ДИСКРЕТНАЯ МАТЕМАТИКА

ИУ5 - 4 семестр, 2015 г.

Лекция 13. РЕГУЛЯРНЫЕ ЯЗЫКИ И КОНЕЧНЫЕ АВТОМАТЫ

Формальный язык.

Термин „формальный язык“ обычно означает искусственный язык, придуманный людьми для специальных целей, например — язык программирования.

Непреодолимой преграды между специально придуманными искусственными (формальными) языками и стихийно возникающими и развивающимися естественными языками нет.

Естественные языки характеризуются сложными грамматическими правилами, т.е. довольно жестко формализованы, а специально разработанный язык программирования содержит „темные места“, однозначное понимание которых является проблемой.

Основные аспекты изучения языков.

1. Синтаксис языка.

Слово „синтаксис“ происходит от древнегреческих слов „syn“ — „вместе“ и „taxis“ — „порядок, строй“.

Язык — это какое-то множество „слов“, где „слово“ есть определенная конечная последовательность „букв“ — символов какого-то заранее фиксированного алфавита.

Термины „буква“ и „слово“ могут пониматься по-разному (математическое определение этих терминов будет дано позже).

„Буквами“ могут быть действительно буквы алфавита какого-нибудь естественного или формального языка, например русского языка или языка программирования „Паскаль“.
„Словами“ будут конечные последовательности „букв“.
Но „буквой“ может быть и некоторое „слово“ целиком. Например: „if“ „then“ „else“ Тогда „слова“ — это предложения языка программирования: „if a then b else c.
Если фиксировано какое-то множество „букв“, то не каждая их последовательность будет „словом“, а только такая последовательность, которая подчиняется определенным правилам.

Синтаксис языка и представляет собой систему правил, в соответствии с которыми можно строить „правильные“ последовательности „букв“. Каждое слово языка характеризуется определенной структурой, специфичной именно для данного языка.

Необходимо разработать механизмы порождения слов с заданной структурой и механизмы проверки того, что данное слово принадлежит данному языку.

Именно эти механизмы и изучает классическая теория формальных языков.

2. Семантика языка.

Семантика („semanticos“ — „обозначающий“) предполагает сопоставление словам языка некоего „смысла“, „значения“. Например, записывая математическую формулу, мы должны соблюдать определенные синтаксические правила (расстановка скобок, правописание символов, порядок символов и т.п.), но, кроме этого, формула имеет вполне определенный смысл, что-то обозначает.

Обсуждение семантики выходит за рамки курса.

13.1. Алфавит, слово, язык

Алфавит — это произвольное *непустое конечное множество* $V = \{a_1, \ldots, a_n\}$, *элементы* которого называют **буквами** или **символами**.

Определение 13.1. **Словом** или **цепочкой** в алфавите V называют произвольный кортеж из множества V^k (k-й *декартовой степени* алфавита V) для различных $k = 0, 1, 2, \ldots$

Например, если $V = \{a, b, c\}$, то (a), (b), (c), (a, b), (a, b, c), (c, b, a, a, c) и т.д. есть слова в алфавите V.

При $k = 0$ получаем **пустой кортеж**, называемый **пустым словом** или **пустой цепочкой** и обозначаемый λ.
Пустое слово — это слово, не имеющее символов.
Слова будем записывать без скобок и запятых. Такая запись согласуется с пониманием слова как цепочки следующих друг за другом символов.
Например: a, b, c, ab, abc, $cbaac$.
Длина слова w — число компонент кортежа, т.е. если $w \in V^r$, то длина слова w равна r. Длину слова можно понимать как число составляющих это слово букв.
Длину слова w обозначим $|w|$.
Для пустого слова $|\lambda| = 0$.

Будем использовать следующую запись непустого слова x в алфавите V по буквам:

$$x = x(1)x(2)\dots x(k),$$

где $x(i)$, $1 \le i \le k$, — i-я буква слова x.
Множество всех слов в алфавите V обозначают V^*, а множество всех непустых слов в V — как V^+.

Определение 13.2. **Языком в алфавите** V называется произвольное подмножество множества V^*.

Определим на множестве 2^{V^*} всех языков в произвольном (но фиксированном!) алфавите V **алгебраическую структуру**.

Языки — это множества, следовательно, над ними можно производить все теоретико-множественные операции: **объединение, пересечение, разность, дополнение** и т.п. **Универсальное множество** в данном случае есть множество слов V^*, которое называют **универсальным языком**.

Специальные операции над языками.

Определение 13.3. **Соединением** или **конкатенацией** слов $x = x(1)x(2)\dots x(k)$ и $y = y(1)y(2)\dots y(m)$ называют слово

$$xy = x(1)x(2)\dots x(k)y(1)y(2)\dots y(m).$$

Соединение иногда обозначают точкой ($\cdot$).
Из определения следует, что соединение слов **ассоциативно**, т.е. для произвольных трех слов x , y , z имеет место $x(yz) = (xy)z$,
и $x\lambda = \lambda x = x$ для любого x .

Множество V^* всех слов в алфавите V с операцией соединения образует **моноид** $(V^*, \cdot, \lambda)$. **Единица моноида** — пустое слово. Этот моноид есть **свободный моноид**, порожденный алфавитом V. Для него используют то же обозначение, что и для множества всех слов в алфавите V, т.е. V^*.

Неформально соединение xy получается приписыванием слова y справа к слову x. Для любых двух слов $x \in V^k$ и $y \in V^m$ конкатенация $xy \in V^{k+m}$ ($k, m \geq 0$). Следовательно, $|xy| = |x| + |y|$.

Определение 13.4. Соединением (конкатенацией) языков L_1 и L_2 называют язык L_1L_2, состоящий из всех возможных соединений слов xy, в которых слово x принадлежит первому, а слово y — второму языку, т.е.

$$L_1L_2 = \{xy\colon x \in L_1 \text{ и } y \in L_2\}\,.$$

Соединение конечных языков легко вычислить.

Пример 13.1. $V = \{a,\, b,\, c\}$, $L_1 = \{ab,\, bcc,\, cab\}$, $L_2 = \{ca,\, bcc\}$

$$\begin{aligned} L_1L_2 &= \{abca,\; abbcc,\; bccca,\; bccbcc,\; cabca,\; cabbcc\},\\ L_2L_1 &= \{caab,\; cabcc,\; cacab,\; bccab,\; bccbcc,\; bcccab\}. \end{aligned}$$

Можно формально написать так:

$$(ab + bcc + cab)(ca + bcc) =$$
$$= abca + abbcc + bccca + bccbcc + cabca + cabbcc.$$

Соединение языков не коммутативно: $L_1L_2 \neq L_2L_1$. В общем случае $L_1L_2 \cap L_2L_1$ не пусто. В примере 13.1 пересечение равно $\{bccbcc\}$.

Операция соединения языков позволяет определить операцию возведения языка в произвольную натуральную степень.

По определению, $L^0 = \{\lambda\}$ для любого $L \subseteq V^*$, а $L^n = L^{n-1}L$ при $n > 1$.

Итерацией языка L называют объединение всех его степеней:

$$L^* = \bigcup_{n=0}^{\infty} L^n.$$

Рассматривая объединение всех степеней языка L, начиная с первой, получим **позитивную итерацию**

$$L^+ = \bigcup_{n=1}^{\infty} L^n.$$

Основное алгебраическое свойство множества всех языков в алфавите V .

Теорема 1. Алгебра $\mathcal{L}(V) = \left(2^{V^*}, \cup, \cdot, \varnothing, \{\lambda\}\right)$ есть **замкнутое полукольцо**.

◀ Необходимо проверить **аксиомы полукольца**, т.е. показать, что:

1) алгебра $\left(2^{V^*}, \cup, \varnothing\right)$ — **коммутативный** и **идемпотентный моноид**;

2) алгебра $\left(2^{V^*}, \cdot, \{\lambda\}\right)$ — моноид;

3) операция соединения дистрибутивна относительно объединения;

4) выполняется аннулирующее свойство нуля.

1. Рассмотрим алгебру $(2^{V^*}, \cup)$, где 2^{V^*} — множество всех языков.

Пусть L_1, L_2, L_3 — произвольные подмножества множества 2^{V^*}, т.е. произвольные языки в алфавите V.

Ассоциативность ($L_1 \cup (L_2 \cup L_3) = (L_1 \cup L_2) \cup L_3$),
коммутативность ($L_1 \cup L_2 = L_2 \cup L_1$) и
идемпотентность ($L_1 \cup L_1 = L_1$) операции $\cup$ доказана ранее
(см. свойства операции объединения множеств).
Нейтральный элемент по операции объединения — пустое множество ($\varnothing$). Этот элемент есть **нуль полукольца**.

Следовательно, по операции объединения множество языков в алфавите V образует **коммутативный** и **идемпотентный моноид**.

2. Рассмотрим алгебру $(2^{V^*}, \cdot)$ и покажем, что по операции соединения множество языков образует **моноид**.

Используя **метод двух включений**, докажем ассоциативность операции соединения языков:

$(L_1 \cdot L_2) \cdot L_3 = L_1 \cdot (L_2 \cdot L_3)$.

Пусть слово w принадлежит языку $(L_1 \cdot L_2) \cdot L_3$. Тогда согласно определению соединения языков, слово w может быть представлено в виде $w = vz$, где $v \in L_1 \cdot L_2$ и $z \in L_3$.

Слово v может быть представлено в виде $v = xy$, где $x \in L_1$ и $y \in L_2$. Следовательно, w может быть представлено в виде $w = (xy)z$.

Согласно определению операции соединения слов и в силу ее ассоциативности имеем $w = (xy)z = x(yz)$. Слово $w = x(yz)$, согласно определению соединения языков, принадлежит языку $L_1 \cdot (L_2 \cdot L_3)$.

Обратно:

$$w \in L_1 \cdot (L_2 \cdot L_3) \Rightarrow w = xu, \text{ где } x \in L_1, u \in (L_2 \cdot L_3) \Rightarrow$$
$$\Rightarrow u = yz, \text{ где } y \in L_2, z \in L_3 \Rightarrow w = x(yz) = (xy)z \Rightarrow$$
$$\Rightarrow w \in (L_1 \cdot L_2) \cdot L_3.$$

Рассмотрим пустой язык $\{\lambda\}$, состоящий из одного пустого слова. Для любого языка L выполняется тождество $\{\lambda\}L = L\{\lambda\} = L$. Это следует из тождества $\lambda x = x\lambda = x$ для любого слова x.
Следовательно, пустой язык $\{\lambda\}$ есть нейтральный элемент по операции соединения языков — **единица полукольца**.

3. Свойство **дистрибутивности** операции соединения относительно объединения имеет вид:

$$L_1 \cdot (L_2 \cup L_3) = L_1 \cdot L_2 \cup L_1 \cdot L_3, \qquad (1)$$

$$(L_1 \cup L_2) \cdot L_3 = L_1 \cdot L_3 \cup L_2 \cdot L_3 \qquad (2)$$

Докажем тождество (1).

Пусть слово x принадлежит языку $L_1 \cdot (L_2 \cup L_3)$.

Тогда, согласно определению соединения языков, это слово может быть представлено в виде $x = yz$, где $y \in L_1$, а $z \in L_2 \cup L_3$, т.е. $z \in L_2$ или $z \in L_3$.

Если $z \in L_2$, то $yz \in L_1 \cdot L_2$, а если $z \in L_3$, то $yz \in L_1 \cdot L_3$, т.е. $x = yz \in L_1 \cdot L_2 \cup L_1 \cdot L_3$.

Пусть теперь $x \in L_1 \cdot L_2 \cup L_1 \cdot L_3$. Тогда $x = yz$, где $y \in L_1$, а $z \in L_2$ или $z \in L_3$, т.е. $x \in L_1 \cdot (L_2 \cup L_3)$.

Первое тождество доказано.

Тождество (2) доказывается аналогично.

4) Аннулирующее свойство нуля $L \cdot \varnothing = \varnothing \cdot L = \varnothing$ выполняется в силу определения операции соединения языков.

Замкнутость полукольца $\mathcal{L}(V)$ всех языков в алфавите V .

Вспомним определение.

Полукольцо $\mathcal{S} = (S, +, \cdot, \mathbf{0}, \mathbf{1})$ называют **замкнутым**, если:

1) оно идемпотентно;

2) любая последовательность элементов множества S имеет точную верхнюю грань относительно естественного порядка $\leq$ этого идемпотентного полукольца;

3) операция умножения полукольца $\mathcal{S}$ сохраняет точные верхние грани последовательностей, т.е. для любого $a \in S$ и любой последовательности $X = \{x_n\}_{n\in\mathbb{N}}$ элементов множества S

$$a \sup X = \sup aX, \quad (\sup X)a = \sup(Xa).$$

В полукольце $\mathcal{S}$ **отношение порядка** вводится следующим образом: для любых $x, y \in S$ по определению полагают $x \leq y$ тогда и только тогда, когда $x + y = y$.
В полукольце всех языков в алфавите V операция **сложения** — это операция объединения множеств и отношение порядка $\leq$ есть отношение **теоретико-множественного включения** $\subseteq$ (включение $L_1 \subseteq L_2$ равносильно тому, что $L_1 \cup L_2 = L_2$).

Замкнутость полукольца $\mathcal{L}(V)$ следует из существования **объединения** любого **семейства множеств** (в частности, бесконечной последовательности множеств) и из следующих тождеств (для любого языка L и любого семейства языков P_i, $i \in I$):

$$L\Big(\bigcup_{i\in I} P_i\Big) = \bigcup_{i\in I}(LP_i), \quad \Big(\bigcup_{i\in I} P_i\Big)L = \bigcup_{i\in I} P_i L. \qquad (13.1)$$

Объединение семейства множеств служит **точной верхней гранью** этого семейства (относительно теоретико-множественного включения).

Тождества гарантируют выполнение **непрерывности** операции **умножения** данного полукольца, т.е. непрерывности операции соединения.

Эти тождества доказываются точно так же, как тождества обычной дистрибутивности.

Докажем второе тождество из (13.1), используя **метод двух включений**.
Если $x \in \left(\bigcup_{i\in I} P_i\right) L$, то $x = yz$, где $y \in \bigcup_{i\in I} P_i$, а $z \in L$.
Согласно определению объединения семейства множеств, найдется такое $i \in I$, что $y \in P_i$, и тогда $yz = x \in P_i L$, т.е. $x \in \bigcup_{i\in I} P_i L$.
Обратное включение:
из $x \in \bigcup_{i\in I} P_i L$ следует, что для некоторого $i \in I$ $x \in P_i L$, т.е. $x = yz$, где $y \in P_i$, а $z \in L$, откуда $y \in \bigcup_{i\in I} P_i$, и, следовательно, $yz = x \in \left(\bigcup_{i\in I} P_i\right) L$. ►

Следствие 13.1. Для любого языка L верно тождество $L^+ = L^*L = LL^*$.

◀ Вычислим соединение LL^*: $LL^* = L\left(\bigcup\limits_{n=0}^{\infty} L^n\right)$.

Применим тождество $L\left(\bigcup_{i\in I} P_i\right) = \bigcup_{i\in I}(LP_i)$ (13.1).

Получим

$L\bigcup\limits_{n=0}^{\infty} L^n = \bigcup\limits_{n=0}^{\infty} LL^n = \bigcup\limits_{n=1}^{\infty} L^n$, т.е. $L^+ = LL^*$.

Тождество $L^+ = L^*L$ доказывается аналогично. ▶

В общем случае нельзя утверждать, что позитивная итерация языка L получается выбрасыванием из обычной итерации пустого слова. Это верно в том и только в том случае, когда язык L не содержит пустого слова. Если же $\lambda \in L$, то $L^+ = L^*$, так как тогда $L^0 = \{\lambda\} \subseteq L$.

13.2. Регулярные языки и регулярные выражения

Опишем с помощью индуктивной порождающей процедуры множество регулярных языков в алфавите

$$V = \{a_1, \ldots, a_n\}:$$

1) пустой язык $\varnothing$, язык $\{\lambda\}$, состоящий из пустого слова, и однобуквенные языки $\{a_i\}$ для каждого $a_i \in V$ считаем регулярными языками в алфавите V;
2) если P и Q — регулярные языки в алфавите V, то **объединение** $P \cup Q$ и **соединение** PQ есть регулярные языки в алфавите V;
3) если P — регулярный язык в алфавите V, то его итерация P^* есть регулярный язык в алфавите V;
4) других нет.

В замкнутом полукольце $\mathcal{L}(V)$ всех языков в алфавите V рассмотрим подалгебру, обозначаемую $\mathcal{R}(V)$, порожденную множеством регулярных языков.
Согласно порождающей процедуре, эта подалгебра **замкнута относительно итерации**.

Эта подалгебра есть полукольцо с итерацией. Оно играет важнейшую роль в теории формальных языков. Это полукольцо называют **полукольцом регулярных языков**.

Теорема 2. *Язык в алфавите V регулярен тогда и только тогда, когда он является элементом полукольца $\mathcal{R}(V)$. #*

Алгебраические операции над регулярными языками удобно представлять с помощью так называемых **регулярных выражений**.

Каждое регулярное выражение задает некоторый однозначно определяемый регулярный язык.

Язык	регулярное выражение
$\varnothing$	$\varnothing$
$\{\lambda\}$	λ
$\{a\}$, где $a \in V$	a
P	p
Q	q
$P \cup Q$	$p + q$
$P \cdot Q$	$p \cdot q$
P^*	p^*

В регулярных выражениях для обозначения операции объединения языков используют знак „ $+$ “ (плюс), а для операции соединения — знак умножения „ $\cdot$ “, который как правило опускают.
Соглашение о приоритетах: самый высокий приоритет имеет операция итерации, затем — соединения и, наконец, — объединения.

Пример 13.2. Регулярное выражение $a^* + (bc)^*$ обозначает множество цепочек, состоящее из цепочек вида a^n, $n \geq 0$, и цепочек вида $(bc)^n$, $n \geq 0$, где a, b, $c \in V$.
Слова (цепочеки) языка, заданного регулярным выражением $a^* + (bc)^*$: $a, aa, \ldots, a^n \ldots$ или $bc, bcbc, \ldots, (bc)^n, \ldots$, где $n \geq 0$.
Вместо регулярного выражения мы должны были бы использовать более громоздкую формулу: $\{a\}^* \cup (\{b\} \cdot \{c\})^*$.

Соответствие между регулярными языками и регулярными выражениями не является взаимно однозначным: один и тот же регулярный язык может представляться многими регулярными выражениями.
Например,

$$(a + b)^* = (a^*b^*)^*.$$

Для регулярного выражения $\alpha\alpha^*$ или $\alpha^*\alpha$ будем использовать обозначение α^+ и называть это выражение **позитивной итерацией** выражения α .
Из определения регулярного языка, теоремы 2 и следствия 13.1 следует, что **позитивная итерация** регулярного языка регулярна.

Конечные автоматы

Индуктивная процедура построения регулярных языков можно рассматривать как порождающую модель для регулярных языков.

Распознающей моделью для регулярных языков служат конечные автоматы.

Неформально конечный автомат можно описать как устройство, состоящее из **блока управления**,**входной ленты** и **головки автомата**

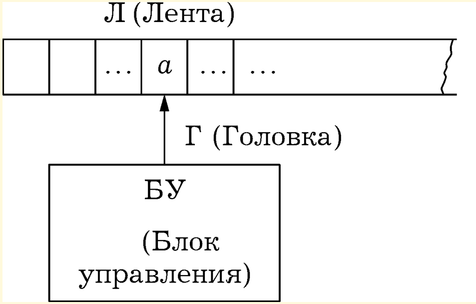

Рис. 1

Блок управления автомата может в каждый момент времени находиться в одном из конечного множества **состояний** Q. Головка автомата может быть установлена в точности на одну ячейку входной ленты.
Входная лента — это неограниченная справа полубесконечная лента, разделенная на ячейки. На ней записываются цепочки во **входном алфавите** V так, что буквы цепочки занимают ячейки ленты. Буквы записываются последовательно, начиная с первой без пропусков, по одной букве в каждой ячейке.
Цепочку, записанную на входной ленте автомата, называют **входной цепочкой**.

Автомат, читая входную цепочку, работает по шагам (или по тактам).
Пусть в некоторый момент времени автомат читает содержимое текущей ячейки ленты, блок управления находится в некотором состоянии $q \in Q$.
Такт работы автомата состоит в том, что в зависимости от содержимого читаемой ячейки, состояния q, а также содержимого внутренней памяти автомат сдвигает головку вправо на одну ячейку либо оставляет ее на прежнем месте, его блок управления переходит в некоторое новое состояние r (это состояние может совпасть с исходным состоянием q).

Пусть на входной ленте автомата записана некоторая цепочка $x \in V^*$.

Среди состояний блока управления выделено некоторое специальное состояние, называемое **начальным**, и некоторое подмножество состояний, которые называются **заключительными**.

В начальный момент времени блок управления находится в начальном состоянии, головка читает содержимое первой (крайней левой) ячейки ленты, в которой записан первый символ входной цепочки x.

Автомат читает цепочку x и делает один такт за другим до тех, пор пока он не прочитает последнюю букву цепочки.

После чтения последней буквы цепочки блок управления окажется в некотором состоянии q'. Если это состояние является заключительным, то тогда говорят, что автомат допустил цепочку x.

Из каждого состояния автомат может переходить в другое состояние, читая символ входной цепочки.

Автомат может переходить из одного состояния в другое по пустому слову, не читая ленту и не продвигая головку, если такие переходы предусмотрены описанием автомата. Такой такт можно рассматривать как переход из состояния в состояние по **пустой цепочке**. Его называют λ-тактом.

Принимается, что переход по пустой цепочке и переход по входному символу исключают друг друга.

Поведение конечного автомата определяется его системой команд, в которой каждая команда задается записью

$$qa \to r, \tag{13.2}$$

(„из состояния q по символу $a \in V$ можно перейти в состояние r “),
или

$$q\lambda \to r, \tag{13.3}$$

(„из состояния q по пустой цепочке можно перейти в состояние r “).
Возможно, что $q = r$.
Если для любой пары $(q,\, r)$ состояний конечного автомата существует команда (13.3), то для той же пары состояний нет ни одной команды (13.2) при $a \in V$ и наоборот.

Конечный автомат допускает интерпретацию в терминах **размеченных ориентированных графов**.

Будем рассматривать **состояния блока** управления конечного автомата как **вершины** ориентированного графа, множество **дуг** которого определяется системой команд следующим образом: дуга ведет из состояния q в состояние r (для данных состояний q и r) тогда и только тогда, когда в системе команд автомата есть команда (13.2) или команда (13.3), т.е. возможен переход из состояния q в состояние r .

Метка дуги (q, r) есть пустая цепочка λ, если из q в r можно перейти по пустой цепочке; в противном случае метка дуги (q, r) есть множество всех входных символов, по которым возможет переход из состояния q в состояние r .

Пример 13.3. Конечный автомат: входной алфавит $\{a, b, c\}$, множество состояний $\{q_0, q_1, q_2, q_3\}$, система команд:

$$q_0\lambda \to q_1,\ \ q_0\lambda \to q_3,\ \ q_1a \to q_2,$$
$$q_1b \to q_2,\ \ q_1a \to q_3,\ \ q_2b \to q_1,$$
$$q_3b \to q_2,\ \ q_3c \to q_2,\ \ q_3c \to q_3.$$

По этой системе команд построим размеченный ориентированный граф.

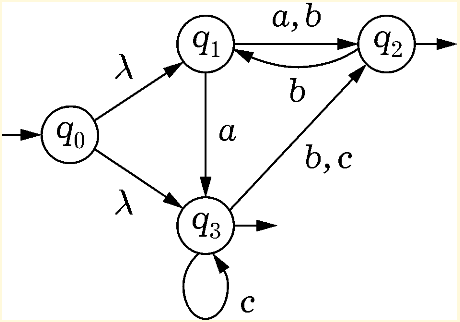

Рис. 2

Среди состояний автомата выделены начальное состояние q_0, помеченное стрелкой $\rightarrow \bigcirc$, и два заключительных состояния q_2 и q_3, также помеченные стрелкой $\bigcirc \rightarrow$.

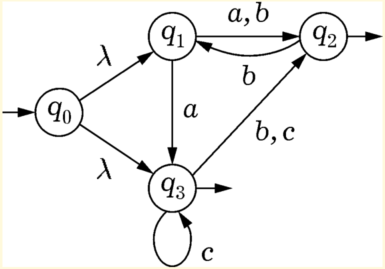

Рис. 3

Последовательности $abac$ отвечает **путь** в ориентированном графе, ведущий из вершины q_0 в вершину q_3:

$$q_0 \longrightarrow_\lambda q_1 \longrightarrow_a q_2 \longrightarrow_b q_1 \longrightarrow_a q_3 \longrightarrow_c q_3$$

Обозначение пустой цепочки λ, в виде метки дуги ориентированного графа, который представляет конечный автомат, можно интерпретировать как **регулярное выражение**, т.е. как регулярный язык, состоящий из одной пустой цепочки. Метка дуги — это множество букв $\{b_1, \ldots, b_m\} \subseteq V$. Метка дуги может быть также записана в виде регулярного выражения, как сумма $b_1 + \ldots + b_m$.
Метку каждой дуги можно считать регулярным выражением определенного вида или регулярным языком.

Это позволяет формально определить конечный автомат как **ориентированный граф, размеченный над полукольцом** $\mathcal{R}(V)$ регулярных языков.
Такое математическое определение конечного автомата дает возможность применить при изучении конечных автоматов алгебраические методы анализа размеченных ориентированных графов.

Математическое определение конечного автомата.

Определение 13.5. **Конечный автомат** — это ориентированный граф, размеченный над полукольцом $\mathcal{R}(V)$ регулярных языков в алфавите V с выделенной вершиной q_0, которая называется **начальной**, и с выделенным подмножеством вершин F, каждый элемент которого называется **заключительной вершиной**.

На **функцию разметки** при этом накладываются следующие ограничения: **метка** каждой **дуги** есть либо язык $\{\lambda\}$, либо непустое подмножество **алфавита** V.

Вершины графа называют обычно в этом случае **состояниями конечного автомата**, начальную вершину — **начальным состоянием**, а заключительную вершину — **заключительным состоянием конечного автомата**.

Пример 13.4.

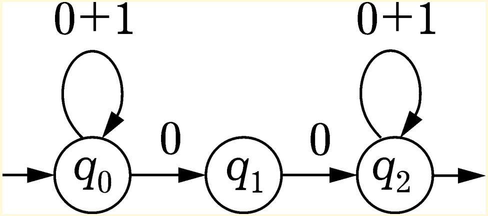

Рис. 4

На рисунке изображен конечный автомат, для которого алфавит $V = \{0, 1\}$.
Начальное состояние показано входной стрелкой, заключительное — выходной.
Метки дуг обычно пишут без фигурных скобок. Разрешена запись меток дуг и в виде *регулярных выражений*.

Конечный автомат может быть задан как пятерка:

$$M = (\mathcal{Q}, E, \varphi, q_0, F),$$

где

$\mathcal{Q}$ — множество состояний автомата;

E — множество дуг;

φ — функция разметки (весовая функция), причем для каждой дуги $e \in E$ ее метка $\varphi(e) = \{\lambda\}$ или $\varphi(e) \subseteq V$, при этом подмножество $\varphi(e)$ не пусто; $q_0 \in Q$ — начальное состояние;

$F \subseteq Q$ — подмножество заключительных состояний.

Алфавит V называется **входным алфавитом автомата** M, а его буквы — **входными символами** (или **входными буквами**) данного **автомата**.

Если $e = (q, r)$ — **дуга** автомата M и ее метка $\varphi(e)$ есть регулярное выражение λ, то в этом случае будем говорить, что в автомате M возможен **переход из состояния** q **в состояние** r по пустой цепочке, и писать $q \to_\lambda r$.
Дугу с меткой λ будем называть λ**-переходом** (или **пустой дугой**).
Если же метка дуги e есть множество, содержащее входной символ a, то будем говорить, что в автомате M возможен переход из состояния q в состояние r по символу a, и писать $q \to_a r$.

Для конечного автомата удобно ввести в рассмотрение **функцию переходов**, определив ее как отображение

$$\delta\colon Q \times (V \cup \{\lambda\}) \to 2^Q,$$

такое, что

$$\delta(q, a) = \{r\colon q \to_a r\},$$

т.е. значение функции переходов на упорядоченной паре (состояние, входной символ или пустая цепочка) есть множество всех состояний, в которые из данного состояния возможен переход по данному входному символу или пустой цепочке. В частности, это может быть пустое множество.

Замечание 13.1. Хотя конечный автомат определен как ориентированный граф, размеченный над полукольцом регулярных языков, метка дуги задается не как произвольный регулярный язык, а как язык, являющийся подмножеством **букв** входного алфавита, или язык, состоящий из одной пустой цепочки. Это связано с приведенным ранее описанием автомата, как устройства.

Рассмотрим автомат из примера 13.4

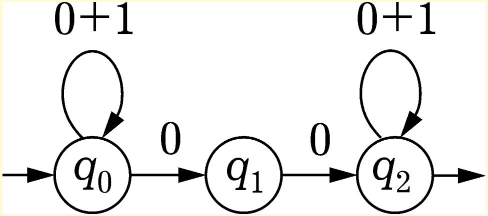

Рис. 5

Функции переходов конечного автомата:

$\delta(q_0, 0) = \{q_0,\ q_1\}$
$\delta(q_0, 1) = \{q_0\}$
$\delta(q_1, 0) = \{q_2\}$
$\delta(q_1, 1) = \varnothing$
$\delta(q_2, 0) = \{q_2\}$
$\delta(q_2, 1) = \{q_2\}.$

Система команд есть конечное множество **команд** вида

$qa \longrightarrow r$,

где q и r — состояния автомата;

a — входной символ или пустая цепочка, причем указанная команда тогда и только тогда содержится в системе команд, когда $q \longrightarrow_a r$.

Стрелка $(\longrightarrow)$, есть „метасимвол“, он не содержится алфавите V.

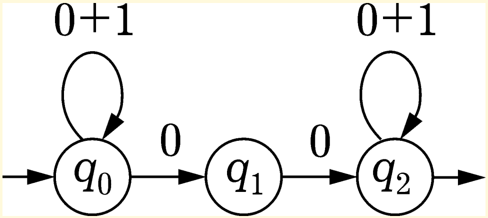

Рис. 6

Система команд автомата из примера 13.4:

$$
\begin{aligned}
q_0 0 &\longrightarrow q_0,\\
q_0 0 &\longrightarrow q_1,\\
q_0 1 &\longrightarrow q_0,\\
q_1 0 &\longrightarrow q_2,\\
q_2 0 &\longrightarrow q_2,\\
q_2 1 &\longrightarrow q_2.
\end{aligned}
$$

Используя функцию переходов, конечный автомат можно задать как упорядоченную пятерку:
$M = (V, Q, q_0, F, \delta)$,
где V — входной алфавит;
Q — множество состояний;
q_0 — начальное состояние;
F — множество заключительных состояний;
δ — функция переходов, заданная в виде системы команд.
Согласно определению **метки пути** в размеченном ориентированном графе, **метка пути** в конечном автомате есть **умножение** меток входящих в этот путь дуг (в порядке их прохождения). **Умножением полукольца** $\mathcal{R}(V)$ является **соединение языков**.
Таким образом, метка любого **пути конечной длины** в конечном автомате есть регулярный язык.

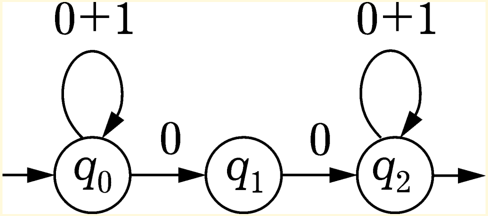

Рис. 7

Метка пути q_0, q_0, q_1, q_2 равна соединению языков $\{0,\ 1\} \cdot \{0\} \cdot \{0\} = \{000,\ 100\}$.

Это можно записать в виде регулярного выражения $(0+1)00$.

Метка пути q_0, q_1, q_2, q_2, q_2 может быть задана таким регулярным выражением:

$$00(0+1)(0+1) = 00(00+01+10+11) =$$
$$= 0000 + 0001 + 0010 + 0011,$$

Эта метка есть множество цепочек $\{0000,\ 0001,\ 0010,\ 0011\}$.

Если цепочка x принадлежит метке некоторого пути W — пути, ведущего из вершины q в вершину r конечного автомата M, то говорят, что цепочка x читается на пути W в M.

Пишем $q \Rightarrow_x^* r$, если x читается на некотором пути из q в r.

В том случае, когда явно надо указать **длину** n **пути**, на котором читается цепочка x, записываем $q \Rightarrow_x^n r$.

Если нужно подчеркнуть, что цепочка x читается на некотором пути ненулевой длины из q в r, то используем запись $q \Rightarrow_x^+ r$.

Стоимость прохождения из состояния q в состояние r есть (согласно общему определению этого понятия в размеченных ориентированных графах) **объединение** меток всех путей ведущих из q в r, т.е. множество всех таких x, что $q \Rightarrow_x^* r$. Это значит, что элемент c_{qr} **матрицы стоимостей** есть язык

$$c_{qr} = \{x \colon q \Rightarrow_x^* r\} \,.$$

Здесь объединение понимается как **бесконечная сумма** замкнутого полукольца $\mathcal{L}(V)$ всех языков в алфавите V, т.е. как **точная верхняя грань** последовательности языков относительно теоретико-множественного включения.

Определение 13.6. **Язык** $L(M)$ **конечного автомата** M есть множество всех цепочек во входном алфавите, читаемых в M на некотором пути из начального состояния в одно из заключительных состояний. Другими словами,

$$L(M) = \{x \colon q_0 \Rightarrow_x^* q_f,\ q_f \in F\} = \bigcup_{q_f \in F} c_{q_0 q_f}, \qquad (13.4)$$

где F — множество заключительных состояний.

Т. о. язык конечного автомата есть объединение тех элементов матрицы стоимостей автомата, которые находятся в строке, соответствующей начальному состоянию q_0, и в столбцах, соответствующих всем заключительным состояниям $q_f \in F$.

Факт, что стоимость прохождения между заданной парой вершин является регулярным языком, требует доказательства.

В конечном автомате метка произвольного пути конечной длины есть регулярный язык, поскольку он вычисляется как соединение конечного семейства регулярных языков.
Однако множество путей, ведущих из одной вершины в другую, может быть бесконечным.
Пусть, например, некоторый конечный путь, ведущий из вершины p в вершину q, содержит **контур**.
Тогда по этому контуру можно пройти любое количество раз. Следовательно множество путей, ведущих из вершины p в q, будет бесконечным.

Стоимость прохождения из вершины p в вершину q будет равна бесконечному объединению меток путей, ведущих из вершины p в q, и необязательно будет регулярным языком. Это бесконечное объединение можно рассматривать как операцию вычисления точной верхней грани в замкнутом полукольце всех языков в данном алфавите.

О языке $L(M)$ говорят, что он допускается КА M. О любой цепочке, принадлежашей языку $L(M)$, говорят, что она допускается КА M. Такую цепочку называют также **допустимой цепочкой** данного **конечного автомата**.

Определение 13.7. Два конечных автомата M_1 и M_2 называют **эквивалентными**, если их языки совпадают:

$$L(M_1) = L(M_2).$$