}$, мы можем указать лишь вероятности перехода в новое состояние и вероятности появления выходных сигналов, т.е. законы распределения, называются вероятностными автоматами (ВА).

По аналогии с детерминированными автоматами, Можно определить ВА Мили и Мура. ВА, у которых вероятности появления выходных сигналов (закон распределения) зависят лишь от состояний автомата, но не зависят от входных сигналов, называются ВА Мура. Если же вероятности появления выходных сигналов (закон распределения) зависят как от состояний автомата, так и от входных сигналов, имеем автомат Мили.

Рассмотрим некоторые частные случаи вероятностных автоматов. Может быть, что выходные сигналы автомата определяются детерминировано, а переходы автомата – случайно. Такие автоматы называются Y – детерминированными вероятностными автоматами. Если состояния определяются детерминировано, то имеем A- детерминированный вероятностный автомат.

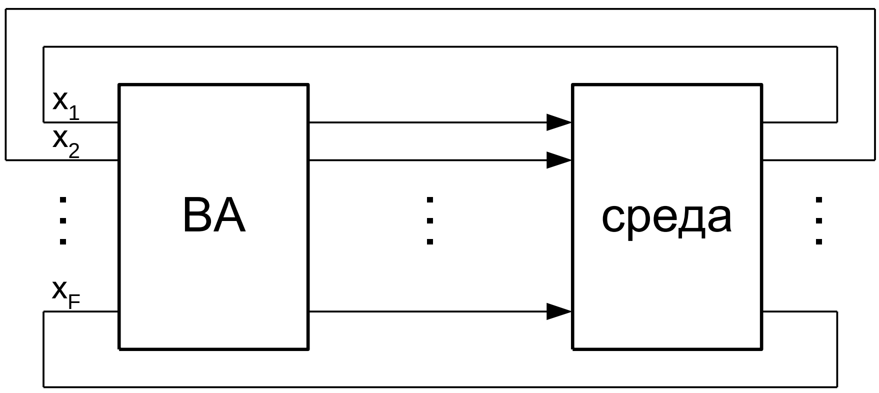
Если в процессе функционирования автомата законы распределения вероятностей появления выходных сигналов и вероятности перехода автомата в новые состояния не меняются во времени, то такие ВА называются **ВА с постоянной структурой**.

Очевидно, можно рассмотреть общий случай, когда эти законы распределения зависят от времени. Такие автоматы называются ВА с переменной структурой.

ВА с переменной структурой в каждый фиксированный такт работы является некоторым обычным ВА, но в период между тактами ВА может изменять свои матрицы переходных вероятностей или таблицы выходных вероятностей, или и то и другое вместе.

Часто при построении ВА изменение вероятностей производят по некоторому закону, причем закон зависит от истории функционирования автомата (т.е. зависит от входных сигналов, поданных на него и от выходных сигналов, т.е. реакции автомата). Такие ВА с переменной структурой называются автоматами компенсирующего типа. Их разработке и уделяется основное внимание.

В этом случае можно сказать, что ВА работает в некоторой среде, в которую он выдает выходные сигналы и из которой он получает входные.



Входные сигналы условно можно разделить на поощрения («нештрафы») и наказания («штрафы»). При этом в зависимости от выходного сигнала на вход подается поощрение или штраф. Если в зависимости от этих сигналов менять вероятности перехода автомата из одного состояния в другое, то оказывается, что с течением времени автомат перестраивается таким образом, что он начинает с большой вероятностью получать сигналы поощрения, т.е. он в некотором смысле приспосабливается к той среде, в которой он находится.

Проблема организации целесообразного поведения автомата в случайной среде тесно связана со способом изменения вероятностей перехода автомата. Возможно изменение вероятностей перехода автомата по строкам и по столбцам.

Рассмотрим У-детерминированный вероятностный автомат. Пусть автомат в некоторый момент времени t находится в состоянии $\mathbf{a_m}$**,** выдал соответствующий этому состоянию выходной сигнал $\mathbf{y_g}$ и получил на вход сигнал поощрения. Тогда вероятность $p_{\mathbf{mm}}$ перехода из состоянии $\mathbf{a_m}$ в состояние $\mathbf{a_m}$ увеличиваются на некоторую величину Δp, а все остальные вероятности в строке уменьшаются на Δp/M. Если же автомат получает сигнал штрафа, то вероятность $p_{\mathbf{mm}}$ перехода из состоянии $\mathbf{a_m}$ в состояние $\mathbf{a_m}$ уменьшаются на некоторую величину Δp, а все остальные вероятности в строке на увеличиваются на Δp/M, чтобы сумма вероятностей осталась равной 1.

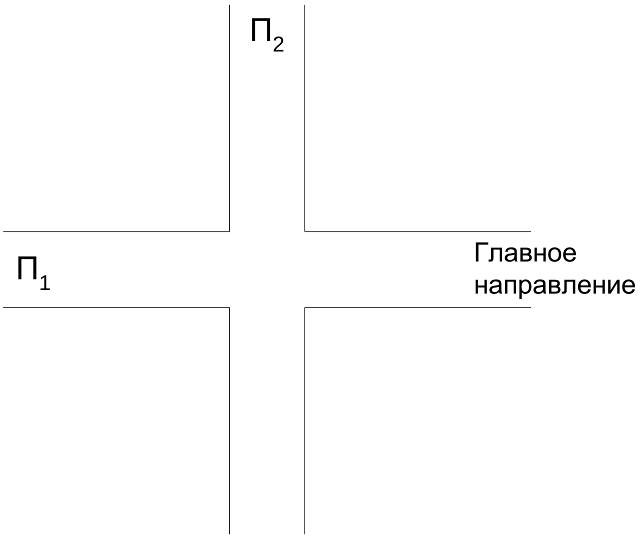
Возможен и другой принцип изменения вероятностей, при котором происходит учет предыстории поведения автомата. Если автомат в момент времени t перешел из состояния $\mathbf{a_m}$ в состояние $\mathbf{a_к}$ и в момент времени t+1 получил сигнал «штраф», то вероятность $p_{\mathbf{mк}}$ заменяется на α $p_{\mathbf{mк}}$, где коэффициент α больше 0 и меньше 1, а все остальные вероятности в строке изменяются на величину $(1-\alpha)p_{mk}/M$. Если же получил сигнал «нештраф», то вероятность $p_{\mathbf{mк}}$ величину $(1-\alpha)+\alpha p_{mk}$, а все остальные уменьшаются на величину $(1-\alpha)(1-p_{mk})/M$.

Можно менять вероятности в матрице перехода не только по строкам, но и по столбцам. Например, возможен следующий алгоритм. Если в момент времени t под влиянием входного сигнала $\mathbf{x_f}$ автомат перешел в состояние $\mathbf{a_m}$ и в момент времени t+1 получил сигнал «штраф», то независимо от того, из какого состояния он перешел, все элементы m-го столбца в матрице переходов заменяются на ($p_{\mathbf{mm}}$ – Δp) или α $p_{\mathbf{mm}}$, а все остальные вероятности изменяются аналогично тому, как это происходило при изменении вероятностей по строкам.

Подобные автоматы уже находят применение при управлении в сложных системах и дают больший эффект там, где раньше работали детерминированные автоматы.

Рассмотрим пример, который приводит Д.А. Поспелов – регулирование движения через автомобильный перекресток. Обычно (мы с вами всегда сталкиваемся именно с таким светофором) задают жесткий режим переключения светофоров, при котором длительность включенных сигналов (красного и зеленого) – постоянны. Однако, как показывает практика работы таких светофоров, решение задачи получается мало эффективным, поскольку предполагается, что потоки машин постоянные, стационарные.

Можно установить датчики на перекрестке, которые бы подсчитывали число машин (очередь), возникающее в данном направлении при красном свете светофора. Пусть на перекрестке стоит ВА компенсирующего типа, который имеет 2 состояния: включен красный свет вдоль главного направления и включен зеленый свет. Каждому состоянию однозначно соответствует выходной сигнал, т.е. автомат У-детерминированный. С датчиков поступают сигналы штрафа и нештрафа.



Матрица переходов выглядит следующим образом: $\begin{Vmatrix} p_{кк} & p_{кз} \\ p_{зк} & pзз \end{Vmatrix}$.

Пусть в начале все эти вероятности равны 0,5 и на главном направлении скопилось $П_1$ машин, а на другом – $П_2$. На вход автомата поступает сигнал $\Delta П = П_1 - П_2$. Пусть $\Delta П > 0$, т.е. на главном направлении больше машин. Тогда если автомат в момент времени t находился в состоянии зеленом, перешел в состояние зеленый и получил сигнал нештраф, то вероятность $p_{зз}(t+1)$ увеличивается, а вероятность $p_{зк}(t+1)$ уменьшается:

$$p_{зк}(t+1) = \alpha\ p_{зк}(t),$$

а вероятность $$p_{зз}(t+1) = 1 - p_{зк}(t+1).$$

Тем самым увеличивается вероятность состояния зеленое вдоль главного направления, т.е. автомат подстраивается под обстановку. Заметим, что если потоки одинаковы, то оптимальной является следующая матрица переходов: $\begin{Vmatrix} 0 & 1 \\ 1 & 0 \end{Vmatrix}$.