Маршруты, цепи, циклы.

1. **Определения**
2. **Связанные компоненты графа**
3. **Расстояния**
4. **Диаметр, радиус, центр графа**
5. **Матрица и цепи, произведение графа**
6. **Эйлеровы и гамильтоновы графы**

1.Определения

Дан неориентированный граф G.

Определение: Маршрутом в графе G называется такая конечная или бесконечная последовательность ребер, что каждые два соседних ребра имеют общую инцедентную вершину.

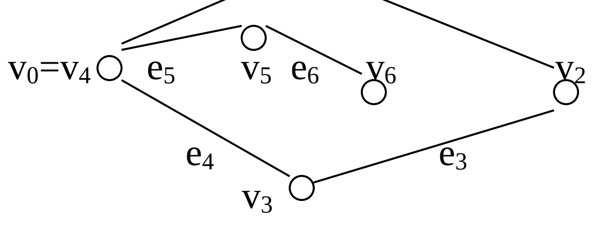
Одно и тоже ребро в маршруте может встречаться несколько раз. В дальнейшем будем рассматривать только конечные маршруты, то есть конечные последовательности ребер. В таких маршрутах имеется первое и последнее ребро.

Определение: Вершина v_0 инцидентная ребру e_0 и неинцидентная ребру e_1 называется началом маршрута.

Если имеются кратные ребра, необходимо специальное указание - какую из двух вершин считать началом маршрута.

Определение: Вершины, инцидентные ребрам маршрута, кроме начальной и конечной, называются внутренними или промежуточными.

Так как, различные ребра маршрута могут быть инцидентны одной и той же вершине, начало и конец маршрута могут одновременно оказаться и внутренней вершиной.



Пусть маршрут $M(e_1, e_2, ..., e_n)$ имеет начало в вершине v_0 и конец в вершине v_n, тогда его называют соединяющим вершины v_0 и v_n.

Число ребер маршрута называют его длинной.

Если $v_0=v_n$, маршрут называют циклическим.

Отрезок конечного или бесконечного маршрута М сам является маршрутом и называется участком маршрута М.

Маршрутом М называется цепью, если каждое ребро встречается в нем более одного раза и простой цепью, если любая вершина графа G инцидентна не более чем двум его ребрам.

Циклическим маршрутом называется число, если он является цепью и простым циклом, когда это простая цепь.

$(e_2, e_3, e_4, e_5, e_6)$ – цепь,

(e_1, e_5, e_6) – простая цепь,

$(e_1, e_2, e_3, e_4, e_5)$ – цепь,

$(e_6, e_5, e_1, e_2, e_3)$ – простая цепь,

$(e_6, e_5, e_1, e_2, e_3, e_4, e_5)$ – маршрут не являющийся цепью.

2.Связанные компоненты графа

<u>Определение:</u> Вершины ν', $\nu'' \in G$ называются связаными, если существует маршрут М с началом в вершине ν' и концом в вершине ν''.

Пусть вершина ν инцидентна более чем двум ребрам маршрута М связывающим вершины $\nu'\nu''$. Ребро l_i - первое из этих ребер, l_j последнее из этих ребер (j>i+1), тогда из маршрута М можно выбросить участок от i+1 ребра до i-1, получится маршрут $M'(l_1,l_2,\ldots,l_i,l_j,\ldots,l_n)$. Если М' не простая цепь, то процесс вырабатывания внутреннего участка можно продолжать до тех пор, пока не получится простая цепь связывающая вершины ν',ν''. Таким образом, связанные маршрутом вершины связаны так же и простой цепью.

Если вершина ν связана с какой-либо другой вершиной ν', она связана и сама с собой. Пусть ν и ν' связывает маршрут $M(l_1,l_2,\ldots,l_n.)$, тогда ν и ν' связывает маршрут $M'(l_1,l_2,\ldots,l_n\ldots,l_{n-1},\ldots,l_2,l_1)$, в котором сначала идут ребра М маршрута, а потом они же, но в обратном порядке. Изолированная вершина так же связана сама с собой. Данное отношение рефлексивно, симметрично, транзитивно. Если ν' связана с ν'', а ν''с ν'''

Маршрут М состоит из маршрута М' и М'', кроме ребер l_n' и l_1'' остальные пары соседних ребер являются соседними в одном из маршрутов М' или М''. Ребра l_n' и l_1'' инцидентны вершине v'' и являтюся соседними из двух разных маршрутов.

Граф G называется связным , если все его вершины связаны между собой, поэтому все подграфы данного графа связаны и называются связными компонентами рассматриваемого графа.

Графы с различными компонентами связности:

1. Расстояния

Пусть G связный неориентированный граф; v' и v'' любые его вершины, тогда существует связывающая их простая цепь $M(e_1, e_2,\dots, e_n)$. Если количество ребер этой цепи не минимальное из возможных, то существует цепь M' из ребер $(e'_1, e_2',\dots, e_n')$, который связывает v' и v'' и имеет меньшее число ребер.
Если M' не минимальна, можно найти цепь с еще меньшим количеством ребер и так далее. Однако, этот процесс выбрасывания ребер можно повторить не более n-раз. Поэтому существует цепь, связывающая V' и V'' с минимальным количеством ребер n.

<u>Определение:</u> Минимальная длинна простой цепи с началом в вершине v' и концом в v'' называется расстоянием между вершинами v' и v'' и обозначается
$d\,(v'v'')$.

Считая каждую вершину связанную саму с собой по существу вводятся нулевые маршруты не содержащие ребер с началом и концом в любой вершине v. В соответствии с этим расстояние $d\,(v, v)=0$. Для любой пары вершин расстояние $d>0$.

Расстояние удовлетворяет следующим аксиомам метрики:

a. $d(v'\,v'')>=0$, причем $d(v'\,v'')=0$ тогда, когда $v' = v''$;
b. $d(v'v'')= d\,(v''v')$;
c. $d(v'v'')+d\,(v''v''')>=d\,(v'v''')$.

2. Диаметр, радиус, центр графа

Пусть дан конечный, неориентированный, связанный граф.

<u>Определение</u>: Диаметром графа называется максимальное расстояние между вершинами v' и v'' , и обозначается d(G).

<u>Определение:</u> Кратчайшие простые цепи, связывающие $v'v'' \in G$ с максимальным расстоянием между ними, называются диаметральными простыми цепями.

Пусть v произвольная вершина рассматриваемого графа, максимальное удаление от v будет величина r (v) = max d(v v''), где $v' \in G$.

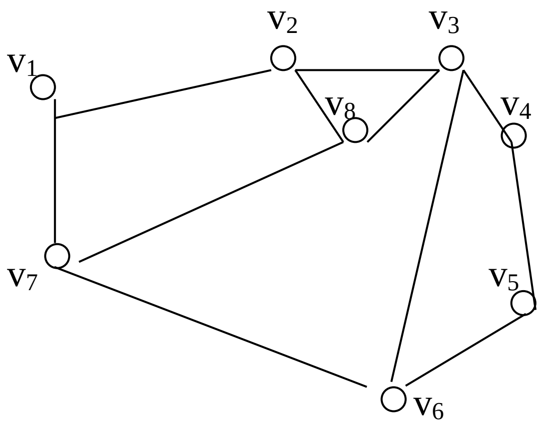
Вершина v называется центром графа, если максимальное удаление от нее принимает минимальное значение.

$$r(G) = \min r(v'),\ v' \in G.$$

Центр необязательно может быть единственным. Если возьмем полный граф, где любые вершины соеденены одним ребром R=1 и любая вершина может быть центром.

Максимальное удаление r (G) от центра называется радиусом графа, а любая кратчайшая цепь от центра v до максимально удаленной от него вершины v' называется радиальной цепью.

Определим расстояния до вершин, центры графа, радиус:



	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	d(v)
v_1	0	1	2	3	3	2	1	2	3
v_2	1	0	1	2	3	2	2	1	3
v_3	2	1	0	1	2	1	2	1	**2**
v_4	3	2	1	0	1	2	3	2	3
v_5	3	3	2	1	0	1	2	3	3
v_6	2	2	1	2	1	0	1	2	**2**
v_7	1	2	2	3	2	1	0	1	3
v_8	2	1	1	2	3	2	1	0	3

v_3 v_6 – центры графа , d(G)=3, r (G)=2.

5. Матрица и цепи, произведение графа

Пусть G и H два графа, оба ориентированные или неориентированные без кратных ребер с одним и тем же множеством вершин V.

Произведение этих графовF=GH: ребро ($v'v''$), существует в том, и только том случае, когда для некоторой вершины $v\in V$ существуют ребра $(v'v)\in G$ и $(vv'')\in H$, то есть существует маршрут из двух ребер с началом в вершине v' и концом в вершине v''. Причем первый элемент маршрута принадлежит графу G, а второй принадлежит графу H.

Матрица смежности произведения графов равна произведению матриц смежности сомножителей, только вместо обычных произведений и сумм нужно рассматривать логические.

$$\delta^{(F)}_{ij}= \vee\ (\delta^{(G)}_{ik}\&\delta^{(H)}_{kj})$$

Прямые произведения графов.

Пусть даны графы G_1 и G_2 с множествами вершин V_1 и V_2 . У прямого произведения этих графов множеством вершин является прямое произведение $V=V_1*V_2$. Ребра произведения могут быть определены различными способами. При одном из определений F графов G_1 и G_2 ребро ($v_1'v'_2$) и ребро ($v_1''v''_2$) имеется тогда, когда в графе G_1 есть ребро ($v_1'v''_1$) или в графе G_2 есть ребро ($v_2'v''_2$). В другом определении для этого требуется существование обоих этих ребер. $G_1(v_1'v''_1)$ и $G_2(v_2'v''_2)$.

Для этих случаев элементы матрицы смежности определяются следующей формулой:

1) $\delta(i_1i_2)\ (j_1j_2)= \delta i_1j_1 V\ \delta i_2j_2$;
2) $\delta(i_1i_2)\ (j_1j_2)= \delta i_1j_1\&\ \delta i_2j_2$

Таким образом, в обоих случаях матрица смежности произведения является кроникkеровским произведением матриц смежности сомножителей, но в первом случае произведение элементов матрицы понимается как логическая сумма ***или***, а во втором - как логическое произведение ***и***.

Чаще всего используется третий способ, согласно которому ребро ($v'_1v'_2$) ($v''_1v''_2$) существует в тех и только тех случаях, когда в графах G_1 и G_2 есть соответственно ребра ($v_1'v''_1$) и ($v_2'v''_2$) или $v'_1= v^1_2$, а в графе G_2 есть ребро ($v_2'v''_2$) или в графе G_1 ($v'_1v'_2$), а вершины $v''_1 = v''_2$.

Пусть $\Delta' = E \vee \Delta$, где E- единичная матрица, а Δ- матрица смежности рассматриваемого графа :

$$\delta'_{ij} = \begin{cases} \delta_{ij}, & i \neq j \\ 1, & i = j \end{cases}$$

тогда матрица смежности прямого произведения графов G_1 G_2 при таком определении равна

$$\Delta^{(F)} = \Delta'^{(G1)} \times \Delta'^{(G2)} - E^{(G1)} \times E^{(G2)}$$

т.е. $\delta(i1i2)(j1j2) = \begin{cases} \delta'_{i1j1}\,\delta'_{i2j2}, & i_1 \neq j_1 \text{ или } i_2 \neq j_2 \\ 0 & i_1 = j_1 \text{ и } i_2 = j_2 \end{cases}$

6. Эйлеровы и гамильтоновы графы

<u>Определение:</u> Эйлеровым путем (циклом) графа называется путь (цикл) содержащий все ребра графа ровно один раз.

Граф, обладающий эйлеровым путем (циклом) называется эйлеровым графом.

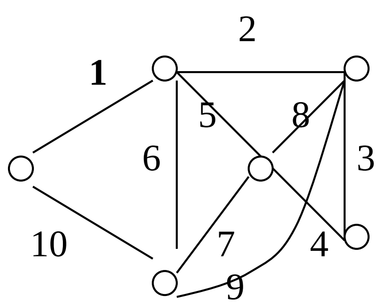


рис.1

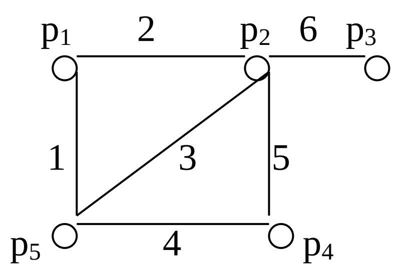


рис.2

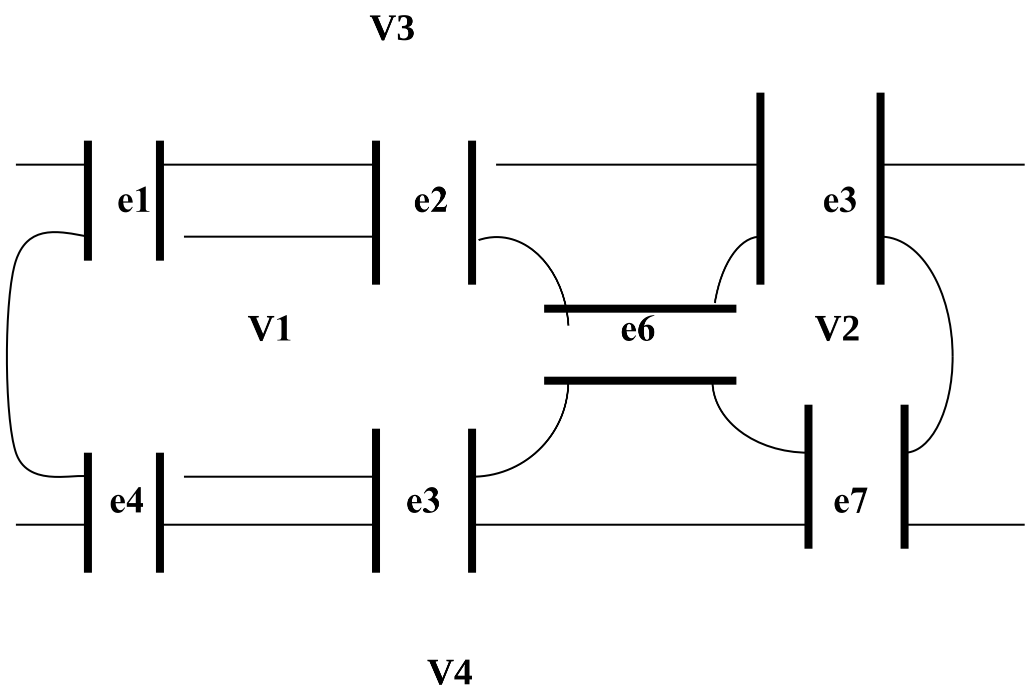
Граф на рис.1 является эйлеровым, т.к. последовательность ребер (1,2,3,4,5,6,7,8,9,10) образует элеров цикл.

Граф на рисунке 2 не является эйлеровым, но он обладает эйлеровым путем (1,2,3,4,5,6) с концевыми вершинами p_1 и p_2.

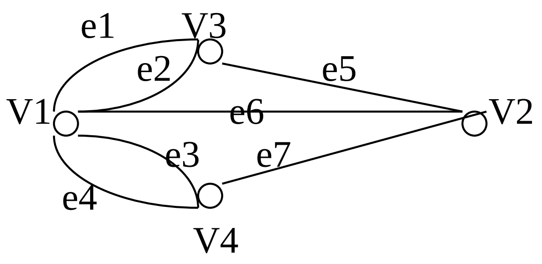
Теорема: Граф G обладает эйлеровым путем с концами в вершинах P_1P_2 тогда и только тогда, когда G связный и P_1P_2 единственные вершины нечетной степени.

Теорема: Граф G является эйлеровым тогда и только тогда, когда G связный и все его вершины имеют четную степень.

Задача о Кенигсбергских мостах:



Задача о мостах переводится на язык теории графов. За вершины графа примем берега реки и острова, а за ребра – мосты, причем ребро должно быть инцидентно двум вершинам, если мост соединяет две соответствующие части города.

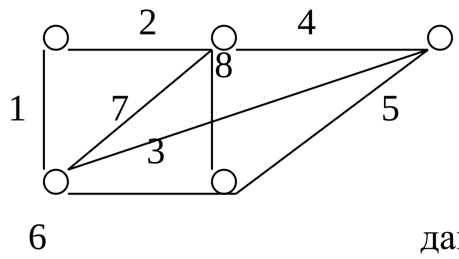


В графе задачи о мостах все вершины имеют нечетную степень, поэтому ни эйлерова цикла, ни эйлеровой цепи в нем не существует.

<u>**Определение:**</u> Гамильтоновым путем (циклом) графа G называется путь (цикл) проходящий через каждую вершину графа G в точности по одному разу.

Граф, обладающий гамильтоновым циклом, называется гамильтоновым.

Достаточным условием существования гамильтонова цикла является полнота графа G.



данный граф имеет гамильтонов цикл (1,2,4,5,6).

Некоторые классы графов и их частей.

1. **Дерево и лес.**
2. **Концевые вершины и ребра.**
3. **Дерево с корнем, ветви.**
4. **Типы вершин и центры деревьев.**
5. **Цикломатическое число графа.**
6. **Максимальные потоки в цепи. Дивергенция потока.**

1. Дерево и лес.

Определение: Неориентированное дерево – это связный неориентированный граф без циклов, петель и кратных ребер.

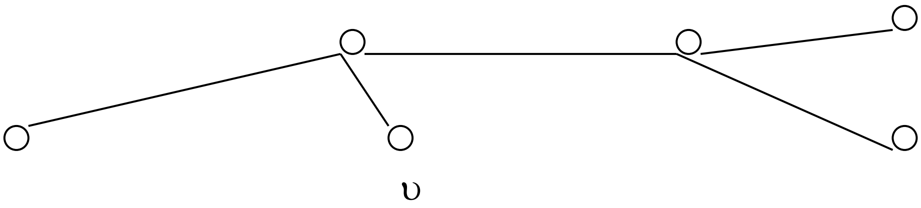
Несвязанный граф без циклов называется лесом. Связанные компоненты леса являются деревьями. Любая цепь в таких графах является простой.

Теорема: Любые две вершины дерева υ’ и υ’’ связаны одной и только одной цепью.

Верно и обратное: если любые две вершины графа G связаны единственной цепью этот граф является деревом. Действительно из простого цикла всегда можно составить две непересекающиеся простые цепи с одним и тем же началом и концом.

2. Концевые вершины и ребра.

<u>Определение:</u> Вершина υ графа G называется концевой или висячей, если ее степень равна единице $\rho(\upsilon)=1$.



Ребро инцидентное концевой вершине является концевым. Если конечное дерево состоит более чем из одной вершины, оно имеет хотя бы две концевые вершины и хотя бы одно концевое ребро.

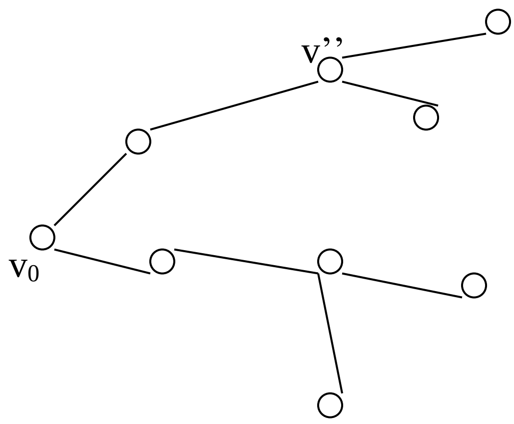
Пусть υ вершина дерева G, т.к. она связана с другими вершинами, из нее выходит хотя бы одно ребро. Если другой конец υ' этого ребра не является концевой вершиной из него выходит еще одно ребро и т.д. Таким образом, строится цепь, проходящая все время через новые вершины, иначе часть этой цепи оказалось бы циклом. Если дерево конечно, процесс построения этой цепи конечен.

3. Дерево с корнем, ветви.

<u>Определение:</u> Пусть в дереве G отмечена вершина υ_0, эту вершину называют – корнем дерева. Само дерево будет – дерево с корнем.

Корень в дереве можно назначить любой. В дереве можно естественным образом ориентировать ребра.

Вершину υ' ребра $(\upsilon'\ \upsilon'')$ можно соединить единственной цепью L с корнем υ_0 $(L(\upsilon', \upsilon''))$. Если эта цепь не содержит ребра $(\upsilon'\upsilon'')$ в цепь вводится ориентация от υ' к υ'', в противном случае от υ'' к υ'. Эта ориентация согласована с ориентацией того же ребра определенной через вершину υ''. Действительно, если цепь L не содержит ребра $(\upsilon'\upsilon'')$, то, добавив это ребро, мы получим единственную цепь L', связывающую υ_0 и υ'', и проходящую через рассматриваемое ребро. Если цепь L содержит ребро $\upsilon'\upsilon''$, то отбросив его мы получим цепь L' связывающую υ_0 и υ'', и не содержащую этого ребра.



Ориентированное таким образом дерево с корнем называется ориентированным деревом, где все ребра имеют направление от корня. Если изменить направление ребер к корню, получится так же ориентированный граф, который называется ориентированным деревом или сетью сборки. В каждую вершину ориентированного дерева входит только одно ребро, то есть эта вершина является концом одного и того же ребра. Действительно, такое ребро является последним в цепи связывающей рассматриваемую вершину с корнем, эта цепь единственна.

В корень не входит не одно ребро, все инцидентные корню ребра связывают его со своими вторыми концами, значит, корень является их началом. Любое дерево можно ориентировать, выбрав в качестве корня любую его вершину. Это доказывает, что в конечном дереве число вершин на единицу больше числа ребер.

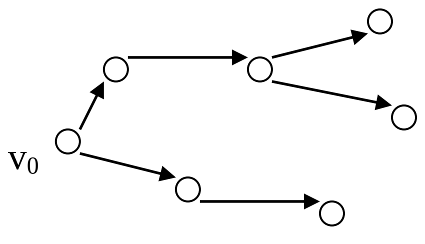


рис.1

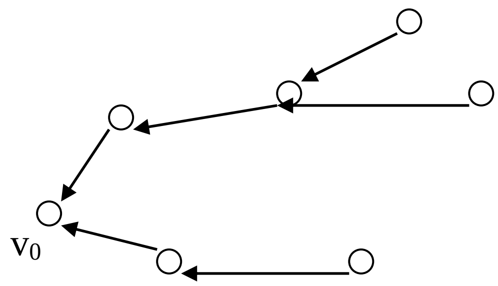
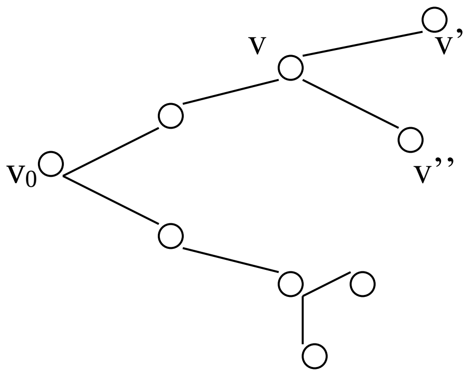


рис.2

Рис.1 представляет ориентированное дерево.
Рис.2 представляет ориентированное дерево или *сеть сборки*
Пусть υ вершина дерева G с корнем υ_0. В (υ)-множество всех вершин связанных с корнем цепями, проходящими через вершину υ, то есть

имеющими ребра инцидентные вершине υ.



Это множество порождает подграф G(υ), который называется ветвью вершины υ в дереве с корнем υ_0, G(υ) тоже не имеет циклов, любые вершины υ'υ'' $\in$ B(υ) связаны в этой ветви маршрутом составленным из участков цепей L(υ') и L(υ'') от корня υ_0 до этих вершин начинающихся в вершине υ. Нужно доказать, что эти участки полностью погружены в подграф G(υ). Это справедливо, т.к. в каждую вершину такого участка ведет цепь из корня υ_0 являющаяся начальным участком цепи L(υ') или L(υ'') и проходит через вершину υ. Таким образом, ветвь G(υ) связана и сама является деревом.

Рассмотрим G(υ), как дерево с корнем υ. Для любой вершины υ'$\in$G(υ), связывающее ее с вершиной υ цепь L(υ υ') – это конец цепи L(υ_0 υ') связывающий в дереве G эту вершину с корнем υ_0 следовательно, последнее ребро этой цепи входящее в вершину υ' в исходном дереве G и в его ветви G(υ) ориентируется в обоих этих деревьях с корнем одинаково, так что вершина υ' является концом.

4. Типы вершин и центры деревьев.

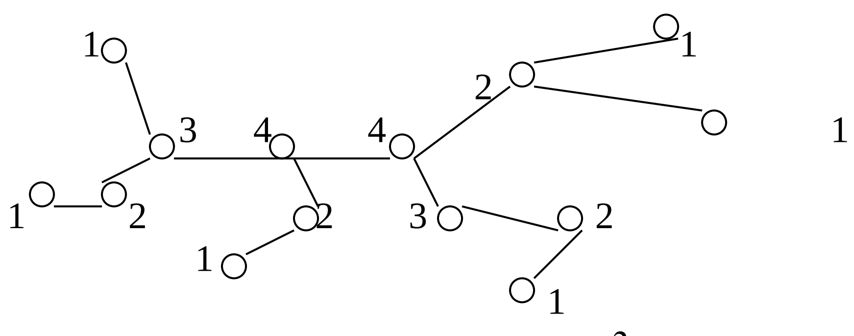


рис.3

Пусть дано некоторое дерево G. Его концевые вершины обозначим типом 1.
Удалим из дерева G все вершины типа 1 и инцидентные им концевые ребра. Останется граф G’ без циклов значит дерево. Дерево G’ также имеет концевые вершины, которые будем называть вершинами типа 2 в дереве G. Аналогично определяются вершины типов 3, 4 и т. д. рис.3

Отметим, что если дерево имеет более 2-х вершин, то среди них есть внутренние вершины. Пусть υ_1 и υ_2 концевые вершины, они связаны цепью. Если эта цепь состоит из одного ребра, то υ_1 и υ_2 не связаны ни с какими другими вершинами, что противоречит связанности дерева. Если эта цепь состоит из 2-х или более ребер, то она проходит через вершины степени большей или равной двум, так как не концевые вершины.

В конечном дереве G имеется вершины конечного числа типов. Максимальный тип вершины равен первому типу и второму. Каждая вершина дерева, кроме концевых, имеет соседей на единицу меньшего типа. Действительно в дереве определяющей предыдущий тип вершина не была концевой, затем стала концевой и так далее, следовательно соответствующие им соседние вершины имеют меньший тип.

5. Цикломатическое число графа.

Пусть G конечный неориентированный граф. Его цикломатическое число рассчитывается по следующей формуле:

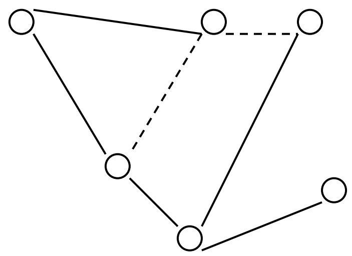
$$\gamma(G)= \nu_c+\nu_e-\nu_v$$

ν_c – число связанных компонент графа.
ν_e – число ребер.
ν_v – число вершин.

Цикломатическое число дерева равно нулю.
Цикломатическое число леса состоит из суммы цикломатических чисел деревьев, равно нулю.
Цикломатические числа остальных конечных графов положительны.
Допустим что в графе есть цикл $Z(e_1,e_2,\ldots,e_n)$. Можно выбросить ребро, граф остается связанным. Пусть маршрут $M(e_1',e_2',\ldots,e_n')$ связывает вершины $\nu'\nu'' \in G$. Этот маршрут содержит выбранное ребро l_q цикла Z. Это ребро можно заменить маршрутом $M'(e_q,e_{q+1},\ldots,e_n,e_1,e_2,\ldots,e_{q-1})$ состоящим из остальных ребер цикла Z и имеющим те же концы, что и само ребро. Если в графе есть еще циклы их можно выбросить еще ребро, не нарушая связности графа и т.д. Ребра удаляем до тех пор из циклов, пока не получим связанное дерево G^γ (т.к. число вершин всех конструируемых графов G’, G'', G''' … G^γ не меняется, если оно больше единицы, то все ребра выбросить нельзя). Число оставшихся ребер на единицу меньше числа вершин графа G^γ , а

значит, и исходного графа G, но в последнем на γ ребер больше следовательно

$$\gamma(G) = \nu_c(G^\gamma) + \nu_e(G^\gamma) - \nu_v(G^\gamma) {}_+ \gamma = \gamma > 0$$



$\gamma(G) = 2$

Для того чтобы определить цикломатическое число этого графа мысленно будем разбивать циклы.

6. Максимальные потоки в цепи. Дивергенция потока.

<u>Определение:</u> Поток на графе – совокупность однородных объектов пересылаемых из одной вершины в другую по его ребрам.

Если вершины p и q соединены другой $\alpha=(p,q)$, то поток из вершины p в вершину q обозначается f(p,q) и таким образом поток, заданный на дугах графа будет являться некоторой функцией.

Пусть имеется ориентированный граф, на дугах которого определена функция C(p,q). В потоковых задачах функция C(p,q) означает пропускную способность дуги (p,q) и является стоимости перевозки единицы потока по этой дуге.

<u>Определение:</u> Дивергенцией потока f в вершине p называется разность выходных и входных потоков:

$$\text{Div } f(p) = \sum_{(p,q)\in A(p)} f(p,q) - \sum_{(r,p)\in B(p)} f(r,p)$$

A(p) – множество дуг выходящих из вершины p.
B(p) – множество входящих вершин.

Вершины, в которых дивергенция потока $\mathrm{div} f(p)>0$ называются источником потока f, а вершины, в которых дивергенция потока $\mathrm{div} f(p)<0$, называются стоками.

Сложим дивергенцию потока f во всех вершинах графа:

$$D = \sum_{p} \mathrm{div}\, f(p) = 0$$

Пусть задана сеть, в которой выделены две вершины s и t, рассмотрим такие потоки $f(\alpha)$, что:

1). $0<=f(\alpha)<=C(\alpha)$

2). Div $f(p)=0$, при всех p, кроме $p=s$, $p=t$.

Вершины s и t назовем полюсами сети, а остальные вершины назовем внутренними.

Из предыдущего следует, что

$$\mathrm{div}\, f(s) = -\,\mathrm{div}\, f(t) = M$$

Если $M=0$, то поток называется циркуляцией.

Если $M\neq 0$, вершина s является источником потока, вершина t – стоком.

Если $M>0$, то M назовем мощностью потока f.

Задача состоит в том, чтобы найти поток f максимальной мощности удовлетворяющей условиям 1,2.

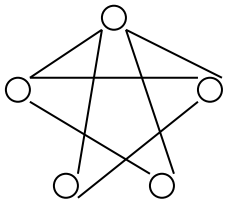
Если рассматриваемая сеть содержит параллельные дуги, нужно рассмотреть новую сеть, с теми же вершинами где параллельные дуги α и β - склеены. Пропускная способность, полученная при этом дуги $\gamma=\alpha+\beta$, будет $C(\gamma)=C(\alpha)+C(\beta)$.

После получения решения потока f для сети поток разбивается на сумму:

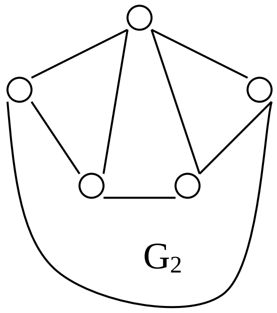
$$f(\gamma)=f_1+f_2 \text{ , где}$$
$$0\leq f_1 \leq C(\alpha)$$
$$0\leq f_2 \leq C(\beta)$$

Плоские и планарные графы.

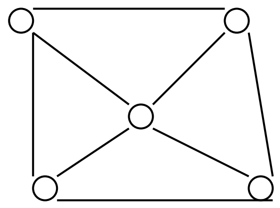
Плоскими – называются графы, изображающиеся на плоскости так, что не какие два ребра этого графа не пересекаются за исключением инцидентных им вершин.



G_1



G_2



G_3

G_2 и G_3 являются плоскими графами.

Граф изоморфный плоскому называют – планарным.
Часть плоскости, любые две точки которой можно соединить так, чтобы линии их соединения не пересекали ребер плоского графа, называется гранью этого графа.
n – вершины, m – ребра, l – грани.

G_2 : n=5; m=8; l=5;
G_3 : n=5; m=8; l=5;
G_4 : n=2; m=1; l=1;

<u>Теорема:</u> для любого плоского связного графа имеет место формула n-m+l=2.

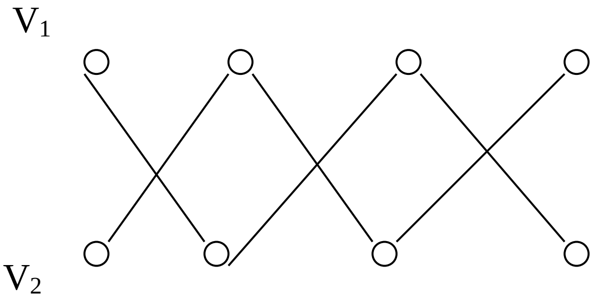
Доказательство: рассмотрим связанный плоский граф с n – вершинами, m – ребрами, l – гранями. Если этот граф дерево, то m=n-1, l=1. Эта формула выполняется для дерева. Пусть G произв. связан. граф не является деревом, построим дерево покрывающее этот граф. Для этого удалим ребра образующие цикл. При удалении любого такого ребра две грани сливаются, поэтому русло ребер уменьшается на единицу и грани тоже, а значит величина l – m не меняется. Следовательно величина n+l-m у плоского графа будет такое же, как у покрывающего его дерева, то есть n+l-m=2.

Двудольные и двойственные графы.

1. **Двудольные графы**
2. **Максимальные двудольные части графа**
3. **Двойственные графы**
4. **Графы - язык в дискретной математики**

1. Двудольные графы

<u>Определение:</u> Граф G называется двудольным, если множество его вершин V распадаются на два пересекающихся подмножества V_1 и V_2 таких, что каждое ребро графа имеет один конец из множества V_1, а другой из множества V_2**.**



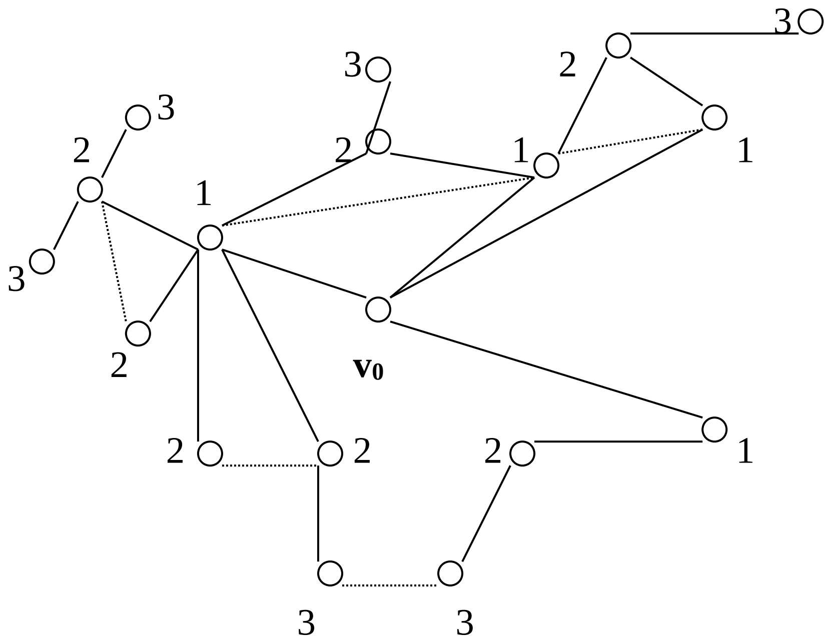
<u>Теорема:</u> Граф G является двудольным тогда и только тогда, когда все его циклы имеют четную длину.

<u>Доказательство:</u> Пусть G двудольный граф. Тогда каждый его цикл имеет четную длину, т.к. вершины, через которые проходит цикл, принадлежат поочередно множествам V’и V’’. Обратно, пусть каждый цикл имеет четную длину. Не нарушая общности, предположим, что G связен. Возьмем в графе произвольную вершину v_0 V’ к V’’ отнесем вершины, расстояния от которых от v_0 нечетны и четны соответственно. Мы должны показать, что в G не существует ребра, соединяющего вершины v_1 и v_2 из одного класса, например из V’. Пусть такое ребро (v_1, v_2) существует. Рассмотрим простые цепи L_1 и L_2 соответственно нечетной и четной длины, соединяющие v_1 и v_2 с v_0. Пусть v_3 их «первая» общая вершина. Тогда цепи L_1 и L_2 разбиваются на части:
$L_1= L_1(v_1v_3) \cup L_1(v_3v_0)$ и $L_2=L_2(v_2v_3) \cup L_2(v_3v_0)$
Ясно, что цепи $L_1(v_3v_0)$ и $L_2(v_3v_0)$ имеют длины одной четности (это следует из того, что все циклы – четной длины) , тогда цикл $L_1(v_1v_3) \cup L_2 (v_3v_2) \cup (v_2v_1)$ имеет четную длину, что противоречит условию. Теорема доказана.

2.Максимальные двудольные части графа

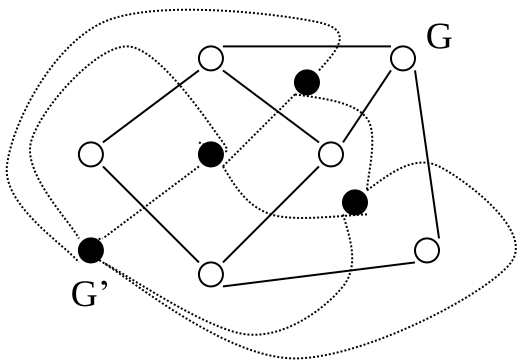
Алгоритм построения максимальной части графа:
В каждой связной компоненте G_0 графа G выберем вершину v_0 . для каждой вершины $v\in G_0$ определим ее расстояние $d(v,v_0)$ до v_0 . Тем самым вершины G разделятся на два класса : вершины с четным и нечетным расстоянием до v_0 . Из G_0 удалим ребра, соединяющие вершины одного класса. Рассуждения, сходные с доказательством предыдущей теоремы, показывают, что оставшаяся часть графа G – его максимальная двудольная часть .

Пример: В данном графе у каждой вершины обозначено ее расстояние до v_0. Пунктиром обозначены ребра, которые требуется удалить, чтобы получить максимальную двудольную часть графа.



3. Двойственные графы

Границы областей карты - это плоский граф, его вершины точки, где сходятся по три области. Ребро - соединяющие их, линии - это общие границы двух областей. В каждой области выберем точку - это будет центр. Её и центр соседних областей соединим линиями, получается двойственный граф, который является плоским.



Области двойственного графа окружают вершины графа границ карты, т.к. рассмотрим карты, в вершинах графов границ которых сходятся по три области получается двойственный граф, каждая область которого треугольник (криволинейный, при чем внешняя область тоже ограничена треугольником). Таким образом, двойственный граф определяет так называемую триангуляцию плоскости, он всегда связан.

Конфигурация- это связный подграф плоского графа, порожденный некоторым подмножеством его вершин. Остальные вершин и ребра графа составляют внешнюю часть конфигурации.

Конфигурацию можно “стянуть”, т.е. исключить из неё некоторые вершины, не имеющие инцидентных ребер из внешней части. Отождествить некоторые другие и соединить оставшиеся вершины ребрами может быть по иному. Внешняя часть присоединится к соответствующей вершине новой конфигурации, при чем требуется, чтобы полеченный граф оказался плоским. Если исходный граф был триангуляцией.

4. Графы - язык в дискретной математики

При формулировке задач в ДМ или описания методов их решений часто употребляется язык теории графов. Покажем один достаточно общий способ постановки задач, при котором естественным образом возникает граф.

Пусть рассмотрим множество V объектов υ, каждый из которых может находиться в состояниях x_υ из заданного для этого элемента υ множество состояний X_υ. Состояния различных объектов взаимозависимы так, что для каждого объекта υ заданы множества Q_υ объектов влияющих на υ и функция $f_\upsilon(\upsilon_1,\ldots,\upsilon_s)$ называется функцией влияния, которая определяет состояние x_υ объекта υ по различным состояниям $X\upsilon_I$, $\upsilon_i \in Q_\upsilon$; $X_\upsilon=f_\upsilon(\upsilon_1,\ldots,\upsilon_s)$. Если $Q_\upsilon=0 \Rightarrow$, что $X_\upsilon=f_\upsilon$, где f_υ является заданным состоянием. В данном случае объекты $\upsilon \in V$ можно рассмотреть, как вершины структурного графа определенного ранее системой соотношения между состояниями этих объектов.

Автоматы. Основные понятия.

Студент должен:

знать:
- базовые множества и принцип работы автомата;
- понятие правильный автомат и упрощенный вид диаграммы для него;
- понятие автомат распознающий свойства слова',

уметь:
- по таблице автомата строить его диаграмму; по диаграмме автомата записывать его таблицу;
- для заданного автомата по заданному входному слову записывать соответствующее выходное слово;

1.Определения
2.Изоморфизм и эквивалентность автоматов
3.Частные автоматы и их минимизация
4.Интерпретация автоматов, основные проблемы абстрактной теории автоматов

1.Определения

Конечным автоматом называется система $S=\{A;Q;V;\delta;\lambda\}$, в которой $A=\{a_1,\ldots,a_m\}$ $Q=\{q_1,\ldots,q_n\}$ $V=\{v_1,\ldots,v_k\}$- конечные множества (алфавиты), а $\delta:Q\times A\to Q$, $\lambda:Q\times A\to V$- это функции определение этих множеств.
Множество A- входной алфавит,
множество V- выходным алфавитом,
множество Q - алфавит состояний.
(сим)- функция переходов
λ- функция выходов.
Если в автомате S выделено 1 состояние, называемое началом (обычно будет считаться, что это q_1), то полученный автомат называется инициальным и обозначается (S;q). Таким образом по не инициальному автомату с n состояниями можно n различными способами определить инициальные автоматы.
Поскольку функции (сим) и λ определены на конечных множествах их можно задавать таблицами, обычно 2 таблицы сводятся в одну. И получается, что $\delta\times\lambda : Q\times A\to Q\times V$. Эта таблица называется таблицей переходов автоматов или автоматной таблицей.

2.Изоморфизм и эквивалентность автоматов

Дано два автомата S и T.
$S=(A_s,Q_s,V_s,(\text{сим})_s,\lambda_s)$ $T=(A_t,Q_t,V_t, \delta_t,\lambda_t)$
f: $A_s\to A_t$

$q: Q_s \to Q_t$
$h: V_s \to V_s$

эти отображения называются гомоморфизмом автомата S в автомат T, если для любых элементов $a \in A_s$, $q \in Q_s$,
$v \in V_s$ выполняются условия:
$\delta_t(g(q), f(a)) = g(\delta_S(q,a)))$
$\lambda_t(g(q)), f(a)) = h(\lambda_S(q,a))$
Если все три отображения сюръективны, то это тройка называется гомоморфизмом автомата S на автомат T, если эти три отображения взаимно однозначны, то они называются изоморфизмом автомата S на автомат T.
Автоматы для которых существует изоморфизм называется изоморфными.
Мощности соответствующих алфавитов изоморфных автоматов одинаковы.
Понятие изоморфизм имеет для автоматов тот же смысл, что и для алгебр: автоматы S и T изоморфны, если входы, выходы и состояния автомата S можно переименовать так, что таблица переходов автомата S преобразуются в таблицу переходов автомата T.
Изоморфизм графов переходов является необходимым, но недостаточным условию изоморфизма соответствующих автоматов.
При гомоморфизме помимо переименования происходит еще и склеивание. Пусть S и T два автомата с одинаковыми состояниями q автомата S и состояния r автомата T называется неотличимыми. Если для любого входного слова α, $S(q;\alpha)=T(r;\alpha)$. Если автомат F=S, то речь идет о неотличимых состояниях одного и того же автомата S. Неотличимость состояний q_i и q_j автомата S означает, что инициальные автоматы $(S;q_i)$ и $(S;q_j)$ реализуют одно и тоже автоматное отображение.
Автоматы S и T называются неотличимыми если для любого состояния q автомата S найдется неотличимое от него состояния r автомата T и наоборот.
Неотличимость автоматов означает, что любое автоматное отображение реализуемое одним из них может быть реализовано другим; их возможности по реализации преобразований входной информации в выходную совпадают. Отношение неотличимости между состояниями и автоматами рефлексивно, семетрично и транзитивно и следовательно является отношением эквивалентности. Неотличимость называется эквивалентностью. Поэтому будим говорить об эквивалентных состояниях или эквивалентных автоматов.
Переход от автомата S к эквивалентному автомату называется эквивалентным преобразованием автомата S. Можно ставить различные задачи о поиске автоматов эквивалентных другому и обладающих заданными свойствами. Наиболее изученной среди таких задач является

задача о минимизации числа состояний автомата или о минимизации автоматов: Среди автоматов эквивалентных S, найти автомат с наименьшим числом состояний, т.е. минимальный автомат.

3.Частные автоматы и их минимизация

Автомат S называется частичным или не полностью определенным автоматом, если хотя бы одна из его двух функций не полностью определена, т.е. для некоторых пар (состояния вход) значение функции (сим) и λ не определены.

Входное слово αдля которого $S(q_i;\alpha)$определенно называется допустимым для q_i. Функции переходов и выходов неравноправны. Если δ не определена на слове α, то она не определена и на всех его продолжениях. Для λ это необязательно, поэтому если δ определена на α, а λ не определена на некоторых начальных отрезков α отображение $S(q_i;\alpha)$ “определено но не совсем ”. Оно представляет собой слово содержание в прочерке. Эта ситуация интерпретируется на графе: Если δ не определена на α, то путь α из состояния q_i не определен, потому не ясно как его продолжить. Если же путь α из q_i определен, то идя по нему можно прочесть выходное слово $S(q_i;\alpha)$; ребра пути αна которых не написано выходных букв соответствуют прочерки.

Состояние q_i автомата S и состояния r_j автомата T называется совместимыми, если существует состояние p_k автомата W покрывающие q_i и r_j. Автоматы S и T совместимы если существует автомат W покрывающие S и T.

Состояние q_i и r_j совместимы если любого α $S(q_i;\alpha)$ и $T(r_j;\alpha)$ не определено, либо входные слова этих автоматов может быть содержащие прочерки не противоречивы, т.е. не содержат на одинаковых местах различных букв.

Состояние q_i автомата S покрывает состояние r_j автомата T (S и T совпадают), если для любого α из того, что $S(r_j;\alpha)$ определено следует, что $S(q_i;\alpha)$ определено и $S(r_j;\alpha)=S(q_i;\alpha)$. Автомат S покрывает T, если для любого состояния автомат T найдется покрывающие его состояния автомата S. Понятие покрытие и совместимости дают общий план минимизации частных автоматов.

4.Интерпретация автоматов, основные проблемы абстрактной теории автоматов

Конечный автомат представляет собой абстрактную, но с функциональной т.д. довольно точную модель дискретного (цифрового) вычислительного или управленческого устройства.

Входная буква - это входной сигнал (точнее комбинации сигналов) на всех входах устройства.

Входное слово- это последовательность входных сигналов поступающих в автомат в
дискретные моменты времени. t=1,2,3,….

Выходное слово - это последовательность выходных сигналов выдаваемым автоматом.

Состояние автомата - это комбинации состояний запоминающих элементов устройства.

Такая интерпретация служила основным стимулом развития и источником задач теории автоматов. Все что действительно существенно в этой теории - это работа со словами при наличии конечной памяти. При подходе к теории автоматов как к части теории алгоритмов центральной проблемой является изучение возможностей автоматов в терминах множеств слов, с которыми работают автоматы.

Можно выделить два основных аспекта работы автоматов:
1. автоматы распознают входные слова, т.е. отвечаю на вопрос принадлежит ли поданное на вход слово данному множеству (это автоматы распознаватели):
2. автоматы преобразуют входное слова в выходное, т.е. реализуют автоматные преображения (это автоматы преобразователи).

Основная задача теории автоматов - это задача описания автоматов и их реализация, т.е. представление автоматов как структуры, состоящей из объектов фиксированной сложности (элементов). Помимо важного прикладного значения для таких задач при проектирование цифровых схем, их исследование стало наиболее существенным вкладом теории автоматов в дискретную математику, поскольку в ее входе в первые было введено и подробно изучено понятие сложности. Это понятие, возникнувшие как обобщение естественной характеристики цифровой схемы - числа его элементов, постепенно становится одним из центральных понятий теории алгоритмов где многие количественные характеристики алгоритма, - память, быстродействие, объем собственного описания - являются различными аспектами его сложности. В этом отношение теория автоматов оказалось наиболее продвинутой ветвью теории алгоритмов.

Автоматы Мура.

1. Автомат Мура.
2. Представление событий в автоматах.

1.Автомат Мура

<u>Определение:</u> Конечный автомат называется автоматом Мура, если его функция выхода зависит только от состтояний, т.е. для любых q, a_i, a_j, $\lambda(q,a_i)=\lambda(q,a_j)$.

Функции выходов автоматов Мура естественно считать однооргументной функцией. Обычно ее обозначают функцией μ и называется функцией отметок (т.к. она каждому состоянию однозначно ставит в соответствие отметку выхода). В графе автомата Мура выход пишется не на ребрах, а при вершине. Общая модель автомата, которая рассматривалась ранее называлась автоматом Мили. Возможности этих двух видов автоматов совпадают:

Пример: для любого автомата Мили существует эквивалентный ему автомат Мура.

Пусть дан автомат Мили: $S(A,Q,V,\delta,\lambda)$, определена $A=\{a_1,\ldots,a_m\}$ и $Q=\{q_1,\ldots,q_n\}$. Определяем автомат Мура: S_m следующим образом. $A_m=A$, $V_m=V$.

Q_m содержит m*n+n состояний: m*n- это $q_{ij}(i=1,\ldots,n;\ j=1\ldots,m)$ это соответствует парам (q_i,a_j) автомата S (автомата Мили); n состояний $q_{i0}(i=1,\ldots,n)$. Функции $\delta м$ и μ определяется следующим образом: $\delta м(q_{i0},a_k)=q_{ik}$, для $i=1,\ldots,n$. $\delta м(q_{ij},a_k)=q_{lk}$, где l таково, что $\delta(q_i,a_k)=q_l$ $\mu(q_{io})$- определена. Для остальных состояний $\mu(q_{ij})=\lambda(q_i,a_j)$.

Для доказательства теоремы достаточно показать, что для любого состояния q_i и любого слова α автомата Мили $S(q_i,\alpha)=S_m(q_{io},\alpha)$. Данное доказательство проводится при помощи индукции по длине α. При исследовании возможностей автоматов достаточно пользоваться автоматами Мура. Это удобно потому, что автомат Мура можно рассматривать как автомат без выхода. Состояние которого различным образом отмечены. Можно считать, что этих отметок всего 2, (например 0 и 1) и они делят состояния на два класса, зафиксируем один из этих классов и будем называть его состояния заключительными. Это приводит к следующему определению автомата без выходов $S=(A,Q, \delta,q_1,F)$,где $F\subseteq Q$ называется множеством заключительных состояний.

2. Представление событий в автоматах.

<u>Определение:</u> Множество слов во входном алфавите называется событием или множеством слов или языком.

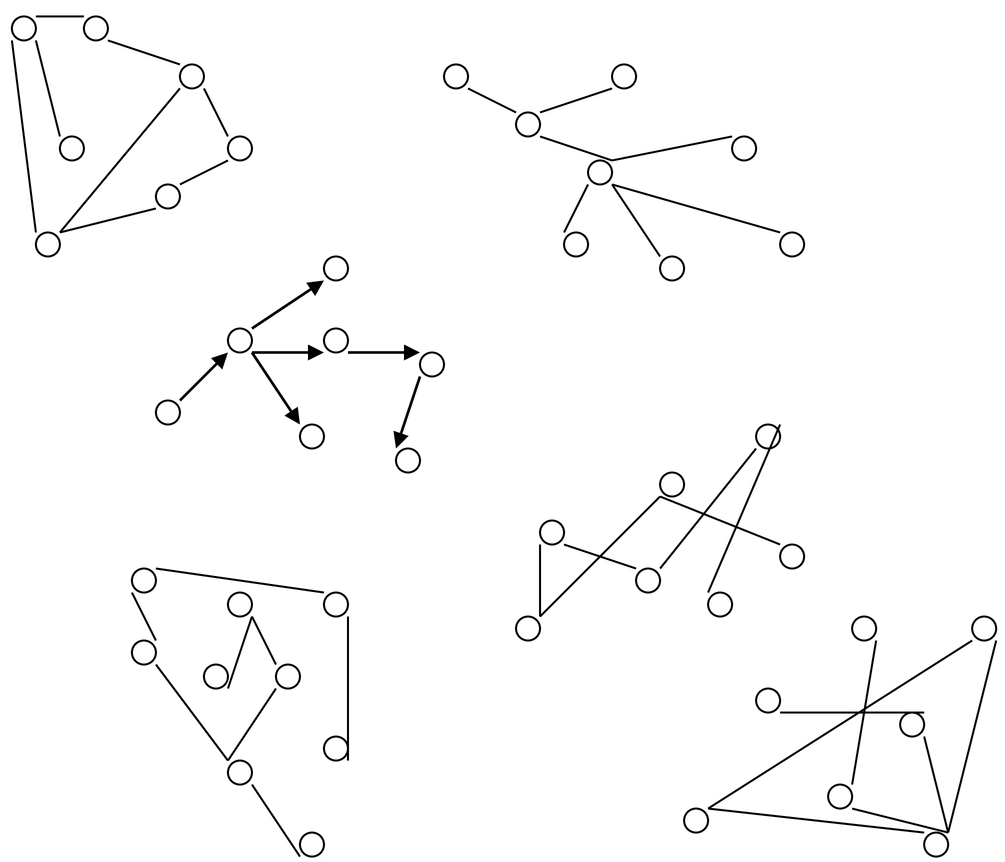
Событие $E\subseteq A^*$ представимо в автомате $S=(A,Q,\delta,q_1,F)$,если $\delta(q_1,\alpha)\in F$, тогда и только тогда, к $\alpha\in F$.
Всякому автомату (при фиксированных q_1и F) однозначно соответствует представимое в нем событие. На графе автомата это событие изображается множеством всех путей ведущих из q_1 в вершины из множества F.
<u>Определение:</u> Событие называется представимым в конечном автомате, если существует конечный автомат в котором оно представимо.

Представимое в автомате событие это конечно автоматный аналог разрешимого множества. Событие E представимое в автомате S_m можно было бы назвать множеством разрешимым автоматом S.
Может оказаться, что $q_1 \in F$ в этом случае автомат еще нечего не получив на входе уже ”что то представляет”. Удобно считать, что это “что то ” - пустое слово (слово нулевой длины); оно создается в событии представимом таким автоматом.
Пустое слово будем обозначать ***e***. Для любого слова α $\boldsymbol{e}*a=\alpha*\boldsymbol{e}=\alpha$, таким образом в свободном полугруппе слов входного алфавита, где умножением является приписывание слов друг другу ***e*** играет роль единицы.
Пустое слово не следует путать с пустым событием, т.е. с пустым множеством.
Автомат представляет пустое событие, если ни одно из его заключительных состояний не достижимо из начального состояния.

Вопросы для подготовки к зачету и экзамену:

1.Уметь определять виды маршрутов.
2.Знать основные определения рассмотренных тем.
3.Уметь работать с графами деревьями:
-определять типы вершин;
-определять цикломатическое число графа;
-уметь определять дивергенцию вершин ориентированного дерева;
-уметь определять истоки и стоки графа;
4.Уметь работать с плоскими и планарными графами: определять число граней, вершин, ребер.
5.Уметь находить максимальные части графов.
6.Уметь строить двойственные графы.
7.Уметь определять эйлеровы и гамильтоновы графы или пути.

У группы графов определить все возможные характеристики:



ЭЛЕМЕНТЫ АБСТРАКТНОЙ ТЕОРИИ АВТОМАТОВ

Определение абстрактного автомата

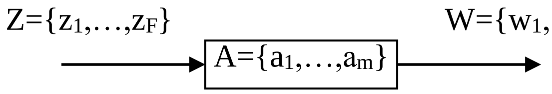
Абстрактный автомат является математической моделью дискретного управляющего устройства. Он задается множеством из шести элементов:

$S=\{A, Z, W, \delta, \lambda, a_1\}$,где

$A=\{a_1,\ldots,a_m,\ldots,a_M\}$ – множество состояний (алфавит состояний);
$Z=\{z_1,\ldots,z_f,\ldots,z_F\}$ – множество входных сигналов (входной алфавит);
$W=\{w_1,\ldots w_g,\ldots,w_G\}$ – множество выходных сигналов (выходной алфавит);
δ – функция переходов, реализующая отображение множества
$D_\delta \subseteq A\times Z$ в A $(a_s= \delta(a_m,z_f), a_s\in A)$;

λ – функция выходов, реализующая отображение множества
$D_\lambda \subseteq A\times Z$ на W $(w_g= \lambda(a_m,z_f))$;
$a_1 \in A$ – начальное состояние автомата.

Автомат называется конечным, если конечны множества A,Z, и W.
Автомат называется полностью определенным, если $D_\delta = D_\lambda \subseteq A\times Z$, т.е. область определения функций δ и λ совпадает со множеством всевозможных пар вида (a_m, z_f).
У частичного автомата функции δ или λ определено не для всех пар $(a_m, z_f)\in A\times Z$.
Понятие состояния в определении автомата введено в связи с тем, что часто возникает необходимость в описании поведения систем, выходы которых зависят не только от состояний входов в данный момент времени, но и от некоторой предыстории, т.е. от сигналов, которые поступали на системы ранее. Состояние как раз и соответствует некоторой памяти о прошлом, позволяя устранить время как явную переменную и выразить выходные сигналы как функцию состояний и входов в данный момент времени.



Абстрактный автомат имеет один входной и один выходной канал. В каждый момент времени t=0, 1, 2… дискретного времени автомат находится в определенном состоянии $a(t)\in A$. При t=0 он всегда находится в начальном состоянии $a(0)=a_1$. В момент t, будучи в состоянии a(t), автомат способен воспринять на входном канале сигнал $z(t)\in Z$ и выдать на выходном канале сигнал $w(t)=\lambda(a(t),z(t))$, переходя в состояние $a(t+1)=\delta(a(t),z(t))$, $a(t)\in A$, $w(t)\in W$.

Смысл понятия абстрактного автомата состоит в том, что он реализует некоторое отображение множества слов входного алфавита Z в множество слов выходного алфавита W.

На практике наибольшее распространение получили автоматы **Мили** и **Мура**.
Закон функционирования автомата Мили задается уравнениями:

$$a(t+1)= \delta(a(t),z(t));\ \ w(t)= \lambda(a(t),z(t)),\ t=0,1,2\ldots$$

Закон функционирования автомата Мура:

$$a(t+1)= \delta(a(t),z(t));\ \ w(t)= \lambda(a(t)),\ t=0,1,2\ldots$$

Методы задания автоматов

Чтобы задать конечный автомат S,необходимо описать все элементы множества

$S=\{A, Z, W, \delta, \lambda, a_1\}$.

Табличный способ задания.

	a_1	…	a_M
z_1	$\delta(a_1, z_1)$	…	$\delta(a_M, z_1)$
…	…	…	…
z_F	$\delta(a_1, z_F)$	…	$\delta(a_M, z_F)$

На пересечении столбца am и строки z_f ставится состояние $a_s=\delta(a_m, z_f)$, в которое автомат переходит из состояния a_m под действием сигнала z_f.

	a_1		a_M
z_1	$\delta(a_1, z_1)$		$\lambda(a_M, z_1)$
…	…		…
z_F	$\delta(a_1, z_F)$		$\lambda (a_M, z_F)$

На пересечении столбца a_m и строки z_f ставится выходной сигнал $w_y= \delta(a_m, z_f)$, соответствующий переходу автомата в состояние a_s .

Пример задания полностью определенного автомата Мили S_1 с тремя состояниями, двумя входными и двумя выходными сигналами.

Таблица переходов автомата Мили S_1

	a_1	a_2	a_3
z_1	a_3	a_1	a_1
z_2	a_1	a_3	a_2

Таблица выходов автомата Мили S_1

	a_1	a_2	a_3
z_1	w_1	w_1	w_2
z_2	w_1	w_2	w_1

Пример задания частично определенного автомата Мили.

Таблица переходов частичного автомата Мили S_2

	a_1	a_2	a_3	a_4
z_1	a_2	a_3	a_4	-
z_2	a_3	-	a_2	a_2

Таблица выходов частичного автомата Мили S_2

	a_1	a_2	a_3	a_4
z_1	w_1	w_3	w_3	-
z_2	w_2	-	w_1	w_2

Так как в автомате Мура выходной сигнал зависит только от состояния, автомат Мура задается одной отмеченной таблицей переходов, в которой каждому столбцу приписан, кроме состояния a_m, ещё и выходной сигнал $w_g= \lambda(a_m)$, соответствующий этому состоянию

Общий вид отмеченной таблицы переходов автомата Мура

	$\lambda(a_1)$	…	$\lambda(a_1)$
	a_1	…	a_M
z_1	$\delta(a_1,z_1)$	…	$\delta(a_M,z_1)$
…	…	…	…
z_f	$\delta(a_1,z_F)$	…	$\delta(a_M,z_F)$

Отмеченная таблица переходов автомата Мура S_3

	w_1	w_1	w_3	w_2	w_3
	a_1	a_2	a_3	a_4	a_5
z_1	a_2	a_5	a_5	a_3	a_3
z_2	a_4	a_2	a_2	a_1	a_1

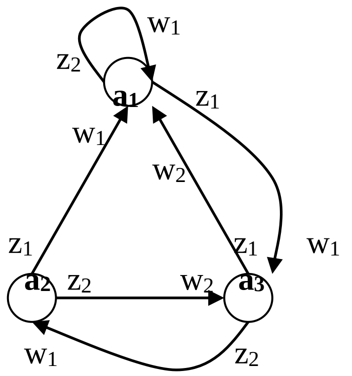
Графический способ задания

Граф автомата – это ориентированный связный граф, вершины которого соответствуют состояниям, а дуги – переходам между ними.

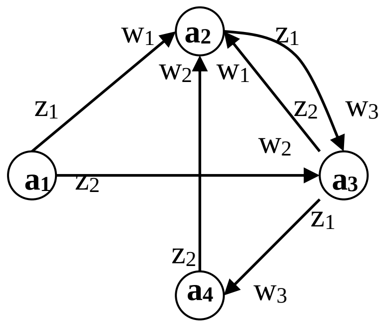
Две вершины графа автомата a_m и a_s (исходное состояние и состояние перехода) соединяются дугой, направленной от a_m к a_s, если в автомате есть переход от a_m к a_s, т.е. Если $a_s=\delta(a_m,z_f)$ при некотором входном сигнале $z_f \in Z$.

Дуге (a_m,a_s) графа автомата приписывается входной сигнал z_f и выходной сигнал $w_g=\lambda(a_m,z_f)$, если он определен и ставится прочерк в противном случае.

Граф автомата Мили S_1

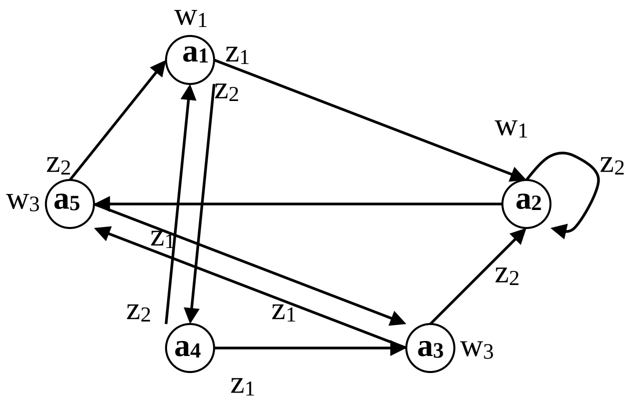


Граф автомата Мили S_2



При описании автомата Мура в виде графа выходной сигнал $w_g=\lambda(a_m)$ записывается внутри вершины a_m или рядом с ней.

Граф автомата Мура S_3



Автомат называется детерминированным, если, находясь в некотором состоянии под действием любого входного сигнала, он не может перейти более чем в одно состояние. Будем рассматривать только детерминированные автоматы.

Состояние a_s автомата S называется устойчивым состоянием, если для любого входа $z_f \in Z$, такого, что $\delta(a_m, z_f)=a_s$, имеет место $\delta(a_s, z_f)=a_s$.

Автомат S называется асинхронным, если каждое его состояние $a_s \in A$ устойчиво.

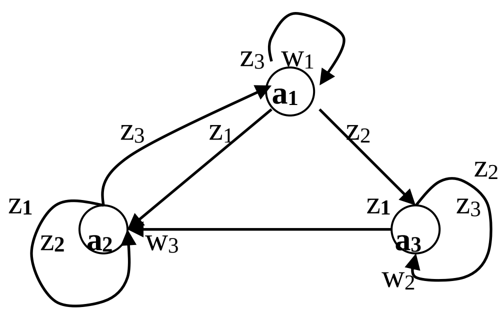
Автомат S называется синхронным, если он не является асинхронным.

Построенные на практике автоматы – всегда асинхронные. Устойчивость их состояния всегда обеспечивается тем или иным способом, например, введением сигналов синхронизации. Однако на уровне абстрактной теории, когда автомат есть лишь математическая модель, которая не отражает многих конкретных особенностей его возможной реализации, часто оказывается более удобно ориентировать с синхронными автоматами.

Отмеченная таблица переходов асинхронного автомата Мура S_4

	w_1	w_3	w_2
	a_1	a_2	a_3
z_1	a_2	a_2	a_2
z_2	a_3	a_2	a_3
z_3	a_1	a_1	a_3

Граф асинхронного автомата Мура S_4



Если в состояние a_s имеется переход из другого состояния под действием входного сигнала z_f, то в a_s должна быть петля, отмеченная символом z_f.

Связь между моделями Мили и Мура.

Выше отмечалось, что абстрактный автомат работает как преобразователь слов входного алфавита в слова в выходного алфавита. Возьмем, например, автомат Мили S_1, установленный в начальное состояние, и подадим на него входное слово $\xi=z_1z_1z_2z_1z_2z_2$. Получим

Входное слово:	z_1	z_1	z_2	z_1	z_2	z_2.	
Последовательность состояний:	a_1	a_3	a_1	a_1	a_3	a_2	a_3
Выходное слово:	w_1	w_2	w_1	w_1	w_1	w_2	

Таким образом, в ответ на входное слово длины k автомат Мили выдает последовательность состояний длины k+1 и выходное слово длины k.

В общем, виде поведение автомата Мили, установленного в состояние a_m можно описать следующим образом:

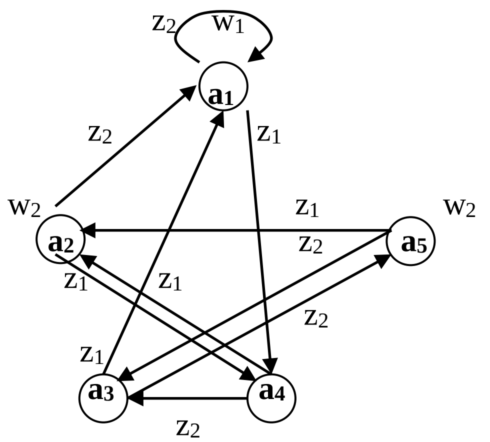
Входное слово:	z_{i1}	z_{i2}	z_{i3}
Последовательность состояний:	a_m	$a_{i2}=\delta(a_m, z_{i1})$	$a_{i3}=\delta(a_{i2}, z_{i2})$
Выходное слово:	$w_{i1}=\lambda(a_m, z_{i1})$	$w_{i2}=\lambda(a_{i2}, z_{i2})$	$w_{i3}=\lambda(a_{i3}, z_{i3})$

Аналогично опишем поведение автомата Мура, находящегося в состоянии a_m, при переходе входного слова $z_{i1}z_{i2}\ldots z_{ik}$.

Входное слово:	z_{i1}	z_{i2}	z_{i3}	z_{i4}
Последовательность состояний:	a_m	$a_{i2}=\delta(a_m, z_{i1})$	$a_{i3}=\delta(a_{i2}, z_{i2})$	$a_{i4}=\delta(a_{i3}, z_{i3})$
Выходное слово:	$w_{i1}=\lambda(a_m)$	$w_{i2}=\lambda(a_{i2})$	$w_{i3}=\lambda(a_{i3})$	$w_{i4}=\lambda(a_{i4})$

Очевидно, что выходной сигнал $w_{i1}=\lambda(a_m)$ в момент времени i_1 не зависит от входного сигнала z_{i1}, а определяется только состоянием a_m. Таким образом, сигнал w_{i1} никак не связан с входным словом, поступающим на вход автомата, начиная с момента i_1. В связи с этим под реакцией автомата Мура, установленного в состояние a_m, на входное слово $\xi=z_{i1}z_{i2}\ldots z_{ik}$ будем понимать выходное слово той же длины $W=\lambda(a_m,\xi)=w_{i2}\,w_{i3}\ldots w_{ik+1}$.

Например, рассмотрим автомат Мура S_5.



Найдем реакцию автомата S_5 на входное слово $\xi=z_1z_1z_2z_1z_2z_2$, то есть на то слово, которое подавалось на вход автомата Мили S_1

Входное слово:	z_1	z_1	z_2	z_1	z_2	z_2.	
Последовательность состояний:	a_1	a_4	a_2	a_1	a_4	a_3	a_5
Выходное слово:	w_1	w_1	w_2	w_1	w_1	w_1	w_2

Отсюда следует, что реакции автоматов S_1 и S_5 начальном состоянии на входное слово ξ совпадают с точностью до сдвига на 1 такт.

Эквивалентность автоматов.

Два автомата S_A и S_B с одинаковыми входными и выходными алфавитами называются эквивалентными, если после установления их в начальные состояния их реакции на любое входное слово совпадают.

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

Далее будет показано, что автомат S_5 эквивалентен автомату S_1

Преобразование автомата Мура в автомат Мили.

Пусть дан автомат Мура $S_A=\{A_A, Z_A, W_A, \delta_A, \lambda_A, a_{1A}\}$, где $A_A=\{a_1,\ldots,a_m,\ldots,a_M\}$, $Z_A=\{z_1,\ldots,z_f,\ldots,z_F\}$, $W_A=\{w_1,\ldots,w_g,\ldots w_G\}$;

δ_A реализует отображение $A_A \times Z_A$ в A_A

λ_A реализует отображение $A_A \times Z_A$ на W_A

$a_{1A} = a_1$ – начальное состояние.

Построим автомат Мили $S_B=\{A_B, Z_B, W_B, \delta_B, \lambda_B, a_{1B}\}$, у которого

$A_B = A_A=\{a_1,\dots,a_m,\dots,a_M\}$,

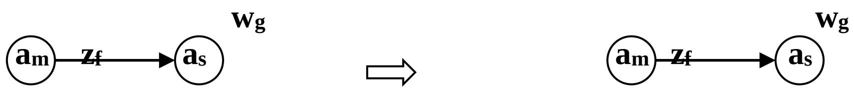
$Z_B = Z_A=\{z_1,\dots,z_f,\dots,z_F\}$,

$W_B = W_A=\{w_1,\dots,w_g,\dots w_G\}$;

$\delta_B = \delta_A$, $a_{1B} = a_{1A} = a_1$.

Функцию выходов λ_B определим таким образом. Если в автомате Мура $\delta_A(a_m, z_f)=a_s$ и $\lambda_A(a_s) = w_g$ то в автомате Мили $\lambda_B(a_m,z_f)= w_g$

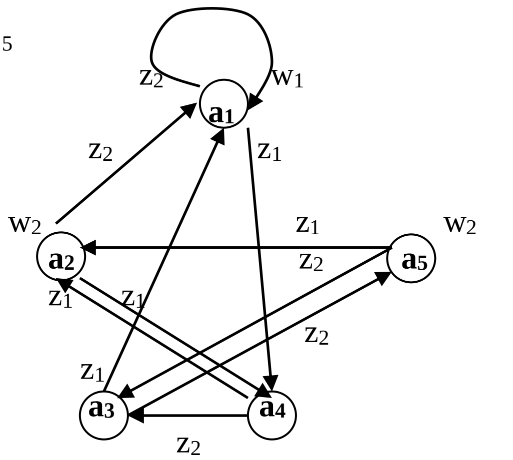
Графическая иллюстрация определения λ_B



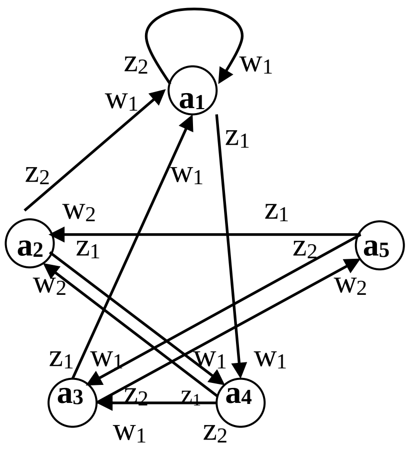
Выходной сигнал w_g, записанный рядом с вершиной a_s в автомате Мура, переносится на все дуги, входящие в a_s в автомате.

Пример. Преобразование автомата Мура S_5 в эквивалентный автомат Мили S_6.

S_5



S_6



При табличном способе задания автомата S_A таблица переходов автомата S_B совпадает с таблицей переходов автомата S_A. Таблица же выходов получается из таблицы переходов заменой символа a_s, стоящего на пересечении строки z_f и столбца a_m, символом выходного сигнала w_g, отмечающего столбец a_s

Отмеченная таблица переходов автомата Мура S_5

	w_1	w_2	w_1	w_1	w_2
	a_1	a_2	a_3	a_4	a_5
z_1	a_1	a_1	a_1	a_2	a_2
z_2	a_1	a_1	a_5	a_3	a_3

Таблица перехода
эквивалентного автомата Мили S_6

	a_1	a_2	a_3	a_4	a_5
z_1	a_4	a_4	a_1	a_2	a_2
z_2	a_1	a_1	a_5	a_3	a_3

Таблица выходов
эквивалентного автомата Мили S_6

	a_1	a_2	a_3	a_4	a_5
z_1	w_1	w_1	w_1	w_2	w_2
z_2	w_1	w_1	w_2	w_1	w_1

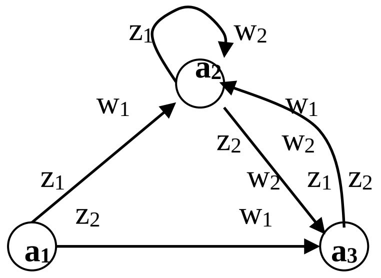
Эквивалентность автоматов следует из самого способа построения автомата Мили. Одинаковые входные последовательности символов они будут переводить в одни и те же выходные последовательности. Подавая на вход автомата S_6 входное слово $\xi=z_1z_1z_2z_1z_2z_2$, получим то же самое выходное слово, как и в автомате S_1.

Входное слово: z_1 z_1 z_2 z_1 z_2 z_2.
Последовательность состояний: a_1 a_4 a_2 a_1 a_4 a_3 a_5
Выходное слово: w_1 w_2 w_1 w_1 w_1 w_2

Преобразование автомата Мили в автомат Мура.

Ограничение: у автомата Мили не должно быть преходящих состояний, т. е. состояний которые не входит ни одна дуга, но из которого есть исходящие дуги.

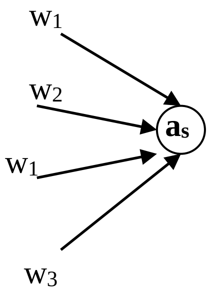
Например, у автомата Мили S_7 преходящим S_7 является состояние a_1.



Задан автомат Мили:
$S_A=\{A_A, Z_A, W_A, \delta_A, \lambda_A, a_{1A}\}$, где $A_A=\{a_1,\ldots,a_m,\ldots,a_M\}$,
$Z_A=\{z_1,\ldots,z_f,\ldots,z_F\}$, $W_A=\{w_1,\ldots,w_g,\ldots w_G\}$;
δ_A реализует отображение A_A x Z_A в A_A
λ_A реализует отображение A_A x Z_A на W_A
$a_{1A} = a_1$ – начальное состояние.
Требуется построить автомат Мура:
$S_B=\{A_B, Z_B, W_B, \delta_B, \lambda_B, a_{1B}\}$, у которого
$A_B = A_A=\{a_1,\ldots,a_m,\ldots,a_M\}$,
$Z_B = Z_A=\{z_1,\ldots,z_f,\ldots,z_F\}$,

$W_B = W_A = \{w_1,\ldots,w_g,\ldots w_G\}$.

Для определения Ав каждому состоянию $a_s \in A_A$, поставим в соответствие множество A_s
всевозможных пар вида (a_s,w_g), где w_g – выходной сигнал, приписанный входящей в a_s дуге: $A_s = \{(a_s,w_g) \mid \delta(a_m, z_f)=a_s$ и $\lambda(a_m, z_f)=w_g\}$



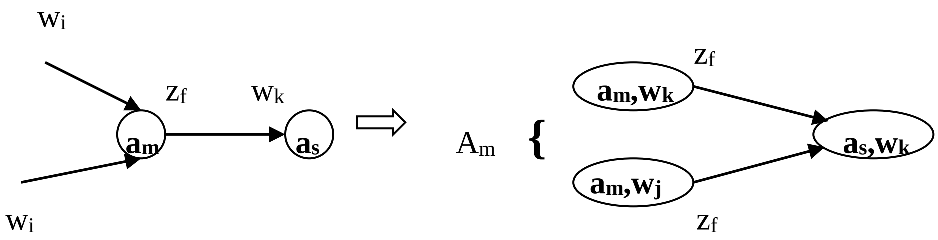
$A_s=\{(a_s, w_1)\ (a_s, w_2)(a_s, w_3)\}$

Вместо одного состояния a_s в автомате Мура будет три состояния.
Множество состояний автомата S_B получим как объединение множеств

$A_S(S=1,\ldots,M)$: $\quad A_B = \bigcup_{S=1}^{M} A_s$

***Определение функции выходов** λ_B:*
Каждому состоянию автомата Мура S_B , представляющему собой пару вида (a_s,w_g), ставится в соответствие выходной сигнал w_g.

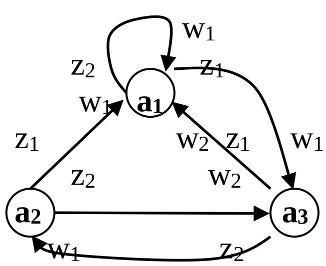
Определение функции переходов:
Если в автомате Мили S_A был переход $\delta_A(a_m, z_f)=a_s$ и при этом выдавался выходной сигнал $\lambda_A(a_m, z_f)=w_k$, то вS_B будет переход из множества состояний A_m , порождаемых a_m , в состояние (a_s, w_k) под действием входного сигнала z_f.



В качестве начального состояния a_{1B} можно взять любое из состояний множества A_1, которое порождается начальным состоянием a_1 автомата S_A.

Пример: Дан автомат Мили S_1

S_1



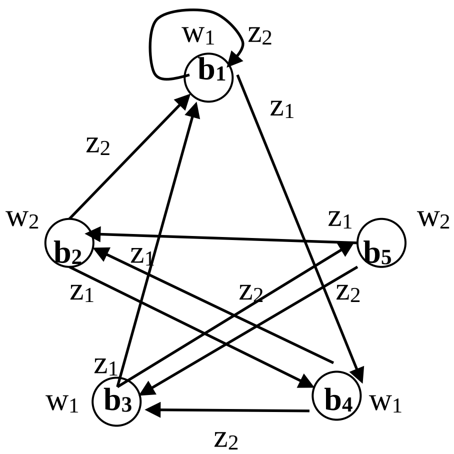
	a_1	a_2	a_3
z_1	a_3	a_1	a_1
z_2	a_1	a_3	a_2

	a_1	a_2	a_3
z_1	w_1	w_1	w_2
z_2	w_1	w_2	w_1

$A_A=\{a_1,a_2,a_3\}$, $Z_A=\{z_1,z_2\}$, $a_{1A}=a_1$.
Функции δ и λ определяются графом автомата либо таблицами переходов и выходов.
Требуется построить эквивалентный ему автомат Мура.
Получим
$Z_B=Z_A=\{z_1,z_2\}$, $W_B=W_A=\{w_1,w_2\}$
Построим множество состояний A_B:
$A_1=\{(a_1,w_1), (a_1,w_2)\}=\{b_1,b_2\}$.
$A_2=\{(a_2,w_1)\}=\{b_3\}$
$A_3=\{(a_3,w_1),(a_3,w_2)\}=\{b_4,b_5\}$.
$A_B=A_1 \cup A_2 \cup A_3= \{b_1,b_2,b_3,b_4,b_5\}$. $a_{1B}=b_1$.

С каждым состоянием, представляющим собой пару, отождествим выходной сигнал, являющийся вторым элементом пары:
$\lambda_B(b_1)= \lambda(b_3)=\lambda(b_4)=w_1$; $\lambda_B(b_2)= \lambda(b_5)=w_2$.

S_8

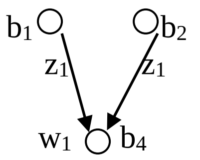


Построим функцию δ_B.

1) В автомате Мили: $a_3= \delta(a_1,z_1)$
$w_1=\lambda(a_1,z_1)$

В автомате Мура: $b_4=\delta(b_1,z_1)$
$b_4=\delta(b_2,z_1)$
$w_1= \lambda(b_4)$

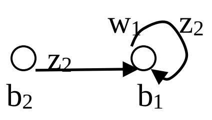
$a_1 \rightarrow b_1,b_2$
$(a_3,w_1) \rightarrow b_4$ ⇨



2) В автомате Мили: $a_1= \delta(a_1,z_2)$
$w_1=\lambda(a_1,z_2)$

В автомате Мура: $b_1=\delta(b_1,z_2)$
$b_1=\delta(b_2,z_2)$
$w_1= \lambda(b_1)$

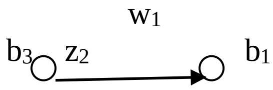
$a_1 \rightarrow b_1,b_2$
$(a_1,w_1) \rightarrow b_1$ ⇨



3) В автомате Мили: $a_1= \delta(a_2,z_1)$
$w_1=\lambda(a_2,z_1)$

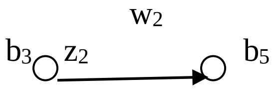
В автомате Мура: $b_1=\delta(b_3,z_2)$
$w_1= \lambda(b_1)$

$a_2 \to b_3$
$(a_1,w_1) \to b_1$ $\Rightarrow$



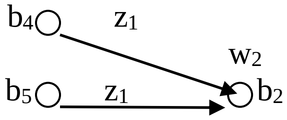
4) В автомате Мили $a_3= \delta(a_2,z_2)$ В автомате Мура: $b_5=\delta(b_3,z_2)$
$w_2=\lambda(a_2,z_2)$ $w_2= \lambda(b_5)$

$a_2 \to b_3$
$(a_3,w_2) \to b_5$ $\Rightarrow$



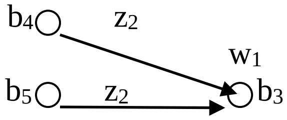
5) В автомате Мили $a_1= \delta(a_3,z_1)$ В автомате Мура: $b_2=\delta(b_4,z_1)$
$w_2=\lambda(a_3,z_1)$ $b_2=\delta(b_5,z_1)$
$w_2= \lambda(b_2)$

$a_3 \to b_4, b_5$ $\Rightarrow$
$(a_1,w_2) \to b_2$



6) В автомате Мили $a_2= \delta(a_3,z_2)$ В автомате Мура: $b_3=\delta(b_4,z_2)$
$w_1=\lambda(a_3,z_2)$ $b_3=\delta(b_5,z_2)$
$w_1= \lambda(b_3)$

$a_3 \to b_4, b_5$ $\Rightarrow$
$(a_2,w_1) \to b_3$



Отмеченная таблица переходов автомата S_8 будет иметь вид:

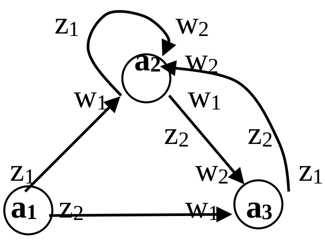
	w_1	w_2	w_1	w_1	w_2
	b_1	b_2	b_3	b_4	b_5
z_1	b_4	b_4	b_1	b_2	b_2
z_2	b_1	b_1	b_5	b_3	b_3

Примем за начальное состояние b_1.
Подавая на вход автомата S_8 входное слово $\xi=z_1z_1z_2z_1z_2z_2$, получим выходное слово как в автоматах Мили S_1 и S_2 со сдвигом на один такт.

Входное слово: z_1 z_1 z_2 z_1 z_2 z_2.
Последовательность состояний: b_1 b_4 b_2 b_1 b_4 b_3 b_5
Выходное слово: w_1 $\boxed{w_1\ w_2\ w_1\ w_1\ w_1\ w_2}$

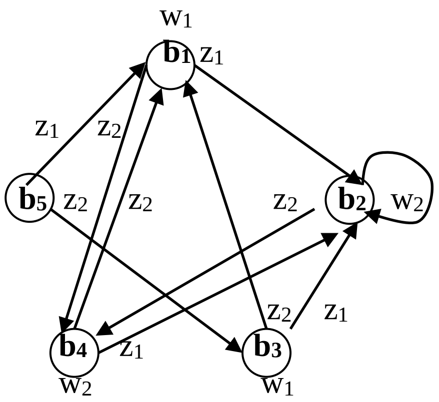
Заменив b_i на a_i, $i=1,\dots,5$, получим автомат Мура S_5
Снимем с автомата Мили ограничение на наличие преходящих состояний.

Рассмотрим пример: Дан автомат Мили S_7
$Z_A= Z_B=\{z_1,z_2\}$, $W_A= W_B=\{w_1,w_2\}$ $a_{1A}=a_1$.
Построим множество состояний A_B:
$A_1=\{\ \}$.
$A_2=\{(a_2,w_1), (a_2,w_2)\}=\{b_1,b_2\}$
$A_3=\{(a_3,w_1),(a_3,w_2)\}=\{b_3,b_4\}$ a_1

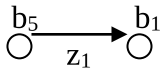


К множеству состояний $A_1 \cup A_2 \cup A_3 = \{b_1, b_2, b_3, b_4\}$ добавим состояние $(a_1, -) = b_5$, порождаемое переходящим состоянием a_1, считая, что выходной сигнал в состоянии a_1 не определен $a_{1B} = b_1$.

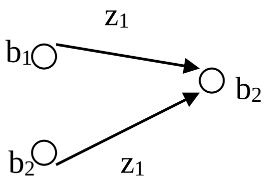
S_9



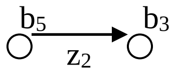
1) $a_2 = \delta(a_1, z_1)$
$w_1 = \lambda(a_1, z_1)$ ⇨



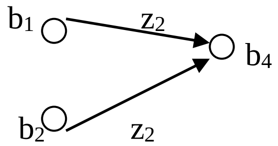
2) $a_2 = \delta(a_2, z_1)$
$w_2 = \lambda(a_2, z_1)$ ⇨



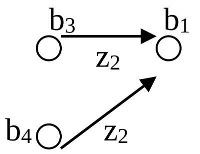
3) $a_3 = \delta(a_1, z_2)$
$w_1 = \lambda(a_1, z_2)$ ⇨



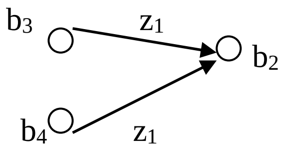
4) $a_3 = \delta(a_2, z_2)$
$w_2 = \lambda(a_2, z_2)$ ⇨



5) $a_2 = \delta(a_3, z_2)$
$w_1 = \lambda(a_3, z_2)$ ⇨



6) $a_2 = \delta(a_3, z_1)$
$w_2 = \lambda(a_3, z_1)$ ⇨



В итоге получим автомат Мура S_9, отмеченная таблица переходов которого будет иметь вид:

	w_1	w_2	w_1	w_2	-
	b_1	b_2	b_3	b_4	b_5
z_1	b_2	b_2	b_2	b_2	b_1
z_2	b_4	b_4	b_1	b_1	b_3

Из вышесказанного следует, что при переходе от автомата Мура к автомату Мили число состояний автоматов не меняется, в то время как при обратном переходе число состояний в автомате Мура, как правило, возрастает. Таким образом, если перейти от автомата Мили к автомату Мура, а затем обратно, то получим два эквивалентных автомата Мили с различным числом состояний.