Лекции по теории формальных языков

Лекция 1.
Введение.
Языки и операции над ними.
Детерминированные конечные автоматы

Александр Сергеевич Герасимов
http://gas-teach.narod.ru

Кафедра математических и информационных технологий
Санкт-Петербургского академического университета.
Весенний семестр 2010/11 учебного года

11 февраля 2011 г.

План

План

Мотивировка к изучению теории формальных языков

Компилятор — программа, которая переводит текст (например, программу) на исходном языке в некотором смысле эквивалентный текст на целевом языке.

Обработка исходного текста компилятором:

- стадия анализа (front end) — проверка исходного текста на корректность и его перевод в некоторое внутреннее представление:
  - лексический анализ,
  - синтаксический анализ,
  - семантический анализ,
  - генерация промежуточного кода

  (точное решение задач распознавания);
- стадия реализации (back end):
  - машинно-независимая оптимизация кода,
  - генерация кода,
  - машинно-зависимая оптимизация кода

  (приближённое решение задач оптимизации).

Литература

Основная литература

- Замятин А. П., Шур А. М. Языки, грамматики, распознаватели: Учебное пособие. Екатеринбург : Изд-во Урал. ун-та, 2007 (электронный вариант книги — на `http://elar.usu.ru`, поиск)

Дополнительная литература

- Ахо А., Лам М., Сети Р., Ульман Дж. Компиляторы: принципы, технологии и инструментарий. М.: ООО "И.Д. Вильямс", 2008
- Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. М.: Мир, 1978
- Мартыненко Б. К. Языки и трансляции: Учеб. пособие. СПб.: Издательство С.-Петербургского университета, 2004 (электронный вариант книги — на `http://www.math.spbu.ru/user/mbk`)

План

Алфавит. Цепочка

- *Алфавит* — конечное множество *символов* (или *букв*).
- *Цепочка* (*слово* или *строка*) в алфавите (над алфавитом) Σ — конечная последовательность записанных подряд символов алфавита Σ (в том числе последовательность, не содержащая ни одного символа).
- Цепочка, не содержащая ни одного символа, называется *пустой* и обозначается через ε.
- Пример: ε, a, ba, aaba — цепочки в алфавите $\{\mathtt{a},\ \mathtt{b}\}$.
- Цепочка, состоящая из $i = 0, 1, 2, \ldots$ символов a, обозначается через $\mathtt{a}^i$. Например, $\mathtt{a}^0 = \varepsilon$, $\mathtt{a}^1 = \mathtt{a}$, $\mathtt{a}^3 = \mathtt{aaa}$.
- Число символов (не обязательно различных) в цепочке w называется *длиной* этой цепочки и обозначается $|w|$.

Конкатенация. Вхождение

Пусть u, v, w — цепочки (в некотором алфавите).

- *Конкатенацией* цепочек u и v называется цепочка uv.
- Примеры: если $u = \mathtt{ab}$ и $v = \mathtt{cd}$, то $uv = \mathtt{abcd}$;
  для любой цепочки x имеем $\varepsilon x = x\varepsilon = x$.
- Цепочка u называется *префиксом* (или *началом*) цепочки uv.
- Цепочка v называется *суффиксом* (или *концом*) цепочки uv.
- Префикс (соответственно, суффикс) цепочки w называется *собственным*, если он отличен от ε и от w.
- Цепочка v *входит* в цепочку (называется *подцепочкой* цепочки) uvw.
- Пример: $\mathtt{ab}$ — собственный префикс и подцепочка цепочки $\mathtt{abb}$.

Определение (формального) языка

Пусть Σ — алфавит.

- Множество всех цепочек в алфавите Σ обозначается через Σ^*.
- Множество всех непустых цепочек в алфавите Σ обозначается через Σ^+.
- *Языком* (*формальным языком*) в алфавите (над алфавитом) Σ называется какое угодно подмножество множества Σ^*.

Пример языка: скобочный язык LB

- Алфавит языка LB: $\{[,\ ]\}$.
- Индуктивное определение языка LB:
  - $\varepsilon \in LB$;
  - если $u \in LB$ и $v \in LB$, то $[u] \in LB$ и $uv \in LB$.
- Примеры:
  $[],\ [][],\ [[]][] \in LB$,
  $[,\ ][,\ []] \notin LB$.

Пример языка: язык списков LL

- Алфавит языка LL: $\{\mathrm{a},\ ;,\ [,\ ]\}$.
- Индуктивное определение языка LL:
  - $\mathrm{a} \in LL$;
  - если $u \in LL$ и $v \in LL$, то $[u] \in LL$ и $u; v \in LL$.
- Примеры:
  $[\mathrm{a}; \mathrm{a}],\ \mathrm{a}; [\mathrm{a}],\ [[\mathrm{a}]]; [\mathrm{a}] \in LL$,
  $[],\ [\mathrm{a}]] \notin LL$.

Пример языка: язык описаний типов LD

- Алфавит языка LD: $\{\texttt{i},\ :,\ ;,\ \texttt{int},\ \texttt{real}\}$.
- Каждая цепочка языка LD имеет вид:
  - непустой список букв `i`, разделённых точкой с запятой,
  - затем двоеточие,
  - наконец, один из символов `int` или `real`.
- Примеры:
  $\texttt{i} : \texttt{real} \in LD$,
  $\texttt{i}; \texttt{i}; \texttt{i} : \texttt{int} \in LD$.

Пример языка: язык двоичных чисел LN

- Алфавит языка LN: $\{0,\ 1,\ .\}$.
- Каждая цепочка языка LN имеет вид u, $.v$ или $u.v$, где u и v — произвольные непустые цепочки в алфавите $\{0,\ 1\}$ (т. е. $u, v \in \{0,\ 1\}^+$).
- Примеры: $1,\ .0100,\ .01,\ 01.1,\ 1.1 \in LN$.
- Язык LN' получаем, если дополнительно потребуем, чтобы целая часть (u) числа начиналась с 1, дробная часть (v) заканчивалась на 1.
- Примеры:
  $1,\ .01,\ 1.1 \in LN'$,
  $.0100,\ 01.1 \notin LN'$.

Пример языка: язык арифметических выражений LA

- Алфавит языка LA: $\{\mathtt{x},\ +,\ *,\ (,\ )\}$.
- Индуктивное определение языка LA:
  - $\mathtt{x} \in LA$;
  - если $u \in LA$ и $v \in LA$, то $(u) \in LA$, $u + v \in LA$ и $u * v \in LA$.
- Примеры: $(\mathtt{x})$, $\mathtt{x} + \mathtt{x} * \mathtt{x}$, $(\mathtt{x} + \mathtt{x}) * \mathtt{x} \in LA$.

План

Теоретико-множественные операции над языками. Произведение языков

Пусть K и L — языки в алфавите Σ.

- Теоретико-множественные операции:
  - объединение $K \cup L$,
  - пересечение $K \cap L$,
  - разность $K \setminus L$,
  - дополнение $\overline{K} = \Sigma^* \setminus K$.
- *Произведение* (или *конкатенация*) языков K и L:

$$KL = \{uv \in \Sigma^* \mid u \in K \text{ и } v \in L\}.$$

Произведение языков — ассоциативная операция.

- Пример: $K = \{\mathtt{ab}, \mathtt{abc}\}$, $L = \{\mathtt{ba}, \mathtt{cba}\}$,
  $KL = \{\mathtt{abba}, \mathtt{abcba}, \mathtt{abccba}\}$, $LK = \{\mathtt{baab}, \mathtt{baabc}, \mathtt{cbaab}, \mathtt{cbaabc}\}$.

Степень языка. Итерация языка

Пусть L — язык в алфавите Σ.

- *Степень* языка L: $L^0 = \{\varepsilon\}$, $L^{i+1} = L^i L$, $i = 0, 1, 2, \ldots$.
- *Итерация* (или *замыкание*) языка L: $L^* = \bigcup\limits_{i \geqslant 0} L^i$.
- *Позитивная итерация* языка L: $L^+ = \bigcup\limits_{i \geqslant 1} L^i$.
- Пример: $L = \{\mathrm{a}, \mathrm{ba}\}$, $L^* = \{\varepsilon, \mathrm{a}, \mathrm{ba}, \mathrm{a}^2, \mathrm{aba}, \mathrm{ba}^2, \mathrm{baba}, \mathrm{a}^3, \ldots\}$.
- Соглашение: приоритет итерации выше приоритета умножения, который выше приоритета всех теоретико-множественных операций.

Некоторые свойства операций над языками

Пусть K, L, M — языки в алфавите Σ.

(1) $K(L \cup M) = KL \cup KM$

(2) $K(L \cap M) \subseteq KL \cap KM$

(3) $L^* \cup M^* \subseteq (L \cup M)^*$

(4) $(L \cap M)^* \subseteq L^* \cap M^*$

Докажем, например, (2). Пусть $x \in K(L \cap M)$. Тогда $x = uv$ для некоторых $u \in K$ и $v \in L \cap M$. Имеем $v \in L$ и $v \in M$. Поэтому $uv \in KL$ и $uv \in KM$, так что $x = uv \in KL \cap KM$.

Но $K(L \cap M) \not\supseteq KL \cap KM$, например, для $K = \{\varepsilon, \mathtt{a}\}$, $L = \{\mathtt{ba}\}$, $M = \{\mathtt{aba}\}$: $K(L \cap M) = \emptyset$, $KL \cap KM = \{\mathtt{aba}\}$.

Подстановка

Пусть $\Sigma = \{a_1, \ldots, a_n\}$, Δ — алфавиты, $\tau : \Sigma \to 2^{\Delta^*}$ $(\tau(b) \subseteq \Delta^*)$.

- Сначала расширим отображение τ на Σ^*, положив $\tau(\varepsilon) = \varepsilon$ и $\tau(b_1 \ldots b_k) = \tau(b_1) \ldots \tau(b_k)$ для каждой непустой цепочки $b_1 \ldots b_k \in \Sigma^*$
- Затем расширим полученное отображение на 2^{Σ^*}, положив $\tau(L) = \bigcup\limits_{w \in L} \tau(w)$ для каждого $L \subseteq \Sigma^*$;
- Отображение τ называется *подстановкой* из алфавита Σ в алфавит Δ.
- Язык $\tau(L) \subseteq \Delta^*$ называется *результатом действия подстановки* τ на язык $L \subseteq \Sigma^*$.
- Пусть $\Sigma = \{a_1, a_2\}$, $L_1, L_2 \subseteq \Delta^*$, $\tau(a_1) = L_1$, $\tau(a_2) = L_2$. Тогда $\tau(\{a_1 a_2\}) = L_1 L_2$, $\tau(\{a_1, a_2\}) = L_1 \cup L_2$, $\tau(\{a_1\}^*) = L_1^*$.

Гомоморфизм

- Пусть Σ и Δ — алфавиты. Отображение $\phi : \Sigma^* \to \Delta^*$ называется *гомоморфизмом*, если $\phi(uv) = \phi(u)\phi(v)$ для любых $u, v \in \Sigma^*$.
- Язык $\phi(L) = \{\phi(w) \mid w \in L\}$ называется *гомоморфным образом* языка L (при гомоморфизме ϕ).
- Для задания гомоморфизма достаточно указать образы всех символов из Σ.
- Пример: если $L = \{\mathtt{aba}, \mathtt{aa}, \mathtt{bbb}\}$, ϕ — гомоморфизм, $\phi(\mathtt{a}) = \mathtt{ac}$, $\phi(\mathtt{b}) = \mathtt{ca}$, то $\phi(L) = \{\mathtt{accaac}, \mathtt{acac}, \mathtt{cacaca}\}$.
- Гомоморфизм можно рассматривать как частный случай подстановки: образ любого символа — язык из одного слова.

План

Распознаватель языка

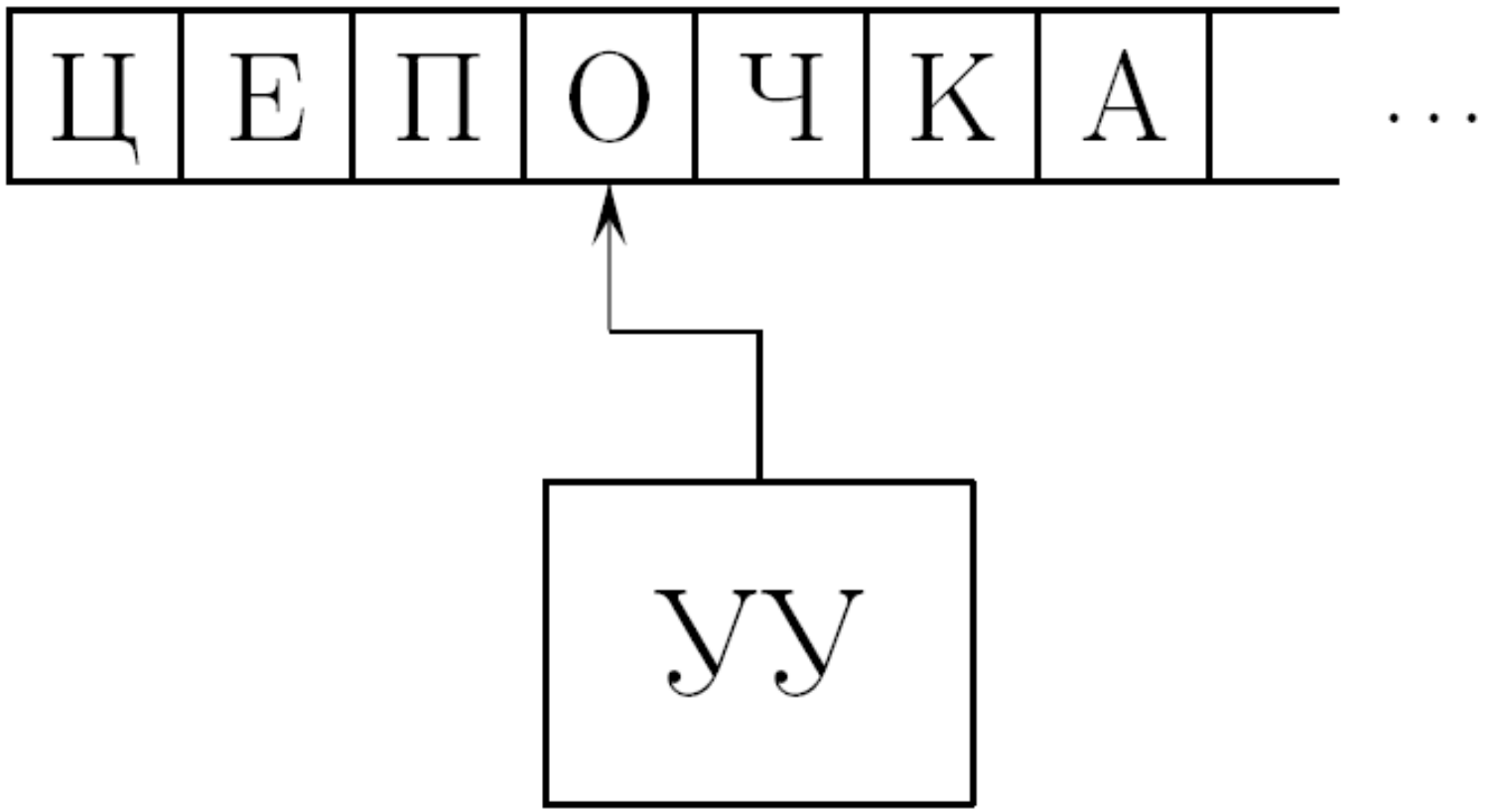
- *Распознавателем* языка L в алфавите Σ называется алгоритм, который по произвольной цепочке $w \in \Sigma^*$ определяет, принадлежит ли w языку L или нет.
- Говорят, что распознаватель языка L *распознаёт язык* L.
- Рассматриваемые далее различные виды конечных автоматов являются распознавателями языков некоторого класса.

Определение детерминированного конечного автомата

Детерминированным конечным автоматом (*ДКА*) называется пятёрка $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, где

- Q — непустое конечное множество *состояний*,
- Σ — алфавит,
- $\delta : Q \times \Sigma \to Q$ — *функция переходов*,
- $q_0 \in Q$ — *начальное* состояние,
- $F \subseteq Q$ — множество *заключительных* (или *допускающих*) состояний.

Представление ДКА в виде физического устройства



Переходы ДКА. Распознавание языка автоматом

- Работа ДКА состоит в последовательном переходе из текущего состояния q при текущем входном символе a в состояние $q' = \delta(q, a)$ и сдвиге к следующему входному символу.
- Переходом (автомата) будем называть тройку вида (q, a, q').
- Доопределим функцию переходов δ на $Q \times \Sigma^*$: $\delta(q, \varepsilon) = q$, $\delta(q, ua) = \delta(\delta(q, u), a)$.
- ДКА $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ *распознает* (или *допускает*) цепочку $w \in \Sigma^*$, если $\delta(q_0, w) \in F$.
- Множество всех цепочек, допускаемых автоматом $\mathcal{A}$, называется *языком, распознаваемым автоматом* $\mathcal{A}$, и обозначается $L(\mathcal{A})$.

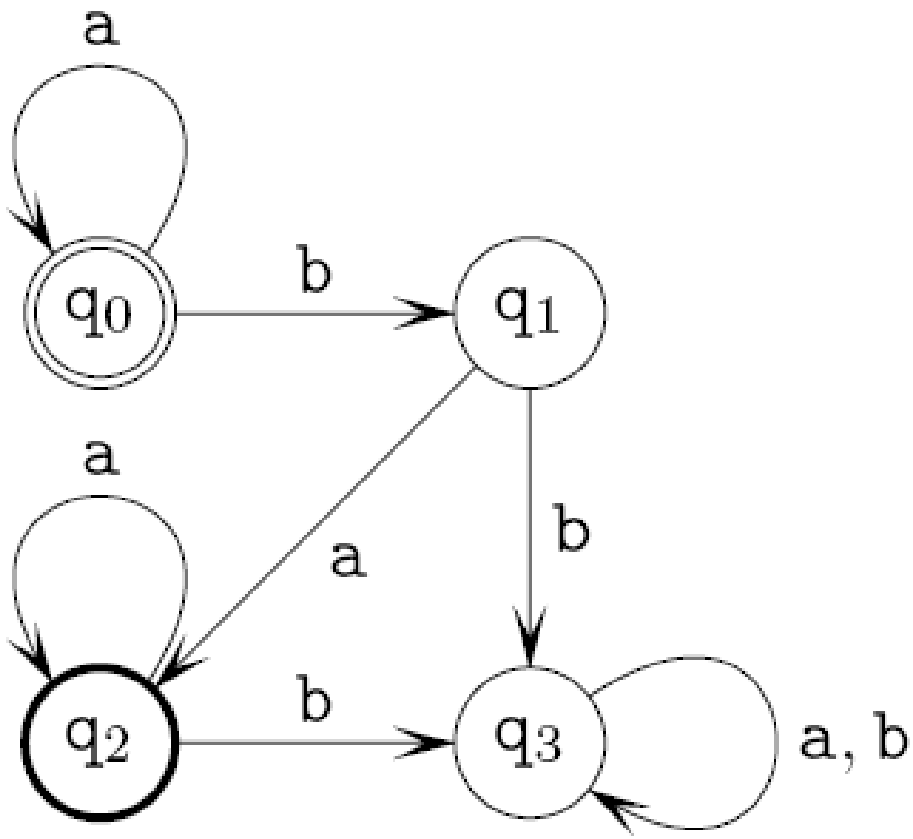
Задание ДКА расширенной таблицей переходов

ДКА $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$,
$Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{a, b\}$, $F = \{q_2\}$

	a	b	F
q_0	q_0	q_1	0
q_1	q_2	q_3	0
q_2	q_2	q_3	1
q_3	q_3	q_3	0

Задание ДКА диаграммой переходов

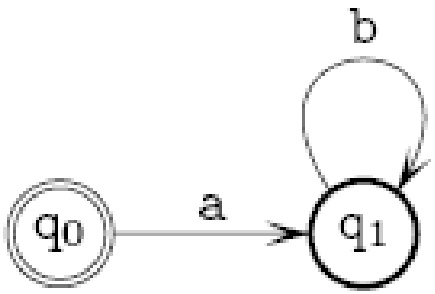
ДКА $\mathcal{A}$ тот же, что и на предыдущем слайде.

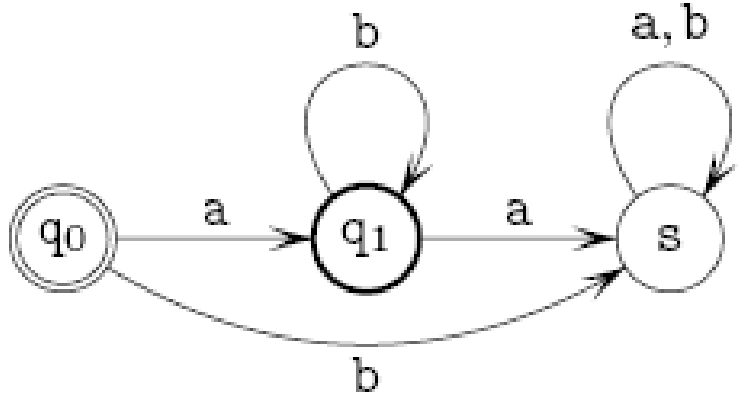


$$L(\mathcal{A}) = \{\mathtt{b}\mathtt{a}^n \mid n \geqslant 1\} \cup \{\mathtt{a}^m\mathtt{b}\mathtt{a}^n \mid m, n \geqslant 1\}$$

Неполные ДКА

- Мы определили ДКА со всюду определённой функцией переходов, такие ДКА также называют *полными*.
- Если в определении ДКА считать функцию переходов частичной, то получим определение *неполного ДКА*. (Остальные определения не изменяются.)
- Всякий неполный ДКА можно доопределить до полного, сохранив распознаваемый язык: введём новое незаключительное состояние s и значения всех неопределённых переходов положим равными s.





План

Примеры ДКА, распознающих один и тот же язык

	a	b	F
q_0	q_0	q_1	0
q_1	q_2	q_3	0
q_2	q_2	q_3	1
q_3	q_3	q_3	0

	a	b	F
q_0	q_0	q_1	0
q_1	q_2	q_3	0
q_2	q_2	q_3	1
q_3	q_3	q_3	0
q_4	q_4	q_4	0

	a	b	F
q_0	q_0	q_1	0
q_1	q_2	q_3	0
q_2	q_2	q_4	1
q_3	q_3	q_4	0
q_4	q_3	q_4	0

Как для ДКА $\mathcal{A}$ найти ДКА $\mathcal{B}$, минимальный по числу состояний и распознающий тот же язык, что и $\mathcal{A}$?

Достижимые и эквивалентные состояния. Приведённые, изоморфные и эквивалентные ДКА

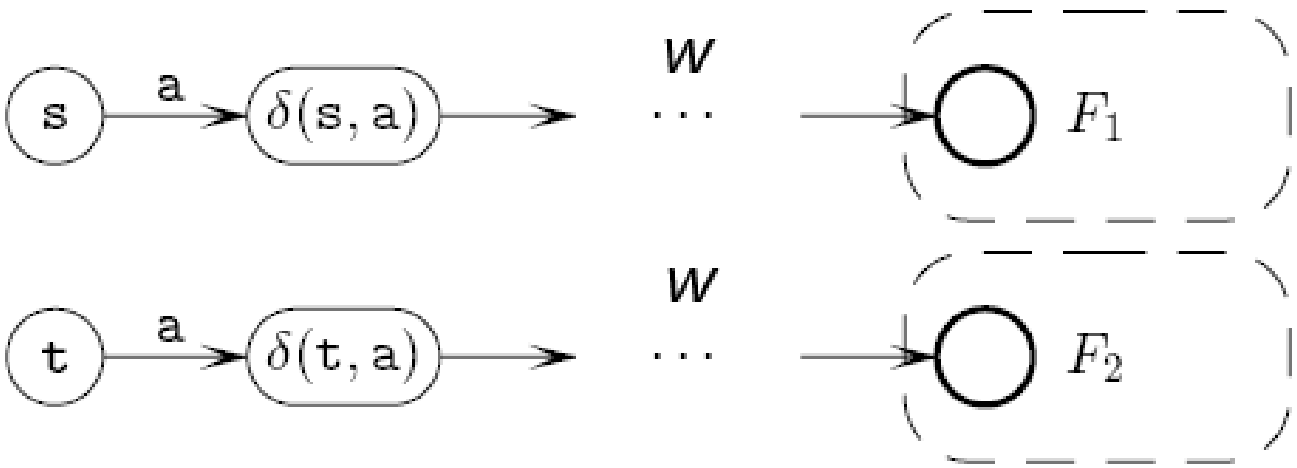
- Через $\mathcal{A}^{\mathtt{q}}$ будем обозначать автомат, полученный из ДКА $\mathcal{A}$ заменой начального состояния на состояние $\mathtt{q}$ автомата $\mathcal{A}$.
- Состояние $\mathtt{s}$ ДКА $\mathcal{A}$ (с начальным состоянием $\mathtt{q}_0$) называется *достижимым*, если существует цепочка w такая, что $\delta(\mathtt{q}_0, w) = \mathtt{s}$.
- Состояния $\mathtt{s}$ и $\mathtt{t}$ ДКА $\mathcal{A}$ называются *эквивалентными* (обозначение: $\mathtt{s} \sim \mathtt{t}$), если $L(\mathcal{A}^{\mathtt{s}}) = L(\mathcal{A}^{\mathtt{t}})$.
- ДКА называется *приведённым*, если он не имеет различных эквивалентных состояний и каждое его состояние достижимо.
- Говорят, что ДКА $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, \mathtt{q}_1, F_1)$ и $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, \mathtt{q}_2, F_2)$ *изоморфны*, если существует биекция $h : Q_1 \to Q_2$, сохраняющая переходы (т. е. $h(\delta_1(\mathtt{q}, \mathtt{a})) = \delta_2(h(\mathtt{q}), \mathtt{a})$ для любых $\mathtt{q} \in Q_1$, $\mathtt{a} \in \Sigma$).
- Два автомата называются *эквивалентными*, если они имеют общий алфавит и распознают один и тот же язык.
- Расширим понятие эквивалентности состояний: состояния $\mathtt{s}$ ДКА $\mathcal{A}_1$ и $\mathtt{t}$ ДКА $\mathcal{A}_2$ *эквивалентны* ($\mathtt{s} \sim \mathtt{t}$), если $L(\mathcal{A}_1^{\mathtt{s}}) = L(\mathcal{A}_2^{\mathtt{t}})$.

Лемма (о стабильности отношения $\sim$)

Пусть $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, \mathtt{q}_1, F_1)$ *и* $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, \mathtt{q}_2, F_2)$ *— ДКА,* $\mathtt{s} \in Q_1$, $\mathtt{t} \in Q_2$, $\mathtt{s} \sim \mathtt{t}$. *Тогда для любого* $\mathtt{a} \in \Sigma$ *имеет место* $\delta_1(\mathtt{s}, \mathtt{a}) \sim \delta_2(\mathtt{t}, \mathtt{a})$.

Доказательство.

- Пусть $w \in L(\mathcal{A}_1^{\delta_1(\mathtt{s},\mathtt{a})})$, т. е. $\delta_1(\delta_1(\mathtt{s}, \mathtt{a}), w) \in F_1$.
  $\delta_1(\delta_1(\mathtt{s}, \mathtt{a}), w) = \delta_1(\mathtt{s}, \mathtt{a}w)$, следовательно, $\mathtt{a}w \in L(\mathcal{A}_1^{\mathtt{s}})$.
- Поскольку $\mathtt{s} \sim \mathtt{t}$, то $\mathtt{a}w \in L(\mathcal{A}_2^{\mathtt{t}})$, т. е.
  $\delta_2(\mathtt{t}, \mathtt{a}w) = \delta_2(\delta_2(\mathtt{t}, \mathtt{a}), w) \in F_2$. Значит, $w \in L(\mathcal{A}_2^{\delta_2(\mathtt{t},\mathtt{a})})$.
- Аналогично получаем, что $w \in L(\mathcal{A}_2^{\delta_2(\mathtt{t},\mathtt{a})})$ влечёт $w \in L(\mathcal{A}_1^{\delta_1(\mathtt{s},\mathtt{a})})$.
  Таким образом, $L(\mathcal{A}_1^{\delta_1(\mathtt{s},\mathtt{a})}) = L(\mathcal{A}_2^{\delta_2(\mathtt{t},\mathtt{a})})$, т. е. $\delta_1(\mathtt{s}, \mathtt{a}) \sim \delta_2(\mathtt{t}, \mathtt{a})$.



Теорема (о приведённом ДКА)

Пусть $\mathcal{A}$ — произвольный ДКА. Среди автоматов, эквивалентных $\mathcal{A}$, существует единственный с точностью до изоморфизма приведённый ДКА $\mathcal{B}$. Число состояний автомата $\mathcal{B}$ не больше числа состояний любого другого автомата, эквивалентного $\mathcal{A}$.

Доказательство. *Существование приведённого ДКА, эквивалентного автомату $\mathcal{A}$.*

- Найдём (например, с помощью поиска в глубину) все достижимые состояния автомата $\mathcal{A}$ и удалим все недостижимые состояния. Получим ДКА, эквивалентный исходному. Далее считаем, что все состояния ДКА $\mathcal{A} = (Q, \Sigma, \delta, \mathrm{q}_0, F)$ достижимы.
- Пусть $K = \{K_1, \ldots, K_m\}$ — множество всех классов эквивалентности, определяемых отношением эквивалентности $\sim$ на множестве Q. Определим ДКА $R(\mathcal{A}) = (K, \Sigma, \phi, K_{\mathrm{q}_0}, F_K)$, где
  - $q_0 \in K_{\mathrm{q}_0}$ (отметим, что $K_{\mathrm{q}_0} = K_i$ для некоторого i),
  - F_K состоит из всех классов, содержащих только состояния из F,
  - $\phi(K_i, \mathrm{a}) = K_j \;\Leftrightarrow\; \exists \mathrm{q} \in K_i \; (\delta(\mathrm{q}, \mathrm{a}) \in K_j)$.

  (ϕ она не зависит от выбора $\mathrm{q} \in K_i$ по лемме о стабильности отношения $\sim$)

Теорема о приведённом ДКА: продолжение доказательства

- $\mathrm{s} \in F \Leftrightarrow \varepsilon \in \mathcal{A}^{\mathrm{s}}$. Поэтому каждый класс K_i содержит либо только заключительные, либо только незаключительные состояния автомата $\mathcal{A}$. В частности, все заключительные состояния $\mathcal{A}$ принадлежат классам из F_K.
- В силу стабильности отношения $\sim$, для $\mathrm{q} \in K_i$, $\mathrm{a} \in \Sigma$ верно: $\delta(\mathrm{q}, \mathrm{a}) \in K_j \Leftrightarrow \phi(K_i, \mathrm{a}) = K_j$.
- По индукции получаем, что для $\mathrm{q} \in K_i$, $w \in \Sigma^*$ верно: $\delta(\mathrm{q}, w) \in K_j \Leftrightarrow \phi(K_i, w) = K_j$.
- Следовательно, $\delta(\mathrm{q}_0, w) \in F \Leftrightarrow \phi(K_{q_0}, w) \in F_K$, т. е. $L(\mathcal{A}) = L(R(\mathcal{A}))$.

Теорема о приведённом ДКА: продолжение доказательства

- Проверим, что автомат $R(\mathcal{A})$ приведённый.
- Все состояния $R(\mathcal{A})$ достижимы:
  - в $\mathcal{A}$ все состояния достижимы;
  - в качестве цепочки, по которой состояние K_j достигается из K_{q_0}, можно взять любую цепочку, по которой некоторое состояние $s \in K_j$ достигается в $\mathcal{A}$.
- Возьмём произвольные различные состояния K_i и K_j автомата $R(\mathcal{A})$. Докажем, что они неэквивалентны.
- Пусть $q \in K_i$, $q' \in K_j$. Поскольку q и q' неэквивалентны, существует цепочка w такая, что ровно одно из состояний $\delta(q, w)$, $\delta(q', w)$ автомата $\mathcal{A}$ является заключительным.
- Тогда ровно одно из состояний $\phi(K_i, w)$, $\phi(K_j, w)$ автомата $R(\mathcal{A})$ заключительное, следовательно, K_i и K_j неэквивалентны.

Теорема о приведённом ДКА: продолжение доказательства

Единственность (с точностью до изоморфизма) приведённого автомата, эквивалентного данному.

- Пусть $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, \mathrm{q}_1, F_1)$ и $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, \mathrm{q}_2, F_2)$ — эквивалентные приведённые ДКА. Докажем, что они изоморфны. Имеем $\mathrm{q}_1 \sim \mathrm{q}_2$.
- Возьмем произвольное состояние $\mathrm{s} \in Q_1$. s достижимо, т. е. существует цепочка w такая, что $\delta_1(\mathrm{q}_1, w) = \mathrm{s}$. Положим $\mathrm{t} = \delta_2(\mathrm{q}_2, w)$. В силу стабильности отношения $\sim$ верно $\mathrm{s} \sim \mathrm{t}$.
- Каждое состояние s автомата $\mathcal{A}_1$ имеет эквивалентное состояние t в автомате $\mathcal{A}_2$, и притом единственное, так как в $\mathcal{A}_2$ нет различных эквивалентных состояний. Следовательно, существует биекция $h : Q_1 \to Q_2$, $h(\mathrm{s}) = \mathrm{t}$.
- Биекция h сохраняет переходы в силу стабильности отношения $\sim$ (подробное доказательство остаётся в качестве упражнения). Поэтому $\mathcal{A}_1$ и $\mathcal{A}_2$ изоморфны.

Теорема о приведённом ДКА: окончание доказательства

Последнее утверждение теоремы: число состояний приведённого автомата, эквивалентного $\mathcal{A}$, не больше числа состояний любого другого автомата, эквивалентного $\mathcal{A}$.

- Если автомат $\mathcal{A}$ не является приведённым, то эквивалентный ему приведённый автомат $R(\mathcal{A})$ по построению имеет меньшее число состояний, чем $\mathcal{A}$.
- Все приведённые автоматы, эквивалентные $\mathcal{A}$, изоморфны, следовательно, они имеют одинаковое число состояний.

□

Для получения приведённого автомата, эквивалентного данному, достаточно уметь эффективно строить отношение $\sim$. Вместо него мы научимся строить все классы эквивалентности.

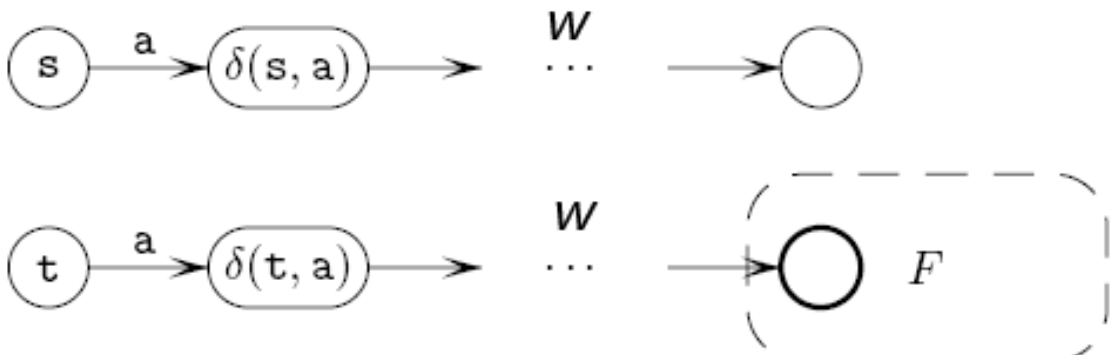
Цепочка, разделяющая состояния

- Будем говорить, что цепочка w *разделяет* состояния $\mathtt{s}$ и $\mathtt{t}$ ДКА $\mathcal{A}$, если w принадлежит ровно одному из языков $L(\mathcal{A}^{\mathtt{s}})$, $L(\mathcal{A}^{\mathtt{t}})$.
- Наименьшую длину цепочки, разделяющей состояния $\mathtt{s}$ и $\mathtt{t}$, обозначим через $sep(\mathtt{s}, \mathtt{t})$, положив $sep(\mathtt{s}, \mathtt{t}) = \infty$ при $\mathtt{s} \sim \mathtt{t}$.

Лемма

Пусть $\mathcal{A} = (Q, \Sigma, \delta, \mathtt{q}_0, F)$ *— ДКА,* $k = 0, 1, 2, \ldots$. *Тогда либо* $sep(\mathtt{s}, \mathtt{t}) = k$ *для некоторых* $\mathtt{s}, \mathtt{t} \in Q$, *либо* $sep(\mathtt{s}, \mathtt{t}) < k$ *для любых неэквивалентных* $\mathtt{s}, \mathtt{t} \in Q$.

Д о к а з а т е л ь с т в о. Цепочка $\mathtt{a}w$ разделяет состояния $\mathtt{s}$ и $\mathtt{t}$ тогда и только тогда, когда w разделяет состояния $\delta(\mathtt{s}, \mathtt{a})$ и $\delta(\mathtt{t}, \mathtt{a})$.



Алгоритм построения классов эквивалентных состояний ДКА

Вход. ДКА $\mathcal{A} = (Q, \Sigma, \delta, \mathrm{q}_0, F)$.
Выход. Множество $P = \{K_1, \ldots, K_m\}$ всех классов эквивалентности, определяемых отношением $\sim$ на множестве Q.
(Отметим, что $Q = K_1 \cup \ldots \cup K_m$ — разбиение.)

1. $P_0 := \{F,\ Q \setminus F\}$; $n := 0$;
2. $P_{n+1} := \bigcup\limits_{K \in P_n} \textsc{separate}(P_n, K)$;
3. **если** $(P_n \neq P_{n+1})$
4. $\quad n := n + 1$; **перейти на** 2
5. **иначе** $P := P_n$

Функция $\textsc{separate}(P, K)$ возвращает разбиение $\{K^{(1)}, \ldots, K^{(l)}\}$ класса (множества) $K \in P$ на подклассы в соответствии со следующим условием: два состояния s и t попадают в один подкласс $K^{(i)}$ тогда и только тогда, когда для любого $\mathrm{a} \in \Sigma$ состояния $\delta(\mathrm{s}, \mathrm{a})$ и $\delta(\mathrm{t}, \mathrm{a})$ находятся в некотором классе $K_{\mathrm{a}} \in P$.

Доказательство корректности алгоритма

- Пусть n — минимальное число такое, что состояния $\mathtt{s}$ и $\mathtt{t}$ принадлежат разным элементам множества P_n. Индукцией по n покажем, что $sep(\mathtt{s}, \mathtt{t}) = n$.
- База индукции ($n = 0$) очевидна: состояние из F и состояние из $Q \setminus F$ разделяются ε.
- Индукционный переход.
  - По выбору n состояния $\mathtt{s}$ и $\mathtt{t}$ принадлежат одному элементу множества P_{n-1}.
  - При $n > 1$ для любого $\mathtt{a} \in \Sigma$ состояния $\delta(\mathtt{s}, \mathtt{a})$ и $\delta(\mathtt{t}, \mathtt{a})$ находятся в некотором классе $K_{\mathtt{a}} \in P_{n-2}$.
  - Для некоторого $\mathtt{a} \in \Sigma$ состояния $\delta(\mathtt{s}, \mathtt{a})$ и $\delta(\mathtt{t}, \mathtt{a})$ принадлежат разным элементам множества P_{n-1} (при $n \geqslant 1$).
  - По индукционному предположению $sep(\delta(\mathtt{s}, \mathtt{a}),\ \delta(\mathtt{t}, \mathtt{a})) = n - 1$. Но тогда $sep(\mathtt{s}, \mathtt{t}) = n$.
- Итак, если алгоритм разделяет два состояния, то они неэквивалентны. Если алгоритм завершился, то $P_n = P_{n+1}$, и потому нет таких состояний, что $sep(\mathtt{s}, \mathtt{t}) = n + 1$. Тогда по лемме на слайде 38 все неэквивалентные состояния разделены.

Пример построения классов эквивалентных состояний ДКА

	a	b	F
0	2	4	0
1	7	4	0
2	3	6	0
3	3	1	0
4	7	1	0
5	2	7	1
6	7	3	0
7	7	5	1

- $P_0 = \{\{0,1,2,3,4,6\},\ \{5,7\}\}$
- $P_1 = \{\{0,2,3\},\ \{1,4,6\},\ \{5\},\ \{7\}\}$
- $P_2 = \{\{0,2,3\},\ \{1,4\},\ \{6\},\ \{5\},\ \{7\}\}$
- $P_3 = \{\{0,3\},\ \{2\},\ \{1,4\},\ \{6\},\ \{5\},\ \{7\}\}$
- $P_4 = \{\{0\},\ \{3\},\ \{2\},\ \{1,4\},\ \{6\},\ \{5\},\ \{7\}\} = P_5 = P$

КОНЕЦ ЛЕКЦИИ

Определения для практического занятия

Пусть Σ — алфавит, $L \subseteq \Sigma^*$, $\mathrm{a} \in \Sigma$.

- Правое деление L на a:
  $L\backslash\mathrm{a} = \{w \in \Sigma^* \mid w\mathrm{a} \in L\}$.
- Левое деление L на a:
  $\mathrm{a}\backslash L = \{w \in \Sigma^* \mid \mathrm{a}w \in L\}$.
- Префиксное замыкание языка L (множество всех префиксов цепочек языка L):
  $init(L) = \{w \in \Sigma^* \mid \exists x \in \Sigma^*(wx \in L)\}$.
- Множество всех цепочек языка L, никакой собственный префикс которых не принадлежит L:
  $min(L) = \{w \in L \mid \forall x, y \in \Sigma^+(w = xy \Rightarrow x \notin L)\}$.
- Пример: $L = \{\mathtt{aba},\ \mathtt{aaba},\ \mathtt{abab},\ \mathtt{aa},\ \mathtt{bbb}\}$, тогда
  - $L\backslash\mathrm{a} = \{\mathtt{ab},\ \mathtt{aab},\ \mathtt{a}\}$,
  - $\mathrm{a}\backslash L = \{\mathtt{ba},\ \mathtt{aba},\ \mathtt{bab},\ \mathtt{a}\}$,
  - $init(L) = \{\varepsilon,\ \mathtt{a},\ \mathtt{b},\ \mathtt{aa},\ \mathtt{ab},\ \mathtt{bb},\ \mathtt{aab},\ \mathtt{aba},\ \mathtt{bbb},\ \mathtt{aaba},\ \mathtt{abab}\}$,
  - $min(L) = \{\mathtt{aba},\ \mathtt{aa},\ \mathtt{bbb}\}$.