

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»

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  $\Sigma^*$ , где (a)  $\Sigma = \{0, 1\}$ ; (b)  $\Sigma$  — произвольный (конечный) алфавит;
- (2) произвольный конечный язык  $L = \{\omega_i\}_{i=1}^n$ ;
- (3)  $\{a^+b^+\}$ ,  $\{a^n b^n \mid n \in \mathbb{N}\}$ ,  $\{a^n b^n a^m \mid m, n \in \mathbb{N}\}$ ;  $\{a^n b^n c^n \mid n \in \mathbb{N}\} — \otimes^1$ ;
- (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  $\{a + nd \mid n \in \mathbb{N}_0\}$ ,  $d > 0$ ,  $0 \leq a < d$  (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.

---

<sup>1</sup>Задания, отмеченные  $\otimes$ , — для самостоятельного выполнения.

## ЗАНЯТИЕ 2

### 1. Системы линейных уравнений с регулярными коэффициентами

Решить системы уравнений:

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

$$\begin{cases} X_1 = 1X_1 + 0X_2 + \emptyset X_3; \\ X_2 = 1X_1 + \emptyset X_2 + 0X_3; \\ X_3 = 0X_1 + \emptyset X_2 + 1X_3. \end{cases} \quad (2)$$

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

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

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

$$\begin{aligned} S &\rightarrow 1A \mid 2S; \\ A &\rightarrow 0B \mid 0S \mid 1A; \\ B &\rightarrow 1C \mid 2C; \\ C &\rightarrow \varepsilon \mid 1S \mid 2A. \end{aligned} \quad (5)$$

$$\begin{aligned} S &\rightarrow 0A \mid 1S \mid \varepsilon; \\ A &\rightarrow 0B \mid 1A; \\ B &\rightarrow 0S \mid 1B. \end{aligned} \quad (6)$$

### 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  $|w| \geq n$ , существует такое разбиение  $w = xyz$  слова  $w$ , что:

- (1)  $y \neq \varepsilon$ ;
- (2)  $|xy| \leq n$ ;
- (3)  $\{xy^kz \mid k \geq 0\} \subset L$ .

*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)  $\{0^n 1^m \mid n \leq m\}$ ;
- (5)  $\{1^p \mid p \text{ — простое}\}$ ;
- (6)  $\{0^n 10^n \mid n \in \mathbb{N}\}$ ;
- (7)  $\{1^{n^2} \mid n \in \mathbb{N}\}$ ;
- (8)  $\{1^{n!} \mid n \in \mathbb{N}\}$ .

## ЗАНЯТИЕ 3: Нахождение языка конечного автомата

**Задача:** описать язык заданного конечного автомата регулярным выражением.

**Способ 1.**

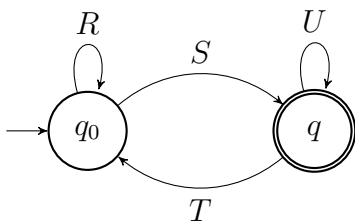
(Объект реального мира  $\xrightarrow{0}$ )  
 конечный автомат  $\xrightarrow{1}$   
 граф переходов конечного автомата  $\xrightarrow{2}$   
 ПЛ-грамматика  $\xrightarrow{3}$   
 система линейных уравнений  $\xrightarrow{4}$   
 решение системы.

**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 начальном графе на дугах простейшие регулярные выражения — однобуквенные).

*Процедура исключения состояния  $s$ :* для каждого 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  $p \rightarrow q$  отсутствовали в исходном графе, то можно считать, что их метки равны  $\emptyset$ ) — таким образом получена дуга с меткой:  $p \xrightarrow{R_3 + R_1 R^* R_2} q$ . Удалить все просмотренные дуги  $p \rightarrow s$  и  $s \rightarrow q$ , инцидентные вершине  $s$ . После этого  $s$  стала изолированной либо имеются только входящие 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.*

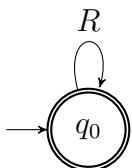
1. Для каждого финального состояния  $q \in F$ , отличного от начального  $q_0$ , применять процедуру исключения состояний до тех пор, пока не останутся две вершины:  $q_0$  и  $q$ . В результате получится подобный автомат:



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
$\rightarrow q_0$	$q_1$	$q_0$
$q_1$	$q_2$	$q_0$
$q_2$	$q_2$	$q_1$

(4)

	0	1
$\rightarrow q_0$	$q_1$	$q_2$
$q_1$	$q_0$	$q_2$
$q_2$	$q_1$	$q_0$

(5)

	0	1
$\rightarrow p$	$s$	$p$
$q$	$p$	$s$
$r$	$r$	$q$
$s$	$q$	$r$

## 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  $\text{RE}(L_1), \text{RE}(L_2)$ . Язык объединения (соответственно, конкатенации, итерации) описывается выражением  $\text{RE}(L_1)+\text{RE}(L_2)$  (соответственно,  $\text{RE}(L_1)\text{RE}(L_2), \text{RE}(L_1)^*$ ), а значит, регулярен.

Для регулярного языка  $L$  построим распознающий его детерминированный<sup>2</sup> конечный автомат  $\mathcal{A}(L) = (Q, \Sigma, \delta, q_0, F)$ . Легко проверить, что автомат  $\tilde{\mathcal{A}}(L) = (Q, \Sigma, \delta, q_0, Q \setminus F)$  допускает дополнение  $\bar{L} = \Sigma^* \setminus L$  языка  $L \otimes$ , которое является, таким образом, регулярным языком.

Поскольку справедливо  $L_1 \cap L_2 = \overline{\bar{L}_1 \cup \bar{L}_2}$ , язык  $L_1 \cap L_2$  регулярен по доказанному выше<sup>3</sup>. Аналогичное можно заключить из равенства  $L_1 \setminus L_2 = L_1 \cap \bar{L}_2$ . ■

Доказать (не)регулярность:

- (1)  $\{0^n 1^m \mid n \neq m\}$ ;
- (2)  $\{w \in \{0, 1\}^* \mid w \text{ содержит одинаковое число } 0 \text{ и } 1\}$ ;
- (3) языка из слов  $w \in \{0, \dots, 9\}^*$ , которые являются десятичной записью чисел, делящихся на 2 или на 3, без лишних лидирующих нулей;
- (4)  $\{0^n 1^m 2^{n-m} \mid n \geq m\}$ ;
- (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.

<sup>2</sup>Сделанное дальше утверждение, однако, неверно для недетерминированного конечного автомата — постройте соответствующий контрпример. ⊗

<sup>3</sup>Имеется более полезная конструкция, которая строит ДКА, допускающий  $L_1 \cap L_2$ , внутри которого «параллельно» работают  $\mathcal{A}(L_1)$  и  $\mathcal{A}(L_2)$  — попробуйте 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  $\{0, 1, \dots, 9\}$  из слов, в которых последняя цифра встречается ещё где-то в них;
- (5) язык над  $\{0, 1, \dots, 9\}$  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 (ноль 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):

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

где:

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

Тогда

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

*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
	$\rightarrow p$	$\{p, q\}$	$\{p\}$
	$q$	$\{r\}$	$\{r\}$
	$r$	$\{s\}$	$\emptyset$
	<span style="border: 1px solid black; padding: 2px;"><math>s</math></span>	$\{s\}$	$\{s\}$

(4)		0	1
	$\rightarrow p$	$\{q, s\}$	$\{q\}$
	<span style="border: 1px solid black; padding: 2px;"><math>q</math></span>	$\{r\}$	$\{q, r\}$
	$r$	$\{s\}$	$\{p\}$
	<span style="border: 1px solid black; padding: 2px;"><math>s</math></span>	$\emptyset$	$\{p\}$

(5)		0	1
	$\rightarrow p$	$\{p, q\}$	$\{p\}$
	$q$	$\{r, s\}$	$\{t\}$
	$r$	$\{p, r\}$	$\{t\}$
	<span style="border: 1px solid black; padding: 2px;"><math>s</math></span>	$\emptyset$	$\emptyset$
	<span style="border: 1px solid black; padding: 2px;"><math>t</math></span>	$\emptyset$	$\emptyset$

## 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

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

(2) *порождающий* в  $G$ , если

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

(3) *достижимый* в  $G$ , если

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

*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:

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

**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)  $A \rightarrow a,$

2)  $A \rightarrow BC,$

где  $A, B, C \in N, a \in \Sigma$ .

**Схема приведения грамматики к НФХ:**

(1) удалить  $\varepsilon$ -продукции;

- (2) удалить цепные продукции (продукции вида  $A \rightarrow B$ );
- (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:

$\begin{aligned} & S \rightarrow ASB \mid \varepsilon, \\ (1) \quad & A \rightarrow aAS \mid a, \\ & B \rightarrow SbS \mid A \mid bb; \end{aligned}$	$\begin{aligned} & S \rightarrow 0A0 \mid 1B1 \mid BB, \\ (2) \quad & A \rightarrow C, \\ & B \rightarrow S \mid A, \\ & C \rightarrow S \mid \varepsilon; \end{aligned}$
$\begin{aligned} & S \rightarrow AAA \mid B, \\ (3) \quad & A \rightarrow aA \mid B, \\ & B \rightarrow \varepsilon; \end{aligned}$	$\begin{aligned} & S \rightarrow aAa \mid bBb \mid \varepsilon, \\ (4) \quad & A \rightarrow C \mid a, \\ & B \rightarrow C \mid b, \\ & C \rightarrow CDE \mid \varepsilon, \\ & D \rightarrow A \mid B \mid ab. \end{aligned}$

## ЗАНЯТИЕ 7

**1. Другое определение нормальной формы Хомского.** Заметим, что грамматика в НФХ не может порождать  $\varepsilon$ , однако мы знаем КС-языки, содержащие  $\varepsilon$ . 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  $\varepsilon$ .

**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)  $A \rightarrow a$ ,
- (2)  $A \rightarrow BC$ ,
- (3)  $S \rightarrow \varepsilon$ ,

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  $\varepsilon$ -правил шаг 4:

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

⊗ 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

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

- (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  $\text{Gen}_G(\Sigma)$  и проверке  $S \in \text{Gen}_G(\Sigma)$ . Рассмотрим один алгоритм, решающий проблему принадлежности для КС-языков.

**Алгоритм** (Кок—Янгер—Касами, «СҮК-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  $1 \leq i \leq j \leq n$  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  $N_{ii} \leftarrow \{A \in N \mid A \rightarrow w_i \in \mathcal{P}\} \triangleright$  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  $s$ 
       $N_{ij} \leftarrow \{A \in N \mid A \rightarrow BC \in \mathcal{P}; \exists k \in [i, j - 1]_{\mathbb{Z}}: B \in N_{ik}, C \in N_{k+1j}\}$ 

```

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

Используя СҮК-алгоритм,

(1) для грамматики  $G$  с продукциями:

$$S \rightarrow AB, \quad A \rightarrow BB \mid a, \quad B \rightarrow AB \mid b$$

определить, принадлежат ли  $L(G)$  строки: (а)  $aabbb$ , (б)  $babab$ , (в)  $b^7$ ;

(2) для грамматики  $G$  с продукциями:

$$S \rightarrow AB \mid BC, \quad A \rightarrow BA \mid a, \quad B \rightarrow CC \mid b, \quad C \rightarrow AB \mid a$$

определить, принадлежат ли  $L(G)$  строки: (а)  $ababa$ , (б)  $baaab$ , (в)  $aabab$ .

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

**Замечание 2** (о сложности СҮК-алгоритма). Нетрудно видеть, что сложность СҮК-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.