Федеральное агенство по образованию

Федеральное государственное образовательное учреждение
высшего профессионального образования
«Южный федеральный университет»

Факультет математики, механики и компьютерных наук

Практические занятия по курсу «Теория автоматов и формальных языков»

А. М. Пеленицын

Ростов-на-Дону
2010

Занятие 1: грамматики, регулярные выражения

1. Порождающие грамматики, иерархия Хомского.
Для каждой грамматики, встречающейся в заданиях, следует указать её тип (в иерархии Хомского). Написать грамматику, порождающую:

(1) язык Σ^*, где (a) $\Sigma = \{0, 1\}$; (b) Σ — произвольный (конечный) алфавит;

(2) произвольный конечный язык $L = \{\omega_i\}_{i=1}^n$;

(3) $\{a^+b^+\}, \{a^nb^n \mid n \in \mathbb{N}\}, \{a^nb^na^m \mid m, n \in \mathbb{N}\}$; $\{a^nb^nc^n \mid n \in \mathbb{N}\}$ — $\otimes$[1];

(4) множество правильных скобочных последовательностей («язык Дика») с одним типом скобок;
множество правильных скобочных последовательностей («язык Дика») с двумя типами скобок — $\otimes$;

(5) арифметическую прогрессию $\{a + nd \mid n \in \mathbb{N}_0\}$, $d > 0$, $0 \leqslant a < d$ (имея в виду изоморфизм моноидов $(\mathbb{N}_0, +) \cong (\{|\}^*, \cdot)$, где $\cdot$ означает операцию конкатенации); язык, являющийся объединением конечного числа арифметических прогрессий.

2. Регулярные выражения.
Написать регулярное выражение для

(1) языка над $\{a, b, c\}$ из всех слов, содержащих хотя бы один символ a;

(2) языка над $\{a, b, c\}$ из всех слов, содержащих хотя бы один символ a и хотя бы один символ b;

(3) языка над $\{0, 1\}$ из всех слов, в которых третий с правого края символ равен 1;

(4) языка над $\{0, 1\}$ из всех слов, в которых нет двух подряд идущих единиц;

(5) языка над $\{0, 1\}$ из всех слов, в которых любая пара смежных нулей, расположена левее любой пары смежных единиц;

(6) языка над $\{0, 1\}$ из всех слов с чередующимися нулями и единицами.

[1] Задания, отмеченные $\otimes$, — для самостоятельного выполнения.

Занятие 2

1. Системы линейных уравнений с регулярными коэффициентами

Решить системы уравнений:

$$\begin{cases} X_1 = aX_1 + aX_2; \\ X_2 = bX_2 + b. \end{cases} \tag{1}$$

$$\begin{cases} X_1 = 1X_1 + 0X_2 + \varnothing X_3; \\ X_2 = 1X_1 + \varnothing X_2 + 0X_3; \\ X_3 = 0X_1 + \varnothing X_2 + 1X_3. \end{cases} \tag{2}$$

$$\begin{cases} X_1 = a^*X_1 + (a+b)^*X_2; \\ X_2 = (a+b^*)X_1 + aX_2 + b^*. \end{cases} \tag{3}$$

$$\begin{cases} X_1 = (a+b)X_1 + \varnothing X_2 + a^*X_3; \\ X_2 = \varnothing X_1 + aX_2 + a^*; \\ X_3 = b^*X_1 + \varnothing X_2 + a^*X_3. \end{cases} \tag{4}$$

Написать регулярные выражения для языков, заданных грамматиками со следующими продукциями:

$$\begin{array}{l} S \to 1A \mid 2S; \\ A \to 0B \mid 0S \mid 1A; \\ B \to 1C \mid 2C; \\ C \to \varepsilon \mid 1S \mid 2A. \end{array} \tag{5}$$

$$\begin{array}{l} S \to 0A \mid 1S \mid \varepsilon; \\ A \to 0B \mid 1A; \\ B \to 0S \mid 1B. \end{array} \tag{6}$$

2. Доказательство нерегулярности формальных языков

Теорема 1 («Лемма о накачке» / «Лемма о разрастании» / «Pumping Lemma», И. Бар-Хиллел — М. Пелис — Э. Шамир, 1961)**.** *Пусть L — регулярный язык. Тогда существует такая константа $n \in \mathbb{N}$, что для любого слова $w \in L$, такого что $|w| \geqslant n$, существует такое разбиение $w = xyz$ слова w, что:*

(1) $y \neq \varepsilon$;

(2) $|xy| \leqslant n$;

(3) $\{xy^kz \mid k \geqslant 0\} \subset L$.

Доказательство. $\otimes$ — найти в одной из предложенных электронных книг. ■

Доказать, что следующие языки нерегулярны:

(1) $\{0^n 1^n \mid n \in \mathbb{N}\}$;

(2) язык из всех слов $w \in \{0,1\}^*$, содержащих одинаковое количество 0 и 1;

(3) $\{ww \mid w \in \{0,1\}^*\}$;

(4) $\{0^n 1^m \mid n \leqslant m\}$;

(5) $\{1^p \mid p$ — простое$\}$;

(6) $\{0^n 1 0^n \mid n \in \mathbb{N}\}$;

(7) $\{1^{n^2} | n \in \mathbb{N}\}$;

(8) $\{1^{n!} | n \in \mathbb{N}\}$.

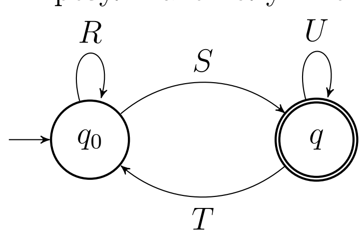
Занятие 3: Нахождение языка конечного автомата

Задача: описать язык заданного конечного автомата регулярным выражением.
Способ 1.

$$
\begin{aligned}
&(\text{Объект реального мира} \overset{0}{\mapsto})\\
&\quad \text{конечный автомат} \overset{1}{\mapsto}\\
&\quad \text{граф переходов конечного автомата} \overset{2}{\mapsto}\\
&\quad \text{ПЛ-грамматика} \overset{3}{\mapsto}\\
&\quad \text{система линейных уравнений} \overset{4}{\mapsto}\\
&\quad \text{решение системы.}
\end{aligned}
$$

Способ 2. Исключение состояний. Метод исключения состояний подразумевает последовательное удаление вершин графа переходов автомата, которое протоколируется с помощью записи на оставшихся дугах регулярных выражений (можно считать, что в изначальном графе на дугах простейшие регулярные выражения — однобуквенные).
Процедура исключения состояния s: для каждых двух (необязательно различных, но несовпадающих с s) состояний p и q, таких что существует фрагмент графа переходов автомата $p \xrightarrow{R_1} s \xrightarrow{R_2} q$, где R_1, R_2 — некоторые регулярные выражения (метки дуг переходов), прибавить к метке дуги $p \xrightarrow{R_3} q$ выражение $R_1 R^* R_2$, где R это метка петли на вершине s (если петля на s и/или дуга $p \to q$ отсутствовали в исходном графе, то можно считать, что их метки равны $\varnothing$) — таким образом получена дуга с меткой: $p \xrightarrow{R_3 + R_1 R^* R_2} q$. Удалить все просмотренные дуги $p \to s$ и $s \to q$, инцидентные вершине s. После этого s стала изолированной либо имеются только вхоящие или только исходящие из неё дуги. Вершину s можно удалить (с входящими или выходящими из неё дугами, если таковые имеются).
Алгоритм нахождения языка автомата методом исключения состояний.

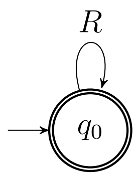
1. Для каждого финального состояния $q \in F$, отличного от начального q_0, применять процедуру исключения состояний до тех пор, пока не останутся две вершины: q_0 и q. В результате получится подобный автомат:

   Допускаемый им язык описывается так:

$$(R + SU^*T)^* SU^*.$$

2. Если начальное состояние q_0 является финальным ($q_0 \in F$), применять процедуру исключения состояний, пока не останется единственная вершина q_0. В результате получится подобный автомат:



Допускаемый им язык описывается так: R^*.

3. Язык исходного автомата определяется как сумма всех регулярных выражений, полученных на шагах (1)–(2).

Решите поставленную задачу каждым из двух способов для конечного автомата:

(1) моделирующего лампочку;

(2) моделирующего лампочку, которая сгорает на третьем включении, считая финальными состояния, когда лампочка выключена, но ещё не сгорела;

(3)

	0	1
$\to q_0$	q_1	q_0
q_1	q_2	q_0
$\boxed{q_2}$	q_2	q_1

(4)

	0	1
$\to q_0$	q_1	q_2
q_1	q_0	q_2
$\boxed{q_2}$	q_1	q_0

(5)

	0	1
$\to \boxed{p}$	s	p
q	p	s
r	r	q
s	q	r

Занятие 4

Теорема 2. *Класс регулярных языков замкнут относительно операций: объединения, конкатенации, итерации, дополнения, пересечения, разности.*

Доказательство. Для регулярных языков L_1, L_2 рассмотрим описывающие их регулярные выражения $\mathrm{RE}(L_1)$, $\mathrm{RE}(L_2)$. Язык объединения (соответственно, конкатенации, итерации) описывается выражением $\mathrm{RE}(L_1)+\mathrm{RE}(L_2)$ (соответственно, $\mathrm{RE}(L_1)\,\mathrm{RE}(L_2)$, $\mathrm{RE}(L_1)^*$), а значит, регулярен.
Для регулярного языка L построим распознающий его детерминированный[2] конечный автомат $\mathcal{A}(L) = (Q, \Sigma, \delta, q_0, F)$. Легко проверить, что автомат $\widetilde{\mathcal{A}}(L) = (Q, \Sigma, \delta, q_0, Q \setminus F)$ допускает дополнение $\overline{L} = \Sigma^* \setminus L$ языка L ⊗, которое является, таким образом, регулярным языком.
Поскольку справедливо $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$, язык $L_1 \cap L_2$ регулярен по доказанному выше[3]. Аналогичное можно заключить из равенства $L_1 \setminus L_2 = L_1 \cap \overline{L_2}$. ■

Доказать (не)регулярность:

(1) $\{0^n 1^m \mid n \neq m\}$;

(2) $\{w \in \{0,1\}^* \mid w$ содержит одинаковое число 0 и 1$\}$;

(3) языка из слов $w \in \{0, \ldots, 9\}^*$, которые являются десятичной записью чисел, делящихся на 2 или на 3, без лишних лидирующих нулей;

(4) $\{0^n 1^m 2^{n-m} \mid n \geqslant m\}$;

(5) $\{a^n b a^m b a^{n+m} \mid n, m \in \mathbb{N}\}$.

Контрпример к достаточности леммы о накачке

Для языка

$$L = \{a^i b^j c^k \mid i, j, k \in \mathbb{N}_0 \wedge (i = 1 \Rightarrow j = k)\}$$

покажите, что

(1) L нерегулярен,

(2) L удовлетворяет условию леммы о накачке.

[2] Сделанное дальше утверждение, однако, неверно для недетерминированного конечного автомата — постройте соответствующий контрпример. ⊗

[3] Имеется более полезная конструкция, которая строит ДКА, допускающий $L_1 \cap L_2$, внутри которого «параллельно» работают $\mathcal{A}(L_1)$ и $\mathcal{A}(L_2)$ — попытайтесь придумать её или разберите Теорему 4.8 раздела 4.2.1 книги Хопкрофта и др. *Введение в теорию автоматов...* ⊗

Занятие 5

1. Построение недетерминированных автоматов

Построить автомат, распознающий

(1) язык над $\{0, 1\}$ из слов, заканчивающихся на 01;

(2) язык, представляющий собой десятичную запись чисел, делящихся на 4;

(3) язык над $\{a, b\}$, заданный регулярным выражением $(ab + aba)^*$;

(4) язык над $\{0, 1, \ldots, 9\}$ из слов, в которых последняя цифра встречается ещё где-то в них;

(5) язык над $\{0, 1, \ldots, 9\}$ из слов, в которых последняя цифра больше нигде в них не встречается;

(6) язык над $\{0, 1\}$ из слов, в которых содержится два 0, разделённых символами, количество которых кратно 4 (нуль символов также считать кратными четырём).

2. Детерминизация конечных автоматов

Теорема 3 (о детерминизации конечных автоматов, М.О. Рабин — Д. Скотт, 1959)**.** *Пусть задан недетерминированный конечный автомат (НКА)*

$$\mathcal{A} = (Q, \Sigma, \delta, q_0 \in Q, F \subset Q).$$

Определим по нему детерминированный конечный автомат (ДКА), используя «конструкцию подмножеств» (subset construction или powerset construction)*:*

$$\widehat{\mathcal{A}} = (\widehat{Q} = 2^Q, \Sigma, \hat{\delta}, \{q_0\}, \widehat{F}),$$

где:

$$\begin{aligned} \widehat{F} =& \{\Omega \in \widehat{Q} \mid \Omega \cap F \neq \varnothing\}; \\ \hat{\delta}(\Omega, a) =& \bigcup_{q \in \Omega} \delta(q, a). \end{aligned}$$

Тогда

$$L(\mathcal{A}) = L(\widehat{\mathcal{A}}).$$

Доказательство. См. лекции или предложенную электронную литературу. ■

В конструкции подмножеств происходит экспоненциальный рост числа состояний автомата, который можно попытаться избежать, с помощью вычисления лишь достижимых состояний ДКА, используя

Алгоритм («ленивое вычисление» подмножеств).

База Состояние $\{q_0\}$ достижимо.

Индукция Если множество состояний S достижимо, тогда для каждого $a \in \Sigma$ достижимо $\hat{\delta}(S, a)$.

Провести детерминизацию следующих НКА:

(1) распознающего язык над $\{0, 1\}$ из слов, заканчивающихся на 01;

(2) распознающего язык, заданный регулярным выражением $(ab + aba)^*$;

(3)

	0	1
$\to p$	$\{p, q\}$	$\{p\}$
q	$\{r\}$	$\{r\}$
r	$\{s\}$	$\varnothing$
$\boxed{s}$	$\{s\}$	$\{s\}$

(4)

	0	1
$\to p$	$\{q, s\}$	$\{q\}$
$\boxed{q}$	$\{r\}$	$\{q, r\}$
r	$\{s\}$	$\{p\}$
$\boxed{s}$	$\varnothing$	$\{p\}$

(5)

	0	1
$\to p$	$\{p, q\}$	$\{p\}$
q	$\{r, s\}$	$\{t\}$
r	$\{p, r\}$	$\{t\}$
$\boxed{s}$	$\varnothing$	$\varnothing$
$\boxed{t}$	$\varnothing$	$\varnothing$

Занятие 6: нормальная форма Хомского

Определение. Пусть дана КС-грамматика $G = (\Sigma, N, \mathcal{P}, S \in N)$. Говорят, что символ $X \in \Sigma \cup N$

(1) *полезный в* G, если

$$\exists \alpha, \beta \in (\Sigma \cup N)^* \; \exists w \in \Sigma^* : \quad S \Rightarrow_G^* \alpha X \beta \Rightarrow_G^* w.$$

(2) *порождающий в* G, если

$$\exists w \in \Sigma^* : \quad X \Rightarrow_G^* w$$

(3) *достижимый в* G, если

$$\exists \alpha, \beta \in (\Sigma \cup N)^* : \quad S \Rightarrow_G^* \alpha X \beta$$

Бесполезным называется любой символ, не являющийся полезным.

Теорема 4. *Если к КС грамматике* G последовательно *применить два преобразования:*

(1) удалить символы, не являющиеся порождающими,

(2) удалить символы, не являющиеся достижимыми,

то будет получена грамматика, не содержащая бесполезных символов.

Замечание. Порядок действий в **теореме** существенен.

Задача. Удалить бесполезные символы в грамматиках с продукциями:

$$(1)\quad \begin{array}{l} S \to 0 \mid A, \\ A \to AB, \\ B \to 1; \end{array} \qquad (2)\quad \begin{array}{l} S \to AB \mid CA, \\ A \to a, \\ B \to BC \mid AB, \\ C \to aB \mid \varepsilon. \end{array}$$

Определение (Хомский, 1959)**.** Говорят, что КС-грамматика $G = (\Sigma, N, \mathcal{P}, S \in N)$ находится в *нормальной форме Хомского (НФХ)*, если она не содержит бесполезных символов и каждая продукция грамматики имеет один из двух видов:

1) $A \to a$,

2) $A \to BC$,

где $A, B, C \in N$, $a \in \Sigma$.

Схема приведения грамматики к НФХ:

(1) удалить ε-продукции;

(2) удалить цепные продукции (продукции вида $A \to B$);

(3) удалить бесполезные символы;

(4) привести грамматику к НФХ, используя метод «разбиения слов на слоги».

Замечание. Порядок действий в **схеме** существенен.

Задача. Привести к нормальной форме Хомского грамматики с продукциями:

(1) $S \to ASB \mid \varepsilon,$ $A \to aAS \mid a,$ $B \to SbS \mid A \mid bb;$	(2) $S \to 0A0 \mid 1B1 \mid BB,$ $A \to C,$ $B \to S \mid A,$ $C \to S \mid \varepsilon;$
(3) $S \to AAA \mid B,$ $A \to aA \mid B,$ $B \to \varepsilon;$	(4) $S \to aAa \mid bBb \mid \varepsilon,$ $A \to C \mid a,$ $B \to C \mid b,$ $C \to CDE \mid \varepsilon,$ $D \to A \mid B \mid ab.$

Занятие 7

1. Другое определение нормальной формы Хомского. Заметим, что грамматика в НФХ не может порождать ε, однако мы знаем КС-языки, содержащие ε. В действительности имеет место

Теорема 5. *Пусть G — КС-грамматика, а G' — грамматика в НФХ, полученная из G применением рассмотренного ранее алгоритма. Тогда*

$$L(G') = L(G) \setminus \{\varepsilon\}.$$

Можно сформулировать другое определение НФХ, которое с практической точки зрения не хуже нашего, но позволяет выводить ε.

Определение (другое определение нормальной формы Хомского)**.** Говорят, что КС-грамматика G находится в *нормальной форме Хомского*, если она не содержит бесполезных символов и каждая продукция грамматики имеет один из видов:

(1) $A \to a$,

(2) $A \to BC$,

(3) $S \to \varepsilon$,

где $a \in \Sigma$, $A, B, C, S \in N$, S — стартовый символ, *не встречающийся в правых частях продукций грамматики.*

Чтобы получить НФХ в смысле последнего определения достаточно добавить в алгоритм удаления ε-правил шаг 4:

> Если $S \in \mathrm{Gen}_G(\varepsilon)$, то ввести в грамматику новый стартовый символ S' и две продукции $S' \to S \mid \varepsilon$.

$\otimes$ Скорректировать решения заданий по получению НФХ так, чтобы ответом служила грамматика в НФХ в смысле второго определения.

2. Алгоритмические проблемы контекстно-свободных языков. Тремя основными проблемами теории формальных языков являются:

(1) *проблема пустоты*: для данной грамматики G определить

$$L(G) \stackrel{?}{=} \varnothing;$$

(2) *проблема принадлежности*: для данных грамматики G и слова $w \in \Sigma^*$ определить

$$w \stackrel{?}{\in} L(G);$$

(3) *проблема эквивалентности*: для данных грамматик G_1, G_2 определить

$$L(G_1) \stackrel{?}{=} L(G_2).$$

Проверка пустоты КС-языка сводится к построению $\mathrm{Gen}_G(\Sigma)$ и проверке $S \in \mathrm{Gen}_G(\Sigma)$. Рассмотрим один алгоритм, решающий проблему принадлежности для КС-языков.

Алгоритм (Кок—Янгер—Касами, «CYK-алгоритм»).
ВХОД: грамматика $G = (\Sigma, N, \mathcal{P}, S \in N)$ в НФХ, слово $w \in \Sigma^*$.
ВЫХОД: да, $w \in L(G)$ / нет, $w \notin L(G)$.
МЕТОД: последовательное определение нетерминалов, выводящих всевозможные подстроки w всё большей длины.
Пусть $w = w_1 \dots w_n$. Для всех $1 \leqslant i \leqslant j \leqslant n$ определим множество

$$N_{ij} = \{A \in N \mid A \Rightarrow_G^* w_i \dots w_j\}.$$

Очевидно, что $w \in L(G) \Leftrightarrow S \in N_{1n}$. Приведём алгоритм построения множеств N_{ij}.

for $i \leftarrow 1$ **to** n
  do $N_{ii} \leftarrow \{A \in N \mid A \to w_i \in \mathcal{P}\}$ $\triangleright$ Подстроки w длины 1
for $s \leftarrow 2$ **to** n $\triangleright$ Цикл по длине подстроки
  do for $i \leftarrow 1$ **to** $n - s + 1$ $\triangleright$ Цикл по месту начала подстроки
    $j \leftarrow i + s - 1$ $\triangleright$ Позиция конца подстроки с началом в w_i длины s
    $N_{ij} \leftarrow \{A \in N \mid A \to BC \in \mathcal{P};\ \exists k \in [i, j-1]_{\mathbb{Z}} \colon B \in N_{ik},\ C \in N_{k+1j}\}$

Замечание. Алгоритм удобно выполнять, заполняя таблицу с N_{ij} в ячейках.

Используя CYK-алгоритм,

(1) для грамматики G с продукциями:
$S \to AB, \quad A \to BB \mid a, \quad B \to AB \mid b$
определить, принадлежат ли $L(G)$ строки: (а) $aabbb$, (б) $babab$, (в) b^7;

(2) для грамматики G с продукциями:
$S \to AB \mid BC, \quad A \to BA \mid a, \quad B \to CC \mid b, \quad C \to AB \mid a$
определить, принадлежат ли $L(G)$ строки: (а) $ababa$, (б) $baaab$, (в) $aabab$.

Замечание 1 (о применении CYK-алгоритма к решению задачи синтаксического анализа)**.** Несложная модификация CYK-алгоритма позволяет в случае $w \in L(G)$ давать на выходе вывод w в G. С точки зрения теории синтаксического анализа CYK-алгоритм проводит *восходящий (bottom—up) анализ*.

Замечание 2 (о сложности CYK-алгоритма)**.** Нетрудно видеть, что сложность CYK-алгоритма может быть оценена как $O(n^3 \cdot |\mathcal{P}|)$, что ограничивает применение алгоритма на практике. Чаще всего в приложениях рассматривается подкласс КС-грамматик, *детерминированные КС-грамматики* (по-другому, $LL(k)$- и $LR(k)$-грамматики), для которых существуют линейные алгоритмы разбора (сложность $O(n)$).

Утверждение. Проблема эквивалентности КС-грамматик является неразрешимой.