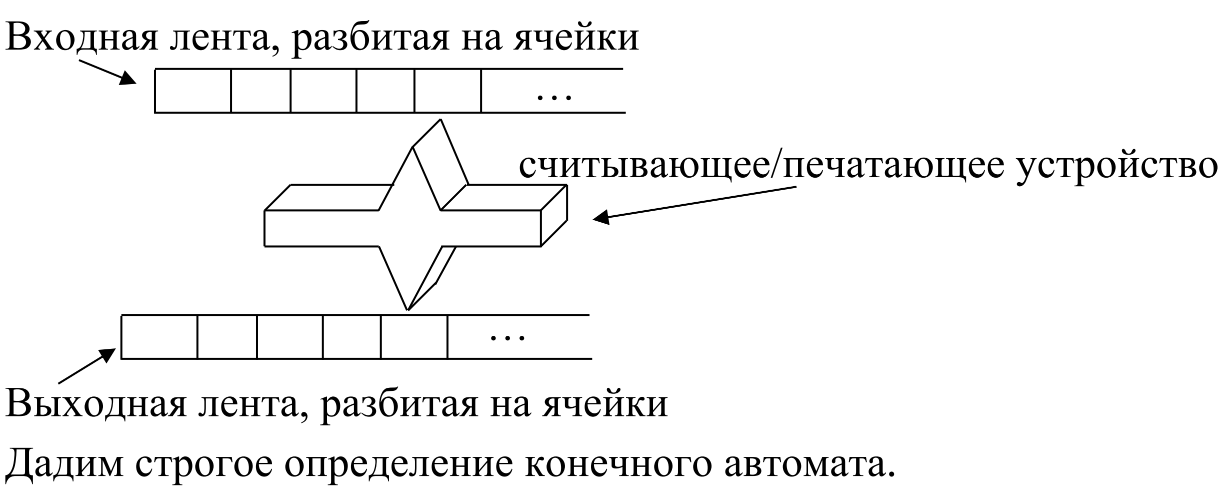
Конечные автоматы.

Представим две бесконечные параллельные ленты, неограниченные вправо - входная лента (верхняя), разбитая на ячейки, в каждой ячейке которой может быть записан один символ некоторого входного алфавита и выходная лента (нижняя), разбитая на ячейки, в каждой ячейке которой может быть записан один символ некоторого выходного алфавита.

Пусть у нас имеется также некоторое устройство, способное перемещаться между лентами ленты вправо, за один шаг проходя расстояние, равное длине ячейки.

Устройство в каждый момент времени находится в некотором внутреннем состоянии, число которых конечно и своё для каждого устройства.

Устройство работает дискретно: за один шаг оно считывает символ из ячейки входной ленты, меняет своё внутреннее состояние и записывает символ в соответствующую ячейку выходной ленты, затем сдвигается вправо. Переход в другое состояние и печатаемый символ на выходной ленте зависит от того, в каком состоянии находилось устройство в предыдущий момент времени и от того, какой символ был считан из ячейки входной ленты в настоящий момент времени. Конечный автомат заканчивает работу, когда всё слово, записанное на входной ленте, заканчивается.



Дадим строгое определение конечного автомата.

Конечным автоматом называется пятёрка $S=(A,Q,V,\delta,\lambda)$, где $A=\{a_1,a_2,\ldots,a_m\}$ – *входной алфавит*, то есть множество символов, которые могут записываться в ячейках входной ленты, $V=\{v_1,v_2,...,v_k\}$– *выходной алфавит*, то есть множество символов, которые могут записываться в ячейках выходной ленты, $Q=\{q_1,q_2,...,q_n\}$– *множество внутренних состояний*, в которых может находиться конечный автомат,

$\lambda: A\times Q\to V$ - *функция выхода*, показывающая, какой символ должно напечатать в настоящий момент времени на выходной ленте считывающее устройство, в зависимости от того, какой символ прочитан на ленте в настоящий момент времени на входной ленте и в каком состоянии находился конечный автомат в предыдущий момент времени.

$\delta: A\times Q\to Q$ - *функция перехода*, показывающая, в какое новое состояние должен перейти конечный автомат в зависимости от того, какой символ прочитан на входной ленте в настоящий момент времени и в каком состоянии находился конечный автомат в предыдущий момент времени.

Если конечный автомат всегда начинает работу из одного и того же состояния (обычно оно обозначается, как q_1), то такой автомат называется *инициальным*.

Если конечный автомат может начинать работу из любого состояния, то такой автомат называется *неинициальным*.

Через A^*обозначим множество слов, составленных из символов входного алфавита. Если $\alpha\in A^*$, то будем называть *входным словом*, а слово, полученное на выходной ленте в результате работы автомата, *выходным словом*.

Выходное слово, полученное в результате работы автомата, запущенного из состояния q_i над словом $\alpha\in A^*$, будем обозначать $S(\alpha,q_i)$.

Пусть $x(t)$ - функция, показывающая значение входного символа в момент времени t, $y(t)$ - функция, показывающая значение выходного символа в момент времени t, а $q(t)$ - состояние автомата в момент времени t, тогда можно записать:

$$\begin{cases} q(t)=\delta(x(t),q(t-1)); \\ y(t)=\lambda(x(t),q(t-1)). \end{cases} \qquad \textbf{(1)}$$

Уравнения системы (1) называются *канонескими уравнениями* конечного автомата.

Способы задания автоматов.

1. Таблица состояний автомата.

Так как конечный автомат описывается с помощью функций перехода и выхода, заданных на одних и тех же конечных наборах переменных, то эти функции сводят в одну таблицу, разделяя значения функции перехода и выхода запятой. Эта таблица *называется таблицей состояний* автомата.

Пример: Пусть $A=\{a,b\}$, $V=\{0,1\}$, $Q=\{q_1,q_2,q_3\}$, таблица состояний:

A \ Q	q_1	q_2	q_3
a	$q_1,0$	$q_1,1$	$q_1,1$
b	$q_1,1$	$q_2,0$	$q_2,0$

Считая автомат неинициальным, покажем работу автомата, запущенного из состояния q_2 над входным словом $baab$:

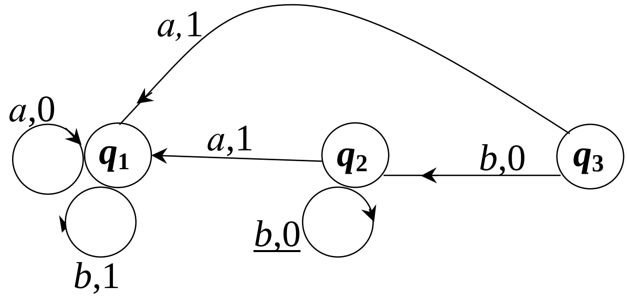
	b	a	a	b	- входное слово
q_2	q_2	q_1	q_1	q_1	
	0	1	0	1	- выходное слово

2. Диаграмма состояний автомата.

При изображении конечного автомата диаграммой состояний каждому состоянию соответствует вершина графа.

Если $\delta(a_j,q_i)=q_k$, $\lambda(a_j,q_i)=v_l$, то из q_i в q_k ведёт дуга, на которой написаны a_j и v_l.

Пример 1: Построить диаграмму состояний автомата, заданного в предыдущем примере таблицей состояний:



Состояние q_i называется *достижимым* из состояния q_k, если существует входное слово α, такое, что автомат, запущенный из состояния q_k, после работы над словом α попадает в состояние q_i.

В примере 1 состояние q_1 достижимо из состояний q_2 и q_3, так как $\delta(a, q_2) = q_1$ и $\delta(a, q_3) = q_1$.

Конечный автомат называется *сильно связным*, если любое его состояние достижимо из любого другого его состояния.

Автомат примера 1 не является сильно связным, так как, например, состояние q_2 не достижимо из состояния q_1.

Критерий сильной связности автомата.

Для того, чтобы автомат был сильно связным, необходимо и достаточно, чтобы в его диаграмме состояний существовал бы *циклический маршрут*, то есть маршрут с совпадающими начальной и конечной вершиной, в который входили бы все состояния автомата.

Необходимость. Пусть автомат $S = (A, Q, V, \delta, \lambda)$, где $Q = \{q_1, q_2, \ldots, q_n\}$ – сильно связный. Так как состояние q_2 достижимо из состояния q_1, существует входное слово α_1, при работе над которым автомат, запущенный из состояния q_1, перейдёт в состояние q_2, то есть найдётся маршрут из q_1 в q_2. Аналогично, так как состояние q_3 достижимо из состояния q_2, существует входное слово α_2, при работе над которым автомат, запущенный из состояния q_2, перейдёт в состояние q_3, то есть найдётся маршрут из q_2 в

q_3. Продолжая рассуждать аналогично, получим, что существует входное слово α_n, при работе над которым автомат, запущенный из состояния q_n, перейдёт в состояние q_1, то есть найдётся маршрут из q_n в q_1. Теперь, если мы рассмотрим входное слово $\alpha_1\alpha_2\ldots\alpha_n$ и запустим автомат из состояния q_1,то мы получим маршрут $q_1 \to \ldots \to q_2 \to \ldots \to q_3 \to \ldots \to q_n \to \ldots \to q_1$, в котором встретятся все состояния автомата S.

Достаточность. Пусть в диаграмме состояний автомата S имеется циклический маршрут, в который входят все состояния автомата. Рассмотрим произвольные состояния автомата q_i и q_k. Выберем вершину графа, соответствующую состоянию q_i и будем двигаться по циклическому маршруту. Так все состояния автомата попали в этот маршрут, на некотором шаге нам встретится состояние q_k.

Таким образом, для произвольных состояний q_i и q_k имеем, что q_k достижима из q_i, то есть автомат S является сильно связным. ■

Эквивалентные автоматы.

Пусть даны автоматы $S = (A, Q, V, \delta, \lambda)$ и $S' = (A, Q', V, \delta', \lambda')$. Два инициальных автомата S и S' называются *эквивалентными*, если для любого входного слова $\alpha \in A^*$выходные слова, полученные в результате работы автоматов S и S' над словом α, будут совпадать. Если инициальные автоматы S и S' эквивалентны, будем писать $S \sim S'$. Это определение можно записать в виде формулы:

$$S \sim S' \Leftrightarrow \forall_{\alpha \in A^*} S(\alpha, q_1) = S'(\alpha, q_1') \qquad (2)$$

Два неинициальных автомата S и S' называются *эквивалентными*, если для любого состояния q автомата S найдётся состояние q'автомата S' так, что для любого входного слова $\alpha \in A^*$выходные слова, полученные в результате работы автоматов S и S' над словом α, запущенные соответственно из состояний q и q' будут совпадать.

И обратно, если для любого состояния q' автомата S' найдётся состояние q автомата S так, что для любого входного слова $\alpha \in A^*$ выходные слова, полученные в результате работы автоматов S и S' над словом α, запущенные соответственно из состояний q и q' будут совпадать.

Это определение можно записать в виде формулы:

$$S \sim S' \Leftrightarrow \forall_{q\in Q}\exists_{q'\in Q'}\forall_{\alpha\in A^*}S(\alpha,q)=S'(\alpha,q')\wedge$$

$$\wedge\forall_{q'\in Q'}\exists_{q\in Q}\forall_{\alpha\in A^*}S(\alpha,q)=S'(\alpha,q'). \qquad \textbf{(3)}$$

В этом случае состояния q и q' будем называть *эквивалентными* и писать $q \sim q'$.

В частном случае, если $S' = S$, будем говорить об *эквивалентных состояниях одного и того же автомата*.

Эквивалентные конечные автоматы могут иметь различное число внутренних состояний, и возникает естественная задача: среди всех автоматов, эквивалентных данному, отыскать наиболее простой автомат, то есть автомат с наименьшим числом внутренних состояний.

Пусть дан автомат S. Автомат $S_{\min}$ называется *минимальным по отношению к автомату* S, если $S_{\min} \sim S$, причём $S_{\min}$ содержит наименьшее число внутренних состояний среди всех автоматов, эквивалентных автомату S.

Рассмотрим процесс минимизации на примере автомата S, заданного таблицей состояний:

A \ Q	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
a	$5, x$	$3, y$	$7, x$	$5, x$	$6, y$	$9, x$	$6, y$	$4, y$	$1, y$
b	$2, y$	$1, x$	$9, y$	$8, y$	$7, x$	$5, y$	$5, x$	$3, x$	$4, x$

Для построения минимального эквивалентного автомата выявим все пары эквивалентных состояний данного автомата.

Будем выявлять все пары неэквивалентных состояний, и тогда все пары состояний, не являющиеся неэквивалентными, будут эквивалентны. Для этого построим треугольную таблицу, где каждой паре различных состояний соответствует одна клетка. Если для какой-нибудь пары состояний одному значению входного сигнала соответствуют различные выходные символы, следовательно, эта пара состояний не является эквивалентной, и в соответствующую клетку мы ставим крест. Таким образом, определяются пары состояний, неэквивалентность которых выявляется при считывании входного слова из одного символа (зачёркнуто крестиками чёрного цвета).

Таблица 1.

2	X							
3		X						
4		X						
5	X		X	X				
6		X			X			
7	X		X	X		X		
8	X		X	X		X		
9	X		X	X		X		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Далее, в не зачеркнутые клетки выписываем пары состояний, в которые переходит автомат из состояний, соответствующих этой клетке, при подаче на вход одинаковых входных символов.

Из этого правила записи есть два исключения:

1) не выписываем пары одинаковых состояний;
2) не выписываем пару, соответствующую заполняемой клетке.

Получим:

Таблица 2.

2	×							
3	5 7 2 9	×						
4	2 8	×	5 7 8 9					
5	×	3 6 1 7	×	×				
6	5 9 2 5	×	7 9 5 9	5 8 5 9	×			
7	×	3 6 1 5	×	×		×		
8	×	3 4 1 3	×	×	4 6 3 7	×	4 6 3 5	
9	×	1 3 1 4	×	×	1 6 4 7	×	1 6 4 5	1 4 3 4
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Затем, если внутри какой-нибудь клетки выписана пара, соответствующая зачёркнутой ранее клетке, то клетка с этой парой также зачёркивается (крестиком синего цвета). Зачёркнутые синим клетки теперь будут соответствовать парам состояний, неэквивалентность которых выявляется при считывании входных слов длины 2:

Таблица 3.

2	×							
3	5 7 2 9	×						
4	2 8	×	5 7 8 9					
5	×	~~3 6~~ ~~1 7~~	×	×				
6	5 9 2 5	×	7 9 5 9	5 8 5 9	×			
7	×	~~3 6~~ ~~1 5~~	×	×		×		
8	×	3 4 1 3	×	×	~~4 6~~ ~~3 7~~	×	~~4 6~~ ~~3 5~~	
9	×	1 3 1 4	×	×	~~1 6~~ ~~4 7~~	×	~~1 6~~ ~~4 5~~	1 4 3 4
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Затем, если внутри какой-нибудь клетки выписана пара, соответствующая клетке, зачёркнутой синим цветом, мы её также зачёркиваем (крестиком красного цвета). Зачёркнутые красным клетки теперь будут соответствовать парам состояний, неэквивалентность которых выявляется при считывании входных слов длины 3.

Таблица 4.

2								
3	5 7 2 9							
4	2 8		5 7 8 9					
5		~~3 6~~ ~~1 7~~						
6	~~5 9~~ ~~2 5~~		~~7 9~~ ~~5 9~~	~~5 8~~ ~~5 9~~				
7		~~3 6~~ ~~1 5~~						
8		3 4 1 3			~~4 6~~ ~~3 7~~		~~4 6~~ ~~3 5~~	
9		1 3 1 4			~~1 6~~ ~~4 7~~		~~1 6~~ ~~4 5~~	1 4 3 4
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Получили, что в каждой незачёркнутой клетке выписаны пары состояний, соответствующих только незачёркнутым клеткам. Значит, новых зачёркнутых клеток не возникнет, каждая незачёркнутая клетка соответствует парам эквивалентных состояний.

Объединяем эквивалентные состояния в классы эквивалентности: 1' = {1,3,4}; 2' = {2,8,9}; 3' = {5,7}; 4' = {6}.

Строим автомат с четырьмя внутренними состояниями 1', 2', 3', 4'. Обозначим его $S_{\min}$. Покажем, что это обозначение корректно, то есть $S_{\min}$ является минимальным автоматом. Если бы это было не так, то нашёлся бы автомат S' с меньшим, чем 4, числом внутренних состояний. Для каждого состояния q_i автомата $S_{\min}$ найдётся эквивалентное ему состояние $q_i^{'}$ автомата S'. Но так, как количество состояний автомата S' меньше, чем 4, найдётся состояние $q_j^{'}$ автомата S', эквивалентное по крайней мере двум различным состояниям q_n и q_m автомата $S_{\min}$, которые у нас не являются эквивалентными, что доказывает минимальность автомата $S_{\min}$.

Для построения таблицы состояний минимального автомата поступаем следующим образом: чтобы найти значение функций перехода и выхода, например, на наборе $(a,1')$ выбираем из класса эквивалентности 1' любого представителя. Пусть это будет состояние 3. Из таблицы состояний исходного автомата мы имеем, что $\lambda(a,3)=x$, значит, определим также функцию выхода минимального автомата: $\lambda(a,1')=x$. Из таблицы состояний исходного автомата имеем, что $\delta(a,3)=7$. Но $7\in\{5,7\}$=3', значит, будем считать, что $\delta(a,3)=3'$.

*Найдя значения функций перехода и выхода на всех наборах пере-*мененных, заполним таблицу состояний минимального автомата:

Таблица 5.

A \ Q'	1'	2'	3'	4'
a	$3',x$	$1',y$	$4',y$	$2',x$
b	$2',y$	$1',x$	$3',x$	$3',y$

Проверим работу автоматов S и $S_{\min}$ над каким-нибудь одним словом, например, над словом *abbabaab*. Пусть автомат S начинает свою работу, например, из 1 состояния:

Входное слово		*a*	*b*	*b*	*a*	*b*	*a*	*a*	*b*
Состояния	1	5	7	5	6	5	6	9	4
Выходное слово		*x*	*x*	*x*	*y*	*y*	*y*	*x*	*x*

Так как $1\in\{1,3,4\}$=1', то минимальный автомат $S_{\min}$ запускаем из состояния 1'. Получим:

	a	*b*	*b*	*a*	*b*	*a*	*a*	*b*
1'	3'	3'	3'	4'	3'	4'	2'	1'
	x	*x*	*x*	*y*	*y*	*y*	*x*	*x*

Как мы видим, результатом работы исходного и минимального автоматов над словом *abbabaab* является одно и то же слово *xxxyyyxx*.