

## Математическая теория формальных языков

### 0. Лекция: Предисловие: версия для печати и PDA

Цель этого курса - познакомить читателя с некоторыми основополагающими моделями и результатами, используемыми в теоретической информатике. Неудивительно, что они относятся к математике, а не к какой-либо другой области знаний - ведь в науке о компьютерах именно математические абстракции являются самыми плодотворными.

Рассматриваемые здесь идеи и результаты принадлежат теории формальных языков, грамматик и автоматов. По существу, эта теория описывает некоторые ограниченные абстрактные машины, способные выполнять определенные операции со строками. Например, конечный автомат может выяснить, содержит ли некоторый файл определенное слово, а автомат с магазинной памятью способен определить, правильна ли система вложенных круглых, квадратных и фигурных скобок.

Как подсказывает само название курса, основным объектом рассмотрения является формальный язык - произвольное множество конечных последовательностей символов, взятых из некоторого конечного множества (такие последовательности называются словами). Важность формальных языков для теоретической информатики обусловлена тем, что наиболее простой и удобной моделью данных, используемых в компьютерных программах, является конечная последовательность, каждый элемент которой взят из некоторого заранее зафиксированного конечного множества (так как для хранения этого элемента отводится фиксированный объем памяти).

В первой лекции дается классификация формальных языков в соответствии с иерархией Хомского: автоматные языки, контекстно-свободные (или бесконтекстные) языки, контекстные (или контекстно-зависимые) языки, языки типа 0. Если исключить языки, содержащие пустое слово, то названные классы вложены друг в друга (здесь они перечислены в возрастающем порядке).

В лекциях [2-6](#) рассматривается самый узкий из этих классов - класс автоматных (или праволинейных, или регулярных) языков, обладающий многими свойствами, важными для приложений. Например, разрешима проблема равенства автоматных языков (естественно, надо сначала договориться о способе конечного задания языка), пересечение двух автоматных языков тоже является автоматным и т.д. Поэтому важно иметь удобные критерии автоматности, то есть условия, являющиеся одновременно необходимыми и достаточными для того, чтобы язык распознавался некоторым конечным автоматом. Наиболее известные критерии изложены в первых лекциях, их можно найти по предметному указателю (язык, автоматный, критерий автоматности). Именно автоматные языки лежат в основе лексических анализаторов, входящих в структуру компиляторов и интерпретаторов языков программирования.

В лекциях [7-11](#) рассматриваются контекстно-свободные языки. Изучаются разные способы выделения формального языка из множества всех слов, оказывающиеся эквивалентными друг другу в том смысле, что они задают один и тот же класс языков, а именно класс всех контекстно-свободных языков. При определении современных языков программирования (а также языков поисковых запросов и т. п.) основная часть синтаксиса описывается в формализме, известном как форма Бэкуса-Наура. По существу, это способ записывать контекстно-свободные грамматики. Другая важная область использования контекстно-свободных грамматик - расширяемый язык разметки (XML) и языки описания структуры XML-документа (например, язык DTD).

Два оставшихся класса из иерархии Хомского - контекстные языки и языки типа 0 - в данном курсе изучаются менее подробно, так как они относятся скорее к теории

сложности вычислений и к теории алгоритмов соответственно. Приводятся эквивалентные определения этих классов в терминах автоматов.

Лекции [12](#) и [13](#) содержат основные результаты теории детерминированных контекстно-свободных языков и формальные определения, демонстрирующие связь этой теории с нисходящими и восходящими синтаксическими анализаторами.

Последние три лекции курса посвящены алгоритмическим проблемам, связанным с формальными грамматиками и автоматами. На основе известной из теории алгоритмов теоремы о неразрешимости проблемы Поста доказывается неразрешимость многих важных проблем теории формальных языков, например проблемы однозначности контекстно-свободной грамматики и проблемы эквивалентности двух контекстно-свободных грамматик. С другой стороны, формулируются и положительные результаты о разрешимости, в том числе теорема о разрешимости проблемы равенства детерминированных контекстно-свободных языков.

© 2003-2008 INTUIT.ru. Все права защищены.

## 0. Предисловие

### 1. Слова, языки и грамматики

В данной лекции определяются такие основные понятия, как алфавит, слово, язык над данным алфавитом - и вводятся обозначения для пустого слова, конкатенации слов, степени слова, длины слова, количества вхождений данного символа. Приведены практические примеры и упражнения для закрепления материала

### 2. Конечные автоматы

В данной лекции рассматриваются два наиболее распространенных способа конечного задания формального языка: грамматики и автоматы. Рассматриваются конечные автоматы, соответствующие в иерархии Хомского праволинейным грамматикам, определяются понятия конечного автомата, недетерминированного конечного автомата и распознаваемого конечным автоматом языка. Приведены практические примеры и упражнения для закрепления материала

### 3. Основные свойства автоматных языков

В данной лекции доказываются свойства замкнутости класса всех автоматных языков относительно итерации, конкатенации, объединения, дополнения и пересечения, доказывается так называемая лемма о разрастании, которая во многих случаях позволяет установить неавтоматность формального языка. Приведены практические примеры и предоставлены упражнения для самостоятельного решения

### 4. Дополнительные свойства автоматных языков

В данной лекции доказывается замкнутость класса всех автоматных языков относительно взятия гомоморфного образа и относительно взятия полного гомоморфного прообраза, определяются понятия побуквенного гомоморфизма и локального языка. Приведены практические примеры и предоставлены упражнения для самостоятельного решения

### 5. Регулярные выражения

В данной лекции рассматривается наиболее удобный и компактный способ конечного описания формального языка - регулярные выражения, - который находит практическое применение во многих компьютерных приложениях, таких как текстовые редакторы, интерпретаторы командной строки и автоматические генераторы лексических анализаторов. Приведены практические примеры и предоставлены упражнения для самостоятельного решения

### 6. Синтаксические моноиды

В данной лекции рассматривается понятие синтаксического моноида и приводится доказательство еще одного критерия автоматности формального языка, который можно сформулировать в терминах классов эквивалентности слов по взаимозаменяемости. Приведены примеры практической реализации критерия и предоставлены упражнения для самостоятельного решения

### 7. Неоднозначность в контекстно-свободных грамматиках

В данной лекции рассматривается класс контекстно-свободных языков, которые находят применение в разнообразных программных продуктах, таких как компиляторы, средства форматирования исходного кода, средства статического анализа программ, синтаксические редакторы, системы верстки, программы просмотра форматированного текста, поисковые системы. Приведены практические примеры и упражнения для самостоятельного решения

### 8. Нормальные формы контекстно-свободных грамматик

В данной лекции рассматривается доказательство того, что каждая контекстно-свободная грамматика эквивалентна некоторой контекстно-свободной грамматике специального вида, а именно грамматике в нормальной форме Хомского. Этот факт используется в доказательствах многих теорем о контекстно-свободных языках и является очень важным. Приведены также практические примеры и самостоятельные упражнения

### 9. Основные свойства контекстно-свободных языков

В данной лекции рассматриваются основные свойства контекстно-свободных языков, приведены некоторые теоремы для класса линейных языков, а также доказательство того, что пересечение контекстно-свободного языка с автоматным языком является контекстно-свободным. Также приводятся практические примеры и упражнения для самостоятельной проработки

### 10. Автоматы с магазинной памятью

В данной лекции рассматриваются автоматы с магазинной памятью, в которых помимо ограниченной памяти, хранящей текущее состояние, имеется потенциально бесконечная память, используемая как магазин. Даются необходимые определения, и доказывается, что автоматы с магазинной памятью распознают в точности контекстно-свободные языки. Приведены примеры практического использования автоматов с магазинной памятью и приведены упражнения для самостоятельной проработки

#### 11. [Дополнительные свойства контекстно-свободных языков](#)

В данной лекции рассматриваются свойства контекстно-свободных языков, которые удобно доказывать с привлечением автоматов с магазинной памятью, а также формулируются два критерия контекстной свободности, интересных в основном с теоретической точки зрения. Приведены практические примеры и упражнения для самостоятельного решения

#### 12. [Детерминированные контекстно-свободные языки](#)

В данной лекции рассматриваются детерминированные контекстно-свободные языки. Формулируется ряд свойств этого класса языков, рассматривается важность детерминированных контекстно-свободных языков для теоретической информатики, которая заключается в том, что для каждого такого языка можно указать быстрый алгоритм, распознающий принадлежность слова этому языку. Приведены практические примеры и упражнения для самостоятельного решения

#### 13. [Синтаксический разбор](#)

В данной лекции рассматриваются формальные определения, связанные с прямыми синтаксическими анализаторами, определяется понятие синтаксического разбора и связанные с ним специфические определения. Приведены практические примеры и упражнения для самостоятельной проработки

#### 14. [Алгоритмические проблемы](#)

В данной лекции рассматриваются понятия вычислимости, разрешимости, перечислимости и универсальных моделей вычислений. Рассмотрены алгоритмические проблемы, связанные с контекстно-свободными грамматиками, вводятся термины "массовая задача", "индивидуальная задача", "схема кодирования", "задача распознавания", "алгоритмическая проблема". Приведены практические примеры и задачи для самостоятельной проработки

#### 15. [Алгоритмически разрешимые проблемы](#)

В данной лекции рассматриваются наиболее известные разрешимые проблемы, связанные с грамматиками, автоматами и регулярными выражениями. Сформулированы также теоремы о разрешимости для некоторых алгоритмических проблем. Приведены практические примеры и упражнения для самостоятельного решения

#### 16. [Алгоритмически неразрешимые проблемы](#)

В данной лекции рассматриваются оказавшиеся неразрешимыми алгоритмические проблемы, связанные с контекстно-свободными языками. Доказывается неразрешимость проблем пустоты пересечения, бесконечности пересечения, однозначности, равенства, автоматности, контекстной свободности дополнения и пересечения контекстно-свободного языка. Доказательства приведены на практических примерах, а также приведены упражнения для самостоятельной проработки.

## Математическая теория формальных языков

### 1. Лекция: Слова, языки и грамматики: версия для печати и PDA

В данной лекции определяются такие основные понятия, как алфавит, слово, язык над данным алфавитом - и вводятся обозначения для пустого слова, конкатенации слов, степени слова, длины слова, количества вхождений данного символа. Приведены практические примеры и упражнения для закрепления материала

В начале этой лекции определяются основные понятия, используемые в дальнейшем: алфавит, слово, язык над данным алфавитом - и вводятся обозначения для пустого слова, конкатенации слов, степени слова, длины слова, количества вхождений данного символа. В разделе 1.2 фиксируются обозначения для префикса и суффикса слова, а также для ряда операций над языками, таких как конкатенация, итерация, обращение. При беглом чтении раздел 1.3, где определяются образ и полный прообраз языка при гомоморфизме, можно пропустить: вводимые в этом разделе понятия понадобятся только в лекциях [4](#), [11](#) и [13](#).

Используемые в приложениях формальные языки, как правило, являются бесконечными. Очевидно, нужен способ конечного описания формального языка. В данном курсе изучаются несколько классических средств: порождающие грамматики, автоматы, регулярные выражения. Определению понятий грамматики и порождаемого ею языка посвящен раздел 1.4. В конце первой лекции вводятся классы грамматик, образующие иерархию Хомского: грамматики типа 0 (все порождающие грамматики), грамматики типа 1 (контекстные грамматики), грамматики типа 2 (контекстно-свободные, или бесконтекстные, грамматики), грамматики типа 3 (праволинейные грамматики), а также менее важный класс линейных грамматик, промежуточный между классами грамматик типа 2 и 3.

Следует заметить, что каждая грамматика порождает ровно один язык, но обратное неверно: некоторые формальные языки нельзя задать никакой порождающей грамматикой, а каждому языку, который порождается хотя бы одной грамматикой, соответствует сразу бесконечное множество грамматик (причем они могут принадлежать разным классам).

### 1.1. Формальные языки

**Определение 1.1.1.** Будем называть **натуральными числами** неотрицательные целые числа. Множество всех натуральных чисел  $\{0, 1, 2, \dots\}$  обозначается  $\mathbb{N}$ .

**Определение 1.1.2.** *Алфавитом* называется конечное непустое множество. Его элементы называются **символами** (**буквами**).

**Определение 1.1.3.** **Словом** (цепочкой, строкой, string) в алфавите  $\Sigma$  называется конечная последовательность элементов  $\Sigma$ .

**Пример 1.1.4.** Рассмотрим алфавит  $\Sigma = \{a, b, c\}$ . Тогда baаа является словом в алфавите  $\Sigma$ .

**Определение 1.1.5.** Слово, не содержащее ни одного символа (то есть последовательность длины 0), называется **пустым словом** и обозначается  $\epsilon$ .

**Определение 1.1.6.** Множество всех слов в алфавите  $\Sigma$  обозначается  $\Sigma^*$ .

**Замечание 1.1.7.** Множество  $\Sigma^*$  счетно. В самом деле, в алфавите  $\Sigma$  множество всех

слов данной длины конечно, следовательно,  $\Sigma^*$  является объединением счетного числа конечных множеств.

**Определение 1.1.8.** Множество всех непустых слов в алфавите  $\Sigma$  обозначается  $\Sigma^+$ .

**Пример 1.1.9.** Если  $\Sigma = \{a\}$ , то  $\Sigma^+ = \{a, aa, aaa, aaaa, \dots\}$ .

**Определение 1.1.10.** Если  $L \subseteq \Sigma^*$ , то  $L$  называется **языком** (или **формальным языком**) над алфавитом  $\Sigma$ .

Поскольку каждый язык является множеством, можно рассматривать операции объединения, пересечения и разности языков, заданных над одним и тем же алфавитом (обозначения  $L_1 \cup L_2, L_1 \cap L_2, L_1 - L_2$ ).

**Пример 1.1.11.** Множество  $\{a, abb\}$  является языком над алфавитом  $\{a, b\}$ .

**Определение 1.1.12.** Пусть  $L \subseteq \Sigma^*$ . Тогда язык  $\Sigma^* - L$  называется **дополнением** языка  $L$  относительно алфавита  $\Sigma$ . Когда из контекста ясно, о каком алфавите идет речь, говорят просто, что язык  $\Sigma^* - L$  является дополнением языка  $L$ .

**Определение 1.1.13.** Если  $x$  и  $y$  - слова в алфавите  $\Sigma$ , то слово  $xy$  (результат приписывания слова  $y$  в конец слова  $x$ ) называется **конкатенацией**, (**катенацией**, **сцеплением**) слов  $x$  и  $y$ . Иногда конкатенацию слов  $x$  и  $y$  обозначают  $x \cdot y$ .

**Определение 1.1.14.** Если  $x$  - слово и  $n \in \mathbb{N}$ , то через  $x^n$  обозначается слово

$\underbrace{x \cdot x \cdot \dots \cdot x}_{n \text{ раз}}$ . Положим  $x^0 = \varepsilon$  (знак  $=$  читается "равно по определению"). Всюду далее показатели над словами и символами, как правило, являются натуральными числами.

**Пример 1.1.15.** По принятым соглашениям  $ba^3 = baaa$  и  $(ba)^3 = bababa$ .

**Пример 1.1.16.** Множество  $\{a^k b a^l \mid k \leq l\}$  является языком над алфавитом  $\{a, b\}$ . Этот язык содержит слова  $b, ba, aba, baa, abaa, baaa, aabaa, abaaa, baaaa$  и т. д.

**Определение 1.1.17.** **Длина** слова  $w$ , обозначаемая  $|w|$ , есть число символов в  $w$ , причем каждый символ считается столько раз, сколько раз он встречается в  $w$ .

**Пример 1.1.18.** Очевидно, что  $|baaa| = 4$  и  $|\varepsilon| = 0$ .

**Определение 1.1.19.** Через  $|w|_a$  обозначается количество вхождений символа  $a$  в слово  $w$ .

**Пример 1.1.20.** Если  $\Sigma = \{a, b, c\}$ , то  $|baaa|_a = 3$ ,  $|baaa|_b = 1$  и  $|baaa|_c = 0$ .

**Упражнение 1.1.21.** Перечислить слова языка  $L_1 \cap L_2$ , где  $L_1 = \{(ab)^n \mid n \geq 0\}$  и  $L_2 = \{a^m b^m \mid m \geq 0\}$ .

**Упражнение 1.1.22.** Пусть  $\Sigma = \{a, b, c\}$ . Равны ли языки  $\{(abc)^n a \mid n \geq 2\}$  и  $L_2 = \{ab(cab)^n ca \mid n \geq 1\}$ ?

**Упражнение 1.1.23.** Пусть  $\Sigma = \{a, b, c\}$ ,  $L_1 = \{w \in \Sigma^* \mid |w| = 4\}$  и  $L_2 = \{w \in \Sigma^* \mid |w|_c = 1\}$ . Сколько слов содержит язык  $L_1 \cap L_2$ ?

**Упражнение 1.1.24.** Пусть даны такие алфавиты  $\Sigma_1$  и  $\Sigma_2$ , что  $\Sigma_1 \cap \Sigma_2 \neq \emptyset$ . В каком случае  $\Sigma_1^+ \cap \Sigma_2^+ = (\Sigma_1 \cap \Sigma_2)^+$ ?

## 1.2. Операции над языками

**Определение 1.2.1.** Пусть  $L_1, L_2 \subseteq \Sigma^*$ . Тогда  $L_1 \cdot L_2 = \{xy \mid x \in L_1, y \in L_2\}$ .

Язык  $L_1 \cdot L_2$  называется **конкатенацией** языков  $L_1$  и  $L_2$ .

**Пример 1.2.2.** Если  $L_1 = \{a, abb\}$  и  $L_2 = \{bbc, c\}$ , то  $L_1 \cdot L_2 = \{ac, abbc, abbbbc\}$ .

**Упражнение 1.2.3.** При каких положительных целых числах  $k, l, m, n$  существуют алфавит  $\Sigma$ , язык  $L_1 \subseteq \Sigma^*$  и язык  $L_2 \subseteq \Sigma^*$ , удовлетворяющие условиям  $|\Sigma| = k$ ,  $|L_1| = l$ ,  $|L_2| = m$ ,  $|L_1 \cdot L_2| = n$ ?

**Определение 1.2.4.** Пусть  $L \subseteq \Sigma^*$ . Тогда

$$L^0 = \{\varepsilon\},$$

$$L^n = \underbrace{L \cdot \dots \cdot L}_n, \text{ если } n > 0.$$

**Пример 1.2.5.** Если  $L = \{a^k b a^l \mid 0 < k < l\}$ , то  $L^2 = \{a^k b a^l b a^m \mid 0 < k < l - 1, m > 1\}$ .

**Упражнение 1.2.6.** Пусть  $\Sigma = \{a, b\}$  и  $L = \{aa, ab\}$ . Найти  $L^3$ .

**Определение 1.2.7.** Итерацией языка (Kleene closure) языка  $L$  (обозначение  $L^*$ ) называется язык

$$\bigcup_{n \in \mathbb{N}} L^n.$$

Эта операция называется также **звездочкой Клини** (Kleene star, star operation).

**Пример 1.2.8.** Если  $\Sigma = \{a, b\}$  и  $L = \{aa, ab, ba, bb\}$ , то  $L^* = \{w \in \Sigma^* \mid |w| \geq 2\}$ .

**Упражнение 1.2.9.** Пусть  $\Sigma = \{a, b, c, d\}$  и  $L = \{w \in \Sigma^* \mid |w|_a = 1, |w|_c = 1\}$ .

Верно ли, что  $abcdcacdcabbacba \in L^*$ ?



**Упражнение 1.2.10.** Существует ли такой язык  $L$ , что выполняется неравенство

$$L^* \neq \{x^n \mid x \in L, n \geq 0\}$$

**Определение 1.2.11.** **Обращением** или **зеркальным образом** слова  $w$  (обозначается  $w^R$ ) называется слово, в котором символы, составляющие слово  $w$ , идут в обратном порядке.

**Пример 1.2.12.** Если  $w = baaca$ , то  $w^R = acaab$ .

**Определение 1.2.13.** Пусть  $L \subseteq \Sigma^*$ . Тогда

$$L^R = \{w^R \mid w \in L\}.$$

Язык  $L^R$  называется **обращением** языка  $L$ .

**Упражнение 1.2.14.** Существует ли такой язык  $L$ , что выполняется неравенство

$$(L^R)^* \neq (L^*)^R?$$

**Определение 1.2.15.** Говорят, что слово  $x$  - **префикс (начало)** слова  $y$  (обозначение  $x \sqsubset y$ ), если  $y = xu$  для некоторого слова  $u$ .

**Пример 1.2.16.** Очевидно, что  $\varepsilon \sqsubset baa$ ,  $b \sqsubset baa$ ,  $ba \sqsubset baa$  и  $baa \sqsubset baa$ .

**Определение 1.2.17.** Пусть  $L \subseteq \Sigma^*$ . Тогда через  $\text{Pref}(L)$  обозначается множество, состоящее из всех префиксов слов языка  $L$ :

$$\text{Pref}(L) = \{x \mid (\exists y \in L) x \sqsubset y\}.$$

Множество  $\text{Pref}(L)$  называется **множеством префиксов** языка  $L$ .

**Определение 1.2.18** Говорят, что слово  $x$  - **суффикс (конец)** слова  $y$  (обозначение  $x \sqsupset y$ ), если  $y = ux$  для некоторого слова  $u$ .

**Определение 1.2.19.** Пусть  $L \subseteq \Sigma^*$ . Тогда через  $\text{Suf}(L)$  обозначается множество, состоящее из всех суффиксов слов языка  $L$ :

$$\text{Suf}(L) = \{x \mid (\exists y \in L) x \sqsupset y\}.$$

Множество  $\text{Suf}(L)$  называется **множеством суффиксов** языка  $L$ .

**Определение 1.2.20.** Говорят, что слово  $x$  - **подслово (substring)** слова  $y$ , если  $y = uxv$  для некоторых слов  $u$  и  $v$ .

**Определение 1.2.21.** Пусть  $L \subseteq \Sigma^*$ . Тогда через  $\text{Subw}(L)$  обозначается множество, состоящее из всех подслов слов языка  $L$ . Множество  $\text{Subw}(L)$  называется **множеством подслов** языка  $L$ .

**Определение 1.2.22.** Слово  $a_1a_2 \dots a_n$  (длины  $n \geq 0$ ) называется **подпоследовательностью** (subsequence) слова  $y$ , если существуют такие слова  $u_0, u_1, \dots, u_n$ , что  $u_0a_1u_1a_2 \dots a_nu_n = y$ .

**Замечание 1.2.23.** Все подслова слова  $y$  являются также подпоследовательностями слова  $y$ .

**Определение 1.2.24.** Пусть  $L \subseteq \Sigma^*$ . Тогда через  $\text{Subseq}(L)$  обозначается множество, состоящее из всех подпоследовательностей слов языка  $L$ . Множество  $\text{Subseq}(L)$



называется **множеством подпоследовательностей** языка  $L$ .

**Пример 1.2.25.** Рассмотрим язык  $L = \{cba, c\}$  над алфавитом  $\{a, b, c\}$ . Очевидно, что  $\text{Subseq}(L) = \{\varepsilon, a, b, c, ba, ca, cb, cba\}$ .

**Определение 1.2.26.** Функция  $f: K \rightarrow L$  называется **биекцией** (bijection), если каждый элемент множества  $L$  является образом ровно одного элемента множества  $K$  (относительно функции  $f$ ).

**Определение 1.2.27.** Множества  $K$  и  $L$  называются **равномощными** (of equal cardinality), если существует биекция из  $K$  в  $L$ .

**Упражнение 1.2.28.** Существуют ли такие языки  $L_1$  и  $L_2$ , что языки  $L_1 \cdot (L_2^R)$  и  $L_1^R \cdot L_2$  неравномощны?

**Упражнение 1.2.29.** Существуют ли такие языки  $L_1$  и  $L_2$ , что языки  $L_1^R \cdot L_2$  и  $L_2^R \cdot L_1$  неравномощны?

### 1.3. Гомоморфизмы

**Определение 1.3.1.** Пусть  $\Sigma_1$  и  $\Sigma_2$  - алфавиты. Если отображение  $h: \Sigma_1^* \rightarrow \Sigma_2^*$  удовлетворяет условию  $h(x \cdot y) = h(x) \cdot h(y)$  для всех слов  $x \in \Sigma_1^*$  и  $y \in \Sigma_1^*$ , то отображение  $h$  называется **гомоморфизмом** (**морфизмом**).

**Замечание 1.3.2.** Можно доказать, что если  $h$  - гомоморфизм, то  $h(\varepsilon) = \varepsilon$ .

**Замечание 1.3.3.** Пусть  $\Sigma_1 = \{a, b\}$  и  $\Sigma_2 = \{c\}$ . Тогда отображение  $h: \Sigma_1^* \rightarrow \Sigma_2^*$ , заданное равенством  $h(w) = c^{2|w|}$ , является гомоморфизмом.

**Замечание 1.3.4.** Каждый гомоморфизм однозначно определяется своими значениями на однобуквенных словах.

**Замечание 1.3.5.** Если  $h: \Sigma_1^* \rightarrow \Sigma_2^*$  - гомоморфизм и  $L \subseteq \Sigma_1^*$ , то через  $h(L)$  обозначается язык  $\{h(w) \mid w \in L\}$ .

**Пример 1.3.6.** Пусть  $\Sigma = \{a, b\}$  и гомоморфизм  $h: \Sigma^* \rightarrow \Sigma^*$  задан равенствами  $h(a) = abba$  и  $h(b) = \varepsilon$ . Тогда  $h(\{baa, bb\}) = \{abbaabba, \varepsilon\}$ .

**Определение 1.3.7.** Если  $h: \Sigma_1^* \rightarrow \Sigma_2^*$  - гомоморфизм и  $L \subseteq \Sigma_2^*$ , то через  $h^{-1}(L)$  обозначается язык  $\{w \in \Sigma_1^* \mid h(w) \in L\}$ .

**Пример 1.3.8.** Рассмотрим алфавит  $\Sigma = \{a, b\}$ . Пусть гомоморфизм  $h: \Sigma^* \rightarrow \Sigma^*$  задан равенствами  $h(a) = ab$  и  $h(b) = abb$ . Тогда  $h^{-1}(\{\varepsilon, abbb, abbab, ababab\}) = \{\varepsilon, ba, aaa\}$ .

**Упражнение 1.3.9.** Пусть  $\Sigma = \{a, b, c, d\}$ . Существует ли такой гомоморфизм  $h: \Sigma^* \rightarrow \Sigma^*$ , что  $h(abc) = bac$  и  $h(da) = ba$ ?

**Упражнение 1.3.10.** Существуют ли такой язык  $L \subseteq \{a, b\}^+$  и такой гомоморфизм  $h: \{a, b\}^* \rightarrow \{a, b\}^*$ , что  $h(h^{-1}(L)) = L$  и  $h^{-1}(h(L)) \neq L$ ?

**Упражнение 1.3.11.** Пусть  $\Sigma = \{a, b\}$ . Существует ли такой гомоморфизм  $h: \Sigma^* \rightarrow \Sigma^*$ , что  $h(h^{-1}(\{\varepsilon\})) = \{\varepsilon\}$  и  $h^{-1}(h(\{\varepsilon\})) \neq \{\varepsilon\}$ ?

**Упражнение 1.3.12.** Существуют ли такой язык  $L \subseteq \{a, b\}^+$  и такой гомоморфизм  $h: \{a, b\}^* \rightarrow \{a, b\}^*$ , что  $h(h^{-1}(L)) \neq L$  и  $h^{-1}(h(L)) = L$ ?

**Упражнение 1.3.13.** Пусть  $h: \{a, b, c\}^* \rightarrow \{a, b\}^*$  - гомоморфизм, заданный соотношениями  $h(a) = a$ ,  $h(b) = ba$ ,  $h(c) = bb$ . Существуют ли такие слова  $u$  и  $v$ , что  $|u| < |v|$  и  $|h(u)| > |h(v)|$ ?

## 1.4. Порождающие грамматики

**Определение 1.4.1.** Порождающей грамматикой (грамматикой типа 0, generative grammar, rewrite grammar) называется четверка  $G = \langle N, \Sigma, P, S \rangle$ , где  $N$  и  $\Sigma$  - конечные алфавиты,  $N \cap \Sigma = \emptyset$ ,  $P \subseteq (N \cup \Sigma)^+ \times (N \cup \Sigma)^*$ ,  $P$  конечно и  $S \in N$ . Здесь  $\Sigma$  - основной алфавит (терминальный алфавит), его элементы называются терминальными символами или терминалами (terminal),  $N$  - вспомогательный алфавит (нетерминальный алфавит), его элементы называются нетерминальными символами, нетерминалами, вспомогательными символами или переменными (nonterminal, variable),  $S$  - начальный символ (аксиома, start symbol). Пары  $(\alpha, \beta) \in P$  называются правилами подстановки, просто правилами или продукциями (rewriting rule, production) и записываются в виде  $\alpha \rightarrow \beta$ .

**Пример 1.4.2.** Пусть даны множества  $N = \{S\}$ ,  $\Sigma = \{a, b, c\}$ ,  $P = \{S \rightarrow acSbcS, cS \rightarrow \varepsilon\}$ . Тогда  $\langle N, \Sigma, P, S \rangle$  является порождающей грамматикой.

**Замечание 1.4.3.** Будем обозначать элементы множества  $\Sigma$  строчными буквами из начала латинского алфавита, а элементы множества  $N$  - заглавными латинскими буквами. Обычно в примерах мы будем задавать грамматику в виде списка правил, подразумевая, что алфавит  $N$  составляют все заглавные буквы, встречающиеся в правилах, а алфавит  $\Sigma$  - все строчные буквы, встречающиеся в правилах. При этом правила порождающей грамматики записывают в таком порядке, что левая часть первого правила есть начальный символ  $S$ .

**Замечание 1.4.4.** Для обозначения  $n$  правил с одинаковыми левыми частями  $\alpha \rightarrow \beta_1, \dots, \alpha \rightarrow \beta_n$  часто используют сокращенную запись  $\alpha \rightarrow \beta_1 \mid \dots \mid \beta_n$ .

**Определение 1.4.5.** Пусть дана грамматика  $G$ . Пишем  $\phi \xRightarrow{G} \psi$ , если  $\phi = \eta\alpha\theta$ ,  $\psi = \eta\beta\theta$  и  $(\alpha \rightarrow \beta) \in P$  для некоторых слов  $\alpha, \beta, \eta, \theta$  в алфавите  $N \cup \Sigma$ . Если

$\phi \xRightarrow{G} \psi$ , то говорят, что слово  $\psi$  **непосредственно выводимо** из слова  $\phi$ .

**Замечание 1.4.6.** Когда из контекста ясно, о какой грамматике идет речь, вместо  $\xRightarrow{G}$  можно писать просто  $\Rightarrow$ .

**Пример 1.4.7.** Пусть  $G = \langle \{S\}, \{a, b, c\}, \{S \rightarrow acSbcS, cS \rightarrow \epsilon\}, S \rangle$ .

Тогда  $cSaccS \xRightarrow{G} cSa$ .

**Определение 1.4.8.** Если  $\omega_0 \xRightarrow{G} \omega_1 \xRightarrow{G} \dots \xRightarrow{G} \omega_n$ , где  $n \geq 0$ , то пишем  $\omega_0 \xRightarrow{*G} \omega_n$  и говорим, что слово  $\omega_n$  **выводимо** из слова  $\omega_0$ . Другими словами, бинарное отношение  $\xRightarrow{*G}$  является рефлексивным, транзитивным замыканием бинарного отношения  $\xRightarrow{G}$ , определенного на множестве  $(N \cup \Sigma)^*$ .

При этом последовательность слов  $\omega_0, \omega_1, \dots, \omega_n$  называется **выводом** (derivation) слова  $\omega_n$  из слова  $\omega_0$  в грамматике  $G$ . Число  $n$  называется **длиной** (**количеством шагов**) этого вывода.

**Замечание 1.4.9.** В частности, для всякого слова  $\omega \in (N \cup \Sigma)^*$  имеет место  $\omega y t^* \xRightarrow{*G} \omega$  (так как возможен вывод длины 0).

**Пример 1.4.10.** Пусть  $G = \langle \{S\}, \{a, b\}, \{S \rightarrow aSa, S \rightarrow b\}, S \rangle$ . Тогда  $aSa \xRightarrow{*G} aaaaaSaaaa$ . Длина этого вывода - 3.

**Определение 1.4.11.** Язык, **порождаемый** грамматикой  $G$ , - это множество  $L(G) = \{\omega \in \Sigma^* \mid S \xRightarrow{*G} \omega\}$ . Будем также говорить, что грамматика  $G$  **порождает** (generates) язык  $L(G)$ .

**Замечание 1.4.12.** Существенно, что в определение порождающей грамматики включены два алфавита -  $\Sigma$  и  $N$ . Это позволило нам в определении 1.4.11 "отсеять" часть слов, получаемых из начального символа. А именно, отбрасывается каждое слово, содержащее хотя бы один символ, не принадлежащий алфавиту  $\Sigma$ .

**Пример 1.4.13.** Если  $G = \langle \{S\}, \{a, b\}, \{S \rightarrow aSa, S \rightarrow bb\}, S \rangle$ , то  $L(G) = \{a^n b b a^n \mid n \geq 0\}$ .

**Определение 1.4.14.** Две грамматики **эквивалентны**, если они порождают один и тот же язык.

**Пример 1.4.15.** Грамматика  $S \rightarrow abS$ ,  
 $S \rightarrow a$   
и грамматика

$$T \rightarrow aU,$$

$$U \rightarrow baU,$$

$$U \rightarrow \varepsilon$$

эквивалентны.

**Упражнение 1.4.16.** Описать язык, порождаемый грамматикой

$$S \rightarrow FF,$$

$$F \rightarrow aFb,$$

$$F \rightarrow ab.$$

**Упражнение 1.4.17.** Описать язык, порождаемый грамматикой

$$C \rightarrow Cc,$$

$$C \rightarrow A,$$

$$A \rightarrow aAb,$$

$$A \rightarrow \varepsilon.$$

**Упражнение 1.4.18.** Описать язык, порождаемый грамматикой

$$K \rightarrow \varepsilon,$$

$$K \rightarrow a,$$

$$K \rightarrow b,$$

$$K \rightarrow aKa,$$

$$K \rightarrow bKb.$$

**Упражнение 1.4.19.** Описать язык, порождаемый грамматикой

$$D \rightarrow DA,$$

$$DAA \rightarrow ADb,$$

$$ADA \rightarrow b,$$

$$A \rightarrow a.$$

**Упражнение 1.4.20.** Описать язык, порождаемый грамматикой

$$F \rightarrow aFH,$$

$$F \rightarrow abc,$$

$$bH \rightarrow bbc,$$

$$cH \rightarrow Hc.$$

**Упражнение 1.4.21.** Описать язык, порождаемый грамматикой

$$S \rightarrow aTS,$$

$$S \rightarrow U,$$

$$Ta \rightarrow aaT,$$

$$TU \rightarrow U,$$

$$U \rightarrow a.$$

**Упражнение 1.4.22.** Эквивалентны ли грамматика

$S \rightarrow ab,$   
 $S \rightarrow aKSb,$   
 $K \rightarrow bSb,$   
 $KS \rightarrow b,$   
 $K \rightarrow \varepsilon$   
 и грамматика  
 $S \rightarrow aTb,$   
 $T \rightarrow \varepsilon,$   
 $T \rightarrow b,$   
 $T \rightarrow S,$   
 $T \rightarrow bSbS?$

**Упражнение 1.4.23.** Эквивалентны ли грамматика

$S \rightarrow aD,$   
 $D \rightarrow bba,$   
 $D \rightarrow baDa,$   
 $D \rightarrow aDaDa$   
 и грамматика  
 $S \rightarrow aaE,$   
 $S \rightarrow abJ,$   
 $E \rightarrow bJJ,$   
 $J \rightarrow aaEa,$   
 $J \rightarrow abJa,$   
 $J \rightarrow ba?$

## 1.5. Классы грамматик

**Определение 1.5.1.** Контекстной грамматикой (контекстно-зависимой грамматикой, грамматикой непосредственно составляющих, НС-грамматикой, грамматикой типа 1, context-sensitive grammar, phrase-structure grammar) называется порождающая грамматика, каждое правило которой имеет вид  $\eta A \theta \rightarrow \eta \alpha \theta$ , где  $A \in N$ ,  $\eta \in (N \cup \Sigma)^*$ ,  $\theta \in (N \cup \Sigma)^*$ ,  $\alpha \in (N \cup \Sigma)^+$ .

**Пример 1.5.2.** Грамматика

$S \rightarrow TS,$   $A \rightarrow a,$   
 $S \rightarrow US,$   $TA \rightarrow AAT,$   
 $S \rightarrow b,$   $UA b \rightarrow b,$   
 $Tb \rightarrow Ab,$   $UAAA \rightarrow AAU$

не является контекстной (последние три правила не имеют требуемого вида).

**Определение 1.5.3.** Контекстно-свободной грамматикой (бесконтекстной грамматикой, КС-грамматикой, грамматикой типа 2, context-free grammar) называется порождающая грамматика, каждое правило которой имеет вид  $A \rightarrow \alpha$ , где  $A \in N$ ,  $\alpha \in (N \cup \Sigma)^*$ .

**Пример 1.5.4.** Грамматика

$$\begin{array}{ll} S \rightarrow ASTA, & AT \rightarrow UT, \\ S \rightarrow AbA, & UT \rightarrow UV, \\ A \rightarrow a, & UV \rightarrow TV, \\ bT \rightarrow bb, & TV \rightarrow TA \end{array}$$

является контекстной, но не контекстно-свободной (последние пять правил не имеют требуемого вида).

**Определение 1.5.5.** *Линейной грамматикой* (linear grammar) называется порождающая грамматика, каждое правило которой имеет вид  $A \rightarrow u$  или  $A \rightarrow uBv$ , где  $A \in N$ ,  $u \in \Sigma^*$ ,  $v \in \Sigma^*$ ,  $B \in N$ .

Грамматика

$$\begin{array}{l} S \rightarrow TT, \\ T \rightarrow cTT, \\ T \rightarrow bT, \\ T \rightarrow a \end{array}$$

является контекстно-свободной, но не линейной (первые два правила не имеют требуемого вида).

**Определение 1.5.7.** *Праволинейной грамматикой* (рациональной грамматикой, грамматикой типа 3, right-linear grammar) называется порождающая грамматика, каждое правило которой имеет вид  $A \rightarrow u$  или  $A \rightarrow uB$ , где  $A \in N$ ,  $u \in \Sigma^*$ ,  $B \in N$ .

**Пример 1.5.8.** Грамматика

$$\begin{array}{l} S \rightarrow aSa, \\ S \rightarrow T, \\ T \rightarrow bT, \\ T \rightarrow \varepsilon \end{array}$$

является линейной, но не праволинейной (первое правило не имеет требуемого вида).

**Пример 1.5.9.** Грамматика

$$\begin{array}{l} S \rightarrow T, \\ U \rightarrow abba \end{array}$$

праволинейная. Она порождает язык  $\emptyset$ .

**Пример 1.5.10.** Грамматика

$$\begin{array}{ll} S \rightarrow aS, & T \rightarrow aT, \\ S \rightarrow bS, & T \rightarrow bT, \\ S \rightarrow aaaT, & T \rightarrow \varepsilon \\ S \rightarrow aabaT, \\ S \rightarrow abaaT, \\ S \rightarrow aabbaT, \\ S \rightarrow ababaT, \\ S \rightarrow abbbaT, \end{array}$$

праволинейная.

**Пример 1.5.11.** Грамматика

$$\begin{array}{ll} S \rightarrow \varepsilon, & T \rightarrow abaT, \\ S \rightarrow aaaS, & T \rightarrow baaT, \\ S \rightarrow abbS, & T \rightarrow bbbT, \\ S \rightarrow babS, & T \rightarrow bbaS \\ S \rightarrow aabT, & \end{array}$$

праволинейная. Обобщенный вариант языка, порождаемого этой грамматикой, используется в доказательстве разрешимости арифметики Пресбургера.

**Определение 1.5.12.** Правила вида  $\alpha \rightarrow \varepsilon$  называются  $\varepsilon$ -правилами или эпсилон-правилами.

**Лемма 1.5.13.** Каждая праволинейная грамматика является линейной. Каждая линейная грамматика является контекстно-свободной. Каждая контекстно-свободная грамматика без  $\varepsilon$ -правил является контекстной грамматикой.

**Определение 1.5.14.** Классы грамматик типа 0, 1, 2 и 3 образуют иерархию Хомского (Chomsky hierarchy).

**Определение 1.5.15.** Язык называется языком типа 0 (контекстным языком, контекстно-свободным языком, линейным языком, праволинейным языком), если он порождается некоторой грамматикой типа 0 (соответственно контекстной грамматикой, контекстно-свободной грамматикой, линейной грамматикой, праволинейной грамматикой). Контекстно-свободные языки называются также алгебраическими языками.

**Пример 1.5.16.** Пусть дан произвольный алфавит

$$\Sigma = \{a_1, \dots, a_n\}.$$

Тогда язык  $\Sigma^*$  является праволинейным, так как он порождается грамматикой

$$\begin{array}{l} S \rightarrow \varepsilon, \\ S \rightarrow a_1 S, \\ \vdots \\ S \rightarrow a_n S. \end{array}$$

**Упражнение 1.5.17.** Какому классу принадлежит грамматика

$$\begin{array}{l} S \rightarrow abba, \\ S \rightarrow baa? \end{array}$$

**Упражнение 1.5.18.** Какому классу принадлежит грамматика

$$\begin{array}{l} S \rightarrow AD, \\ A \rightarrow aA, \\ A \rightarrow \varepsilon, \\ D \rightarrow bDc, \\ D \rightarrow \varepsilon? \end{array}$$

**Упражнение 1.5.19.** Найти праволинейную грамматику, порождающую язык



$$\{a^n b^m \mid n \geq 1 \text{ или } m \geq 1\}.$$

**Упражнение 1.5.20.** Найти праволинейную грамматику, эквивалентную грамматике

$$S \rightarrow K b b a K,$$

$$K \rightarrow K a,$$

$$K \rightarrow K b,$$

$$K \rightarrow \varepsilon.$$

**Упражнение 1.5.21.** Найти праволинейную грамматику, эквивалентную грамматике

$$S \rightarrow a S b,$$

$$K \rightarrow a K,$$

$$S \rightarrow K,$$

$$J \rightarrow J b,$$

$$S \rightarrow J,$$

$$K \rightarrow a,$$

$$J \rightarrow \varepsilon.$$

## Математическая теория формальных языков

### 2. Лекция: Конечные автоматы: версия для печати и PDA

В данной лекции рассматриваются два наиболее распространенных способа конечного задания формального языка: грамматики и автоматы. Рассматриваются конечные автоматы, соответствующие в иерархии Хомского праволинейным грамматикам, определяются понятия конечного автомата, недетерминированного конечного автомата и распознаваемого конечным автоматом языка. Приведены практические примеры и упражнения для закрепления материала

Два наиболее распространенных способа конечного задания формального языка - это грамматики и автоматы. Автоматами в данном контексте называют математические модели некоторых вычислительных устройств. В этой лекции рассматриваются конечные автоматы, соответствующие в иерархии Хомского праволинейным грамматикам. Более сильные вычислительные модели будут определены позже, в лекциях [10](#), [14](#) и [15](#). Термин "автоматный язык" закреплен за языками, распознаваемыми именно конечными автоматами, а не какими-либо более широкими семействами автоматов (например, автоматами с магазинной памятью или линейно ограниченными автоматами).

В разделе 2.1 определяются понятия конечного автомата (для ясности такой автомат можно называть недетерминированным конечным автоматом) и распознаваемого конечным автоматом языка. В следующем разделе дается другое, но эквивалентное первому определение языка, распознаваемого конечным автоматом. Оно не является необходимым для дальнейшего изложения, но именно это определение поддается обобщению на случаи автоматов других типов.

В разделе 2.3 доказывается, что тот же класс автоматных языков можно получить, используя лишь конечные автоматы специального вида (они читают на каждом такте ровно один символ и имеют ровно одно начальное состояние). Во многих учебниках конечными автоматами называют именно такие автоматы.

Целую серию классических результатов теории формальных языков составляют теоремы о точном соответствии некоторых классов грамматик некоторым классам автоматов. Первая теорема из этой серии, утверждающая, что праволинейные грамматики порождают в точности автоматные языки, доказывается в разделе 2.4.

Другая серия результатов связана с возможностью сузить некоторый класс грамматик, не изменив при этом класс порождаемых ими языков. Обычно в таком случае грамматики из меньшего класса называются грамматиками в нормальной форме. В разделе 2.5\* формулируется результат такого типа для праволинейных грамматик. Сама эта теорема не представляет большого интереса, но аналогичные результаты, доказываемые позже для контекстно-свободных грамматик, используются во многих доказательствах и алгоритмах.

Не все конечные автоматы подходят для конструирования распознающих устройств, пригодных для практических приложений, так как в общем случае конечный автомат не дает точного указания, как поступать на очередном шаге, а разрешает продолжать вычислительный процесс несколькими способами. Этого недостатка нет у детерминированных конечных автоматов (частного случая недетерминированных конечных автоматов), определенных в разделе 2.6. В разделе 2.7 доказывается, что каждый автоматный язык задается некоторым детерминированным конечным автоматом.

### 2.1. Недетерминированные конечные автоматы

**Определение 2.1.1.** *Конечный автомат* (finite automaton, finite-state machine) - это

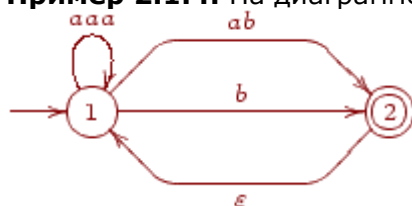
пятерка  $M = \langle Q, \Sigma, \Delta, I, F \rangle$ , где  $\Sigma$  - конечный *входной алфавит* (или просто *алфавит*) данного конечного автомата,  $Q$  и  $\Delta$  - конечные множества,  $\Delta \subseteq Q \times \Sigma^* \times Q$ ,  $I \subseteq Q$ ,  $F \subseteq Q$ . Элементы  $Q$  называются *состояниями* (state), элементы  $I$  - *начальными* (initial) состояниями, элементы  $F$  - *заключительными* или *допускающими* (final, accepting) состояниями. Если  $\langle p, x, q \rangle \in \Delta$ , то  $\langle p, x, q \rangle$  называется *переходом* (transition) из  $p$  в  $q$ , а слово  $x$  - *меткой* (label) этого перехода.

**Пример 2.1.2.** Пусть  $Q = \{1, 2\}$ ,  $\Sigma = \{a, b\}$ ,  $I = \{1\}$ ,  $F = \{2\}$ ,  $\Delta = \{\langle 1, aaa, 1 \rangle, \langle 1, ab, 2 \rangle, \langle 1, b, 2 \rangle, \langle 2, \varepsilon, 1 \rangle\}$ .

Тогда  $\langle Q, \Sigma, \Delta, I, F \rangle$  - конечный автомат.

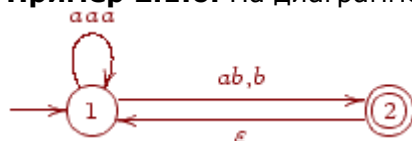
**Замечание 2.1.3.** Конечные автоматы можно изображать в виде *диаграмм состояний* (state diagram) (иногда их называют *диаграммами переходов* (transition diagram)). На диаграмме каждое состояние обозначается кружком, а переход - стрелкой. Стрелка из  $p$  в  $q$ , помеченная словом  $x$ , показывает, что  $\langle p, x, q \rangle$  является переходом данного конечного автомата. Каждое начальное состояние распознается по ведущей в него короткой стрелке. Каждое допускающее состояние отмечается на диаграмме двойным кружком.

**Пример 2.1.4.** На диаграмме изображен конечный автомат из примера 2.1.2.



**Замечание 2.1.5.** Если в конечном автомате имеются несколько переходов с общим началом и общим концом, то такие переходы называются *параллельными*. Иногда на диаграмме состояний конечного автомата параллельные переходы изображают одной стрелкой. При этом метки переходов перечисляют через запятую.

**Пример 2.1.6.** На диаграмме снова изображен конечный автомат из примера 2.1.2.



**Определение 2.1.7.** Путь (path) конечного автомата - это кортеж

$\langle q_0, e_1, q_1, e_2, \dots, q_n \rangle$ , где  $n \geq 0$  и  $e_i = \langle q_{i-1}, w_i, q_i \rangle \in \Delta$  для каждого  $i$ . При этом  $q_0$  - начало пути  $q_n$  - конец пути  $n$  - длина пути  $w_1 \dots w_n$  - метка пути.

**Замечание 2.1.8.** Для любого состояния  $q \in Q$  существует путь  $\langle q \rangle$ . Его метка -  $\varepsilon$ , начало и конец совпадают.

**Определение 2.1.9.** Путь называется *успешным* если его начало принадлежит  $I$ , а конец принадлежит  $F$ .

**Пример 2.1.10.** Рассмотрим конечный автомат из примера 2.1.2. Путь

$\langle 1, \langle 1, b, 2 \rangle, 2, \langle 2, \varepsilon, 1 \rangle, 1, \langle 1, aaa, 1 \rangle, 1, \langle 1, b, 2 \rangle, 2 \rangle$  является успешным. Его

метка - baaab. Длина этого пути - 4.

**Определение 2.1.11.** Слово  $w$  *допускается* (is accepted, is recognized) конечным автоматом  $M$ , если оно является меткой некоторого успешного пути.

**Определение 2.1.12.** Язык, *распознаваемый* конечным автоматом  $M$ , - это язык  $L(M)$ , состоящий из меток всех успешных путей (то есть из всех допускаемых данным автоматом слов). Будем также говорить, что автомат  $M$  *распознает* (recognizes, accepts) язык  $L(M)$ .

**Замечание 2.1.13.** Если  $I \cap F \neq \emptyset$ , то язык, распознаваемый конечным автоматом  $\langle Q, \Sigma, \Delta, I, F \rangle$ , содержит  $\varepsilon$ .

**Пример 2.1.14.** Пусть  $M = \langle Q, \Sigma, \Delta, I, F \rangle$ , где  $Q = \{1, 2\}$ ,  $\Sigma = \{a, b\}$ ,  $I = \{1\}$ ,  $F = \{1, 2\}$ ,

$\Delta = \{\langle 1, a, 2 \rangle, \langle 2, b, 1 \rangle\}$ .



Тогда

$L(M) = \{(ab)^n \mid n \geq 0\} \cup \{(ab)^n a \mid n \geq 0\}$ .

**Определение 2.1.15.** Два конечных автомата *эквивалентны*, если они распознают один и тот же язык.

**Определение 2.1.16.** Язык  $L$  называется *автоматным* (finite-state language), если существует конечный автомат, распознающий этот язык.

**Замечание 2.1.17.** Обычно в учебниках используется более узкое определение недетерминированного конечного автомата, но это не меняет класса автоматных языков (см. лемму 2.3.3.).

**Пример 2.1.18.** Рассмотрим однобуквенный алфавит  $\Sigma = \{a\}$ . При любых фиксированных  $k \in \mathbb{N}$  и  $m \in \mathbb{N}$  язык  $\{a^{k+mn} \mid n \in \mathbb{N}\}$  является автоматным.

**Замечание 2.1.19.** Каждый конечный язык является автоматным.

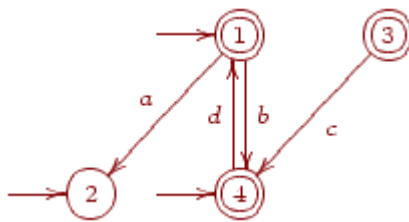
**Определение 2.1.20.** Состояние  $q$  *достижимо* (reachable) из состояния  $p$ , если существует путь, началом которого является  $p$ , а концом -  $q$ .

**Лемма 2.1.21.** Каждый автоматный язык распознается некоторым конечным автоматом, в котором каждое состояние достижимо из некоторого начального состояния и из каждого состояния достижимо хотя бы одно заключительное состояние.

**Пример 2.1.22.** Рассмотрим конечный автомат  $\langle Q, \Sigma, \Delta, I, F \rangle$ , где  $Q = \{1, 2, 3, 4\}$ ,

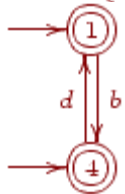
$\Sigma = \{a, b, c, d\}$ ,  $I = \{1, 2, 4\}$ ,  $F = \{1, 3, 4\}$ ,

$\Delta = \{\langle 1, a, 2 \rangle, \langle 1, b, 4 \rangle, \langle 3, c, 4 \rangle, \langle 4, d, 1 \rangle\}$ .



Удалим те состояния (и содержащие их переходы), которые не удовлетворяют требованиям леммы 2.1.21. Получается эквивалентный конечный автомат

$\langle Q', \Sigma, \Delta', I', F' \rangle$ , где  $Q' = \{1, 4\}$ ,  $I' = \{1, 4\}$ ,  $F' = \{1, 4\}$ ,  
 $\Delta' = \{\langle 1, b, 4 \rangle, \langle 4, d, 1 \rangle\}$ .



**Замечание 2.1.23** Естественным обобщением конечного автомата является *конечный преобразователь* (finite-state transducer), позволяющий на каждом такте отправить несколько символов в дополнительный "выходной" поток. В некоторых книгах конечными автоматами называют именно такие устройства (с выходным потоком).

В данном пособии конечные преобразователи рассматриваться не будут.

**Упражнение 2.1.24.** Найти конечный автомат, распознающий язык

$$\{ucv \mid u \in \{a, b\}^*, v \in \{a, b\}^*\}$$

**Упражнение 2.1.25.** Найти конечный автомат, распознающий язык

$$\{w \in \{a, b, c\}^* \mid |w|_c \neq 1\}$$

**Упражнение 2.1.26.** Найти конечный автомат, распознающий язык

$$\{a, b\}^* - (\{a^n \mid n \geq 0\} \cup \{b^n \mid n \geq 0\})$$

**Упражнение 2.1.27.** Найти конечный автомат, распознающий язык

$$\{aib \mid i \in \{a, b\}^*\} \cup \{bua \mid u \in \{a, b\}^*\}$$

**Упражнение 2.1.28.** Найти конечный автомат, распознающий язык

$$\{w \in \{a, b\}^* \mid |w|_a \geq 3\}$$

**Упражнение 2.1.28.** Найти конечный автомат, распознающий язык

$$\{w \in \{a, b\}^+ \mid f(w) \neq 4\}, \text{ где } f(\varepsilon) = 0, f(xa) = 1 - f(x) \text{ и } f(xb) = -1 - f(x) \text{ для любого } x \in \{a, b\}^*$$

## 2.2. Конфигурация конечного автомата

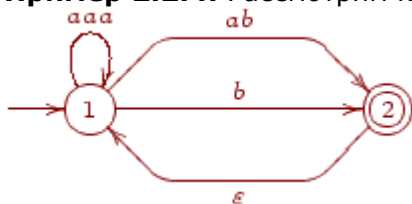
**Определение 2.2.1.** Конфигурацией или мгновенным описанием (instantaneous description) конечного автомата  $\langle Q, \Sigma, \Delta, I, F \rangle$  называется любая упорядоченная пара  $\langle q, w \rangle$ , где  $q \in Q$  и  $w \in \Sigma^*$ .

**Замечание 2.2.2.** Содержательно конфигурация представляет собой "мгновенное описание" конечного автомата. Если представить, что исходное слово, принадлежность

которого рассматриваемому языку надо проверить, дано в некотором "входном потоке", то в конфигурации  $\langle q, w \rangle$  слово  $w$  есть та часть исходного слова, которая пока осталась во входном потоке (это некоторый суффикс исходного слова), а  $q$  - текущее состояние "управляющего устройства".

**Определение 2.2.3.** Определим на множестве всех конфигураций конечного автомата  $M$  бинарное отношение  $\vdash$  (такт работы (step)) следующим образом. Если  $\langle p, x, q \rangle \in \Delta$  и  $w \in \Sigma^*$ , то  $\langle p, xw \rangle \vdash \langle q, w \rangle$ . Иногда вместо  $\vdash$  пишут  $\vdash_{\Delta}$ .

**Пример 2.2.4.** Рассмотрим конечный автомат



из примера 2.1.2. Тогда  $\langle 1, abba \rangle \vdash \langle 2, ba \rangle$ .

**Определение 2.2.5.** Бинарное отношение  $\vdash^*$  определяется как рефлексивное, транзитивное замыкание отношения  $\vdash$ .

**Пример 2.2.6.** Для конечного автомата из примера 2.1.2 выполняется

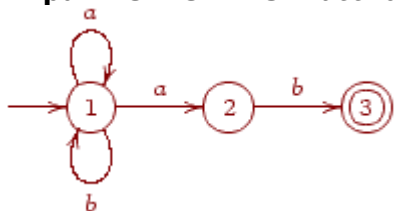
$\langle 1, aaaab \rangle \vdash^* \langle 1, aaaab \rangle$  и  $\langle 1, aaaab \rangle \vdash^* \langle 2, \varepsilon \rangle$ .

**Лемма 2.2.7.** Пусть дан конечный автомат  $M = \langle Q, \Sigma, \Delta, I, F \rangle$ . Слово  $w \in \Sigma^*$  принадлежит языку  $L(M)$  тогда и только тогда, когда для некоторых  $p \in I$  и  $q \in F$  верно  $\langle p, w \rangle \vdash^* \langle q, \varepsilon \rangle$ .

**Лемма 2.2.8.** Если  $\langle q_1, x \rangle \vdash^* \langle q_2, \varepsilon \rangle$  и  $\langle q_2, y \rangle \vdash^* \langle q_3, \varepsilon \rangle$ , то  $\langle q_1, xy \rangle \vdash^* \langle q_3, \varepsilon \rangle$ .

*Доказательство.* Лемму легко доказать индукцией по количеству тактов в вычислительном процессе, ведущем из конфигурации  $\langle q_1, x \rangle$  в конфигурацию  $\langle q_2, \varepsilon \rangle$ .

**Упражнение 2.2.9.** Рассмотрим конечный автомат.



Перечислить все конфигурации  $\langle q, w \rangle$ , удовлетворяющие условию

$\langle 1, abaa \rangle \vdash^* \langle q, w \rangle$ .

**Упражнение 2.2.10.** Существуют ли конечный автомат  $M$ , состояния  $q_1, q_2$  и слова  $x, y, z$ , такие что

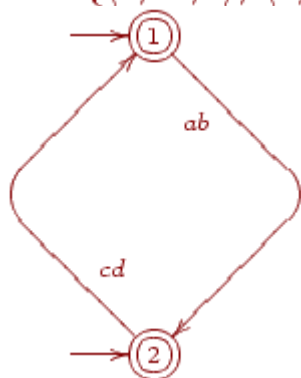
$\langle q_1, xy \rangle \vdash^* \langle q_2, y \rangle$  и  $\langle q_1, xz \rangle \not\vdash^* \langle q_2, z \rangle$ ?

**Упражнение 2.2.11.** Как связаны  $|Q|$ ,  $|\Sigma|$ ,  $|\Delta|$ ,  $|w|$  и число достижимых из  $\langle q, w \rangle$  (в смысле  $\vdash^*$ ) конфигураций?

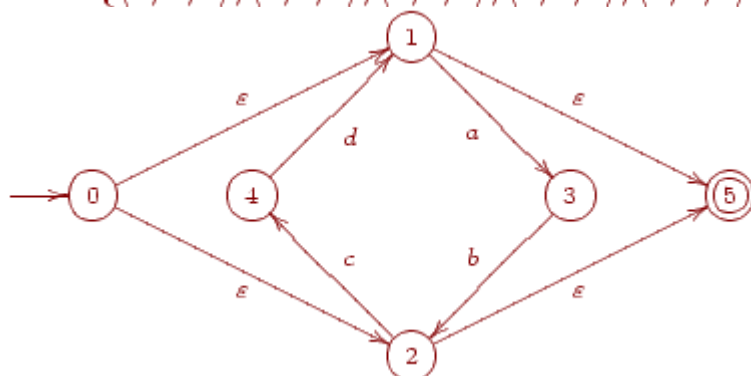
## 2.3. Конечные автоматы с однобуквенными переходами

**Лемма 2.3.1.** Каждый автоматный язык распознается некоторым конечным автоматом, не содержащим переходов с метками длины больше единицы и имеющим ровно одно начальное состояние и ровно одно заключительное состояние.

**Пример 2.3.2.** Рассмотрим язык, заданный конечным автоматом  $\langle Q, \Sigma, \Delta, I, F \rangle$ , где  $Q = \{1, 2\}$ ,  $\Sigma = \{a, b, c, d\}$ ,  $I = \{1, 2\}$ ,  $F = \{1, 2\}$ ,  $\Delta = \{\langle 1, ab, 2 \rangle, \langle 2, cd, 1 \rangle\}$ .



Тот же язык распознается конечным автоматом  $\langle Q', \Sigma, \Delta', I', F' \rangle$ , где  $Q' = \{0, 1, 2, 3, 4, 5\}$ ,  $I' = \{0\}$ ,  $F' = \{5\}$ ,  $\Delta' = \{\langle 1, a, 3 \rangle, \langle 3, b, 2 \rangle, \langle 2, c, 4 \rangle, \langle 4, d, 1 \rangle, \langle 0, \varepsilon, 1 \rangle, \langle 0, \varepsilon, 2 \rangle, \langle 1, \varepsilon, 5 \rangle, \langle 2, \varepsilon, 5 \rangle\}$ .



Здесь первые два перехода заменяют старый переход  $\langle 1, ab, 2 \rangle$  и следующие два перехода заменяют старый переход  $\langle 2, cd, 1 \rangle$ . Чтобы обеспечить единственность начального состояния, добавлены переходы  $\langle 0, \varepsilon, 1 \rangle$  и  $\langle 0, \varepsilon, 2 \rangle$ . Последние два перехода в  $\Delta'$  обеспечивают единственность заключительного состояния.

**Лемма 2.3.3.** Каждый автоматный язык распознается некоторым конечным автоматом, содержащим только переходы с метками длины единица и имеющим ровно одно начальное состояние.

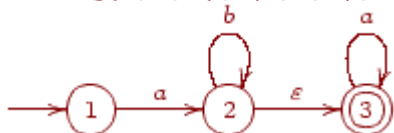
**Доказательство.** Согласно лемме 2.3.1 можно предположить, что исходный язык задан конечным автоматом  $\langle Q, \Sigma, \Delta, I, F \rangle$ , не содержащим переходов с метками длины больше единицы, причем  $|I| = 1$ . Построим искомый конечный автомат

$\langle Q', \Sigma, \Delta', I', F' \rangle$ , положив  $Q' = Q$ ,  $I' = I$ ,

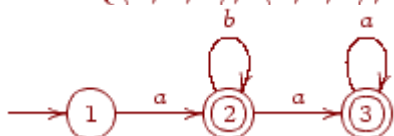


$\Delta' = \{ \langle p, a, r \rangle \mid a \in \Sigma \text{ и найдется такое } q \in Q, \text{ что } \langle q, a, r \rangle \in \Delta \text{ и существует путь из } p \text{ в } q \text{ с меткой } \varepsilon \},$   
 $F' = \{ p \in Q \mid \text{найдется такое } q \in F, \text{ что существует путь из } p \text{ в } q \text{ с меткой } \varepsilon \}.$

**Пример 2.3.4.** Пусть  $M = \langle Q, \Sigma, \Delta, I, F \rangle$ , где  $Q = \{1, 2, 3\}$ ,  $\Sigma = \{a, b\}$ ,  $I = \{1\}$ ,  $F = \{3\}$ ,  
 $\Delta = \{ \langle 1, a, 2 \rangle, \langle 2, b, 2 \rangle, \langle 2, \varepsilon, 3 \rangle, \langle 3, a, 3 \rangle \}.$



Легко убедиться, что  $L(M) = \{ab^m a^n \mid m \geq 0, n \geq 0\}$ . Тот же язык распознается конечным автоматом  $\langle Q, \Sigma, \Delta', I, F' \rangle$ , где  $F' = \{2, 3\}$  и  $\Delta' = \{ \langle 1, a, 2 \rangle, \langle 2, b, 2 \rangle, \langle 2, a, 3 \rangle, \langle 3, a, 3 \rangle \}.$



**Упражнение 2.3.5.** Найти конечный автомат с однобуквенными переходами, распознающий язык  $\{a^n b^m \mid n \geq 3, m \geq 3\}$

**Упражнение 2.3.6.** Найти конечный автомат с однобуквенными переходами, распознающий язык  $\{a^n b^m c^k d \mid n \geq 0, m \geq 0, k \geq 0\}$

**Упражнение 2.3.7.** Существуют ли автоматный язык, который не распознается никаким конечным автоматом, содержащим только переходы с метками длины единица и имеющим ровно одно начальное состояние и ровно одно заключительное состояние?

## 2.4. Характеризация праволинейных языков

**Теорема 2.4.1.** Каждый автоматный язык является праволинейным.

Без ограничения общности можно предположить, что исходный язык задан конечным автоматом  $\langle Q, \Sigma, \Delta, I, F \rangle$ , где  $Q \cap \Sigma = \emptyset$  и  $I = \{q_0\}$ . Положим  $N = Q$ ,  $S = q_0$  и  $P = \{p \rightarrow xq \mid \langle p, x, q \rangle \in \Delta\} \cup \{p \rightarrow \varepsilon \mid p \in F\}.$

**Пример 2.4.2.** Язык, распознаваемый конечным автоматом из примера 2.1.2, порождается грамматикой

$K_1 \rightarrow aaaK_1,$   
 $K_1 \rightarrow abK_2,$   
 $K_1 \rightarrow bK_2,$   
 $K_2 \rightarrow K_1,$   
 $K_2 \rightarrow \varepsilon.$

**Теорема 2.4.3.** Каждый праволинейный язык является автоматным.

*Доказательство.* Без ограничения общности можно предположить, что исходный язык задан праволинейной грамматикой, не содержащей правил вида  $A \rightarrow u$ , где  $u \in \Sigma^+$ .

Положим  $Q = N$ ,  $I = \{S\}$ ,  $F = \{A \in N \mid (A \rightarrow \varepsilon) \in P\}$  и

$$\Delta = \{\langle A, u, B \rangle \mid (A \rightarrow uB) \in P\}.$$

**Пример 2.4.4.** Пусть  $\Sigma = \{a, b\}$ . Рассмотрим грамматику

$$S \rightarrow aa,$$

$$S \rightarrow T,$$

$$T \rightarrow baT,$$

$$T \rightarrow a.$$

Она эквивалентна грамматике

$$S \rightarrow aaE,$$

$$S \rightarrow T,$$

$$T \rightarrow baT,$$

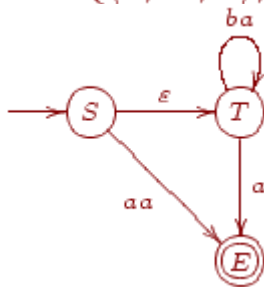
$$T \rightarrow aE,$$

$$E \rightarrow \varepsilon.$$

Язык, порождаемый этими грамматиками, распознается конечным автоматом

$\langle Q, \Sigma, \Delta, I, F \rangle$ , где  $Q = \{S, T, E\}$ ,  $I = \{S\}$ ,  $F = \{E\}$  и

$$\Delta = \{\langle S, aa, E \rangle, \langle S, \varepsilon, T \rangle, \langle T, ba, T \rangle, \langle T, a, E \rangle\}.$$



**Упражнение 2.4.5.** Найти праволинейную грамматику, порождающую язык

$$\{w \in \{a, b\}^* \mid |w|_a \geq 2, |w|_b \geq 2\}$$

**Упражнение 2.4.6.** Существует ли такая праволинейная грамматика  $G$ , что язык  $L(G)^R$  не порождается ни одной праволинейной грамматикой, имеющей столько же правил, сколько грамматика  $G$ ?

**Упражнение 2.4.7.** Существует ли такая праволинейная грамматика  $G$ , что язык  $L(G)^R$  не порождается ни одной праволинейной грамматикой с количеством правил  $n + 1$ , где  $n$  - количество правил в грамматике  $G$ ?

**Упражнение 2.4.8.** Существует ли такая праволинейная грамматика  $G$  с тремя вспомогательными символами, что язык  $L(G)^R$  не порождается ни одной праволинейной грамматикой с тремя вспомогательными символами?

## 2.5. Нормальная форма праволинейных грамматик

**Определение 2.5.1.** Праволинейная грамматика в нормальной форме (автоматная грамматика, регулярная грамматика, finite-state grammar) - это праволинейная грамматика, в которой каждое правило имеет вид  $A \rightarrow \varepsilon$ ,  $A \rightarrow a$ , или  $A \rightarrow aB$ , где

$$A \in N, B \in N, a \in \Sigma.$$

**Теорема 2.5.2.** Каждая праволинейная грамматика эквивалентна некоторой праволинейной грамматике в нормальной форме.

*Доказательство.* Применим последовательно теорему 2.4.3, лемму 2.3.3 и воспользуемся конструкцией из доказательства теоремы 2.4.1.

**Теорема 2.5.3.** Если праволинейный язык не содержит пустого слова, то он порождается некоторой праволинейной грамматикой в нормальной форме без  $\varepsilon$ -правил.

**Упражнение 2.5.4.** Найти праволинейную грамматiku, эквивалентную грамматике

$$\begin{array}{ll} S \rightarrow E, & E \rightarrow a, \\ S \rightarrow bE, & E \rightarrow bS. \\ S \rightarrow caE, & \end{array}$$

**Упражнение 2.5.5.** Найти праволинейную грамматiku в нормальной форме без  $\varepsilon$ -правил, порождающую язык

$$\{a^k b^m c^n \mid k \geq 0, m \geq 1, n \geq 0\}.$$

**Упражнение 2.5.6.** Найти праволинейную грамматiku в нормальной форме без  $\varepsilon$ -правил, порождающую язык

$$\{a, b\}^* - (\{a^n \mid n \geq 0\} \cup \{b^n \mid n \geq 0\}).$$

## 2.6. Детерминированные конечные автоматы

**Определение 2.6.1.** Конечный автомат  $\langle Q, \Sigma, \Delta, I, F \rangle$  называется *детерминированным* (deterministic), если

1. множество  $I$  содержит ровно один элемент;
2. для каждого перехода  $\langle p, x, q \rangle \in \Delta$  выполняется равенство  $|x| = 1$ ;
3. для любого символа  $a \in \Sigma$  и для любого состояния  $p \in Q$  существует не более одного состояния  $q \in Q$  со свойством  $\langle p, a, q \rangle \in \Delta$ .

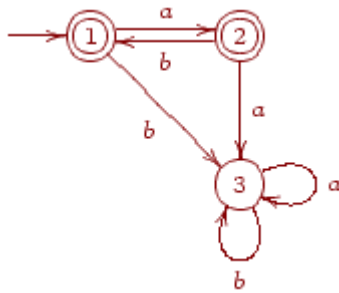
**Пример 2.6.2.** Конечный автомат из примера 2.1.14 является детерминированным.

**Определение 2.6.3.** Детерминированный конечный автомат  $M = \langle Q, \Sigma, \Delta, I, F \rangle$  называется *полным* (complete), если для каждого состояния  $p \in Q$  и для каждого символа  $a \in \Sigma$  найдется такое состояние  $q \in Q$ , что  $\langle p, a, q \rangle \in \Delta$ .

**Пример 2.6.4.** Конечный автомат из примера 2.1.14 эквивалентен полному детерминированному конечному автомату  $\langle Q, \Sigma, \Delta, I, F \rangle$ , где  $Q = \{1, 2, 3\}$ ,

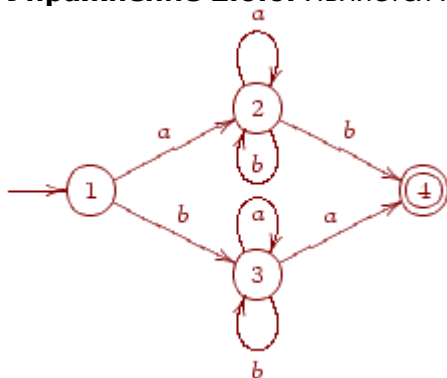
$$\Sigma = \{a, b\}, I = \{1\}, F = \{1, 2\},$$

$$\Delta = \{\langle 1, a, 2 \rangle, \langle 1, b, 3 \rangle, \langle 2, a, 3 \rangle, \langle 2, b, 1 \rangle, \langle 3, a, 3 \rangle, \langle 3, b, 3 \rangle\}.$$

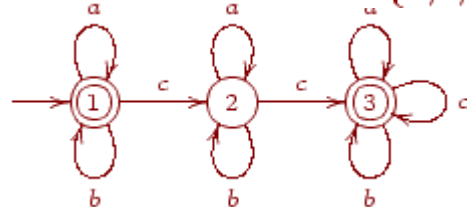


**Замечание 2.6.5.** Некоторые авторы используют в определении полного детерминированного конечного автомата вместо отношения  $\Delta$  функцию  $\delta: Q \times \Sigma \rightarrow Q$ . От функции  $\delta$  можно перейти к отношению  $\Delta$ , положив  $\Delta = \{ \langle p, a, \delta(p, a) \rangle \mid p \in Q, a \in \Sigma \}$ .

**Упражнение 2.6.6.** Является ли детерминированным следующий конечный автомат?



**Упражнение 2.6.7.** Является ли полным следующий детерминированный конечный автомат с алфавитом  $\Sigma = \{a, b, c\}$ ?



## 2.7. Преобразование конечного автомата к детерминированному виду

**Теорема 2.7.1** Каждый автоматный язык распознается некоторым полным детерминированным конечным автоматом.

*Доказательство.* Без ограничения общности можно предположить, что исходный язык задан конечным автоматом  $\langle Q, \Sigma, \Delta, I, F \rangle$ , содержащим только переходы с метками длины единица. Для любых  $a \in \Sigma$  и  $H \subseteq Q$  обозначим  $\Delta_a(H) = \{q \in Q \mid \langle p, a, q \rangle \in \Delta \text{ для некоторого } p \in H\}$ .

Обозначим через  $\mathcal{P}(Q)$  множество всех подмножеств множества  $Q$ . Построим искомый полный детерминированный конечный автомат  $\langle Q', \Sigma, \Delta', I', F' \rangle$ , положив

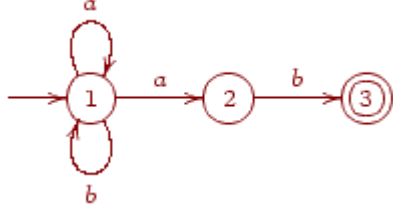
$$Q' = \mathcal{P}(Q),$$

$$\Delta' = \{ \langle H, a, \Delta_a(H) \rangle \mid H \subseteq Q, a \in \Sigma \},$$

$$I' = I \text{ и } F' = \{H \subseteq Q \mid H \cap F \neq \emptyset\}.$$

**Пример 2.7.2.** Пусть  $\Sigma = \{a, b\}$ . Рассмотрим конечный автомат

$M = \langle \{1, 2, 3\}, \Sigma, \Delta, \{1\}, \{3\} \rangle$ , где  
 $\Delta = \{ \langle 1, a, 1 \rangle, \langle 1, b, 1 \rangle, \langle 1, a, 2 \rangle, \langle 2, b, 3 \rangle \}$ .

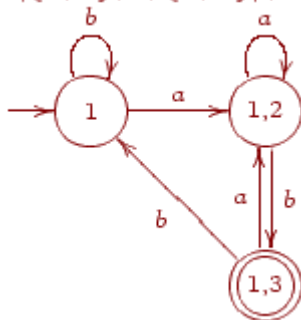


Если применить конструкцию из доказательства теоремы 2.7.1. и затем удалить состояния, не достижимые из начального состояния, то получится полный детерминированный конечный автомат

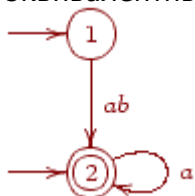
$M' = \langle \{\{1\}, \{1, 2\}, \{1, 3\}\}, \Sigma, \Delta', \{\{1\}\}, \{\{1, 3\}\} \rangle$ ,

где

$\Delta' = \{ \langle \{1\}, a, \{1, 2\} \rangle, \langle \{1\}, b, \{1\} \rangle, \langle \{1, 2\}, a, \{1, 2\} \rangle, \langle \{1, 2\}, b, \{1, 3\} \rangle, \langle \{1, 3\}, a, \{1, 2\} \rangle, \langle \{1, 3\}, b, \{1\} \rangle \}$ .



**Упражнение 2.7.3.** Найти полный детерминированный конечный автомат, эквивалентный автомату, изображенному на диаграмме.



**Упражнение 2.7.4.** Найти детерминированный конечный автомат для языка, порождаемого грамматикой

$$S \rightarrow aT,$$

$$S \rightarrow aR,$$

$$T \rightarrow bT,$$

$$T \rightarrow \varepsilon,$$

$$R \rightarrow aR,$$

$$R \rightarrow \varepsilon.$$

**Упражнение 2.7.5.** Найти детерминированный конечный автомат, распознающий язык

$$\{aub \mid u \in \{a, b\}^*\} \cup \{bua \mid u \in \{a, b\}^*\}$$

## Математическая теория формальных языков

### 3. Лекция: Основные свойства автоматных языков: версия для печати и PDA

В данной лекции доказываются свойства замкнутости класса всех автоматных языков относительно итерации, конкатенации, объединения, дополнения и пересечения, доказывается так называемая лемма о разрастании, которая во многих случаях позволяет установить неавтоматность формального языка. Приведены практические примеры и предоставлены упражнения для самостоятельного решения

Для практического применения теории конечных автоматов нужны средства, позволяющие выяснять, является ли некоторый формальный язык автоматным. Для получения положительного ответа на такой вопрос могут пригодиться достаточные условия автоматности, для отрицательного ответа - необходимые условия автоматности. В этой лекции рассматриваются наиболее часто используемые условия, касающиеся автоматности формального языка.

В первых двух разделах этой лекции доказываются свойства замкнутости класса всех автоматных языков (относительно итерации, конкатенации, объединения, дополнения, пересечения и т. д.). Эти свойства можно использовать как достаточные условия автоматности. Например, если нужно выяснить, является ли язык  $L$  автоматным, и удастся представить  $L$  в виде  $L_1 \cap L_2$ , где языки  $L_1$  и  $L_2$  автоматные, то и язык  $L$  обязательно является автоматным.

В последних двух разделах этой лекции доказывается так называемая лемма о разрастании (в англоязычной литературе pumping lemma), которая во многих случаях позволяет установить неавтоматность формального языка. К сожалению, эта лемма помогает не всегда, так как дает всего лишь необходимое условие, а не критерий автоматности.

#### 3.1. Свойства замкнутости класса автоматных языков

**Теорема 3.1.1.** Класс автоматных языков замкнут относительно итерации, конкатенации и объединения.

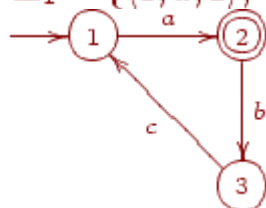
*Доказательство.* Без ограничения общности можно предположить, что каждый из исходных языков задан конечным автоматом с одним начальным и одним заключительным состоянием. Тогда во всех трех случаях результирующий автомат получается из исходных путем добавления нескольких  $\varepsilon$ -переходов и состояний и назначения новых начальных и заключительных состояний.

**Пример 3.1.2.** Пусть  $\Sigma = \{a, b, c\}$ . Рассмотрим конечный автомат

$M_1 = \langle \{1, 2, 3\}, \Sigma, \Delta_1, \{1\}, \{2\} \rangle$ ,

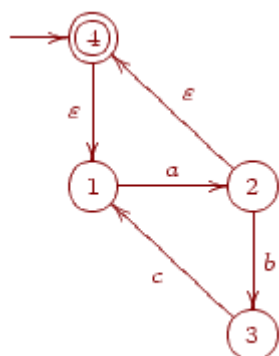
где

$\Delta_1 = \{ \langle 1, a, 2 \rangle, \langle 2, b, 3 \rangle, \langle 3, c, 1 \rangle \}$ .



Тогда язык  $L(M_1)^*$  распознается конечным автоматом

$M_2 = \langle \{1, 2, 3, 4\}, \Sigma, \Delta_1 \cup \{ \langle 4, \varepsilon, 1 \rangle, \langle 2, \varepsilon, 4 \rangle \}, \{4\}, \{4\} \rangle$ .



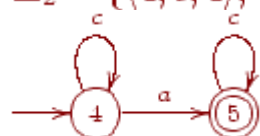
**Пример 3.1.3.** Пусть  $\Sigma = \{a, b, c\}$ . Рассмотрим конечный автомат  $M_1$  из примера 3.1.2

и конечный автомат

$$M_2 = \langle \{1, 5\}, \Sigma, \Delta_2, \{1\}, \{5\} \rangle,$$

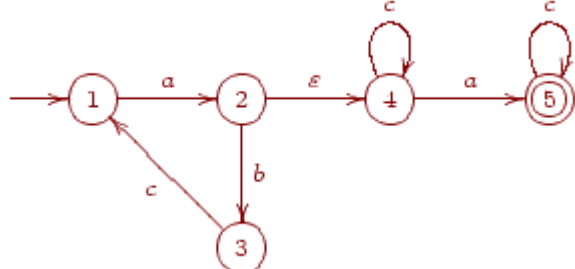
где

$$\Delta_2 = \{ \langle 1, c, 1 \rangle, \langle 1, a, 5 \rangle, \langle 5, c, 5 \rangle \}.$$



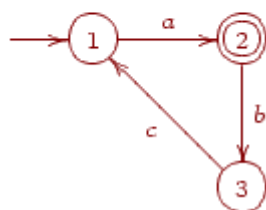
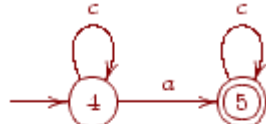
Тогда язык  $L(M_1) \cdot L(M_2)$  распознается конечным автоматом

$$M_3 = \langle \{1, 2, 3, 4, 5\}, \Sigma, \Delta_1 \cup \Delta_2 \cup \{ \langle 2, \varepsilon, 4 \rangle \}, \{1\}, \{5\} \rangle,$$



а язык  $L(M_1) \cup L(M_2)$  распознается конечным автоматом

$$M_4 = \langle \{1, 2, 3, 4, 5\}, \Sigma, \Delta_1 \cup \Delta_2, \{1, 4\}, \{2, 5\} \rangle.$$



**Упражнение 3.1.4.** Существует ли такой автоматный язык  $L$ , что язык  $L^R$  не является автоматным?

**Упражнение 3.1.5.** Существует ли такой автоматный язык  $L \subseteq \Sigma^*$ , что язык  $\text{Pref}(L)$  не является автоматным?

**Упражнение 3.1.6.** Существует ли такой автоматный язык  $L \subseteq \Sigma^*$ , что язык  $\text{Suf}(L)$



не является автоматным?

**Упражнение 3.1.7.** Существует ли такой автоматный язык  $L \subseteq \Sigma^*$ , что язык  $\text{Subw}(L)$  не является автоматным?

**Упражнение 3.1.8.** Существует ли такой автоматный язык  $L \subseteq \Sigma^*$ , что язык  $\text{Subseq}(L)$  не является автоматным?

**Упражнение 3.1.9.** Существует ли такой автоматный язык  $L$ , что язык

$$\text{Cycle}(L) = \{xy \mid yx \in L \text{ для некоторых слов } x \text{ и } y\}$$

не является автоматным?

**Упражнение 3.1.10.** Существует ли такой автоматный язык  $L$ , что язык

$$\{w \mid w \in L \text{ или } w^R \in L\}$$

не является автоматным?

**Упражнение 3.1.11.** Найти праволинейную грамматику, порождающую язык

$$\{w \in \{a, b\}^* \mid |w|_a \geq 2\} \cdot \{w \in \{c, d\}^* \mid |w|_c \geq 2\}.$$

**Упражнение 3.1.12.** Найти праволинейную грамматику, порождающую язык  $L^*$ , если язык  $L$  порождается грамматикой

$$S \rightarrow aT,$$

$$T \rightarrow bS,$$

$$T \rightarrow aR,$$

$$R \rightarrow cT,$$

$$R \rightarrow \varepsilon.$$

## 3.2. Пересечение и дополнение автоматных языков

**Теорема 3.2.1.** Класс автоматных языков замкнут относительно дополнения и пересечения.

*Доказательство.* Если язык  $L$  распознается полным детерминированным конечным

автоматом  $\langle Q, \Sigma, \Delta, I, F \rangle$ , то язык  $\Sigma^* - L$  распознается конечным автоматом

$$\langle Q, \Sigma, \Delta, I, Q - F \rangle.$$

Пересечение выражается через объединение и дополнение (закон де Моргана).

**Замечание 3.2.2.** Автоматность пересечения двух автоматных языков можно легко доказать и без привлечения теоремы 2.7.1. Для этого достаточно построить по двум конечным автоматам с однобуквенными переходами

$$M_1 = \langle Q_1, \Sigma, \Delta_1, I_1, F_1 \rangle, \quad M_2 = \langle Q_2, \Sigma, \Delta_2, I_2, F_2 \rangle$$

новый конечный автомат

$$M = \langle Q_1 \times Q_2, \Sigma, \Delta, I_1 \times I_2, F_1 \times F_2 \rangle,$$

где

$$\Delta = \{ \langle \langle p_1, p_2 \rangle, a, \langle q_1, q_2 \rangle \rangle \mid \langle p_1, a, q_1 \rangle \in \Delta_1, \langle p_2, a, q_2 \rangle \in \Delta_2 \}.$$

**Упражнение 3.2.3.** Обозначим через  $L$  язык, порождаемый грамматикой

$S \rightarrow aS,$	$T \rightarrow aT,$
$S \rightarrow bS,$	$T \rightarrow bT,$
$S \rightarrow abaaT,$	$T \rightarrow \varepsilon.$
$S \rightarrow babbT,$	

Найти праволинейную грамматику, порождающую язык  $\{a, b\}^* - L$ .

**Упражнение 3.2.4.** Обозначим через  $L$  язык, порождаемый грамматикой

$S \rightarrow F,$	$M \rightarrow aM,$	$F \rightarrow bF$
$S \rightarrow aaM,$	$M \rightarrow bM,$	$F \rightarrow abF$
$S \rightarrow T,$	$M \rightarrow \varepsilon,$	$F \rightarrow baF$
	$T \rightarrow aT,$	$F \rightarrow abaF$
	$T \rightarrow bT,$	$F \rightarrow aaaF$
	$T \rightarrow aa,$	$F \rightarrow \varepsilon.$

Найти праволинейную грамматику, порождающую язык  $\{a, b\}^* - L$ .

**Упражнение 3.2.5.** Обозначим через  $L$  язык, порождаемый грамматикой

$S \rightarrow aT,$	$T \rightarrow aT,$	$C \rightarrow aC,$
$S \rightarrow bC,$	$T \rightarrow bT,$	$C \rightarrow bC,$
	$T \rightarrow b,$	$C \rightarrow a.$

Найти праволинейную грамматику, порождающую язык  $\{a, b\}^* - L$ .

**Упражнение 3.2.6.** Существуют ли такие детерминированные конечные автоматы  $M_1$  и

$M_2$ , что язык  $L(M_1) \cup L(M_2)$  не порождается ни одним детерминированным конечным автоматом с количеством состояний  $n_1 n_2 + n_1 + n_2$ , где  $n_1$  - количество состояний автомата  $M_1$  и  $n_2$  - количество состояний автомата  $M_2$ ?

### 3.3. Лемма о разрастании для автоматных языков

**Лемма 3.3.1 (pumping lemma, лемма о разрастании, лемма о накачке, лемма-насос).** Пусть  $L$  автоматный язык над алфавитом  $\Sigma$ . Тогда найдется такое положительное целое число  $p$ , что для любого слова  $w \in L$  длины не меньше  $p$  можно подобрать слова  $x, y, z \in \Sigma^*$ , для которых верно  $x y z = w$ ,  $y \neq \varepsilon$ ,  $|xy| \leq p$  и  $x y^i z \in L$  для всех  $i \geq 0$ .

**Доказательство.** Пусть язык  $L$  распознается конечным автоматом  $\langle Q, \Sigma, \Delta, I, F \rangle$ , содержащим только переходы с метками длины единица. Положим  $p = |Q|$ . Пусть слово  $w$  является меткой успешного пути

$\langle q_0, e_1, q_1, e_2, \dots, q_n \rangle$

и  $|w| = n \geq p$ . Согласно принципу Дирихле найдутся такие индексы  $j$  и  $k$ , что

$0 \leq j < k \leq p$  и  $q_j = q_k$  (ведь множество индексов  $\{0, 1, \dots, p\}$  содержит  $p+1$  натуральных чисел, а значения  $q_i$  берутся из множества, содержащего всего  $p$  элементов). Выберем слова  $x, y$  и  $z$  так, что  $|x| = j$ ,  $|y| = k - j$  и  $x y z = w$ .

**Пример 3.3.2.** Пусть  $\Sigma = \{a, b\}$ . Рассмотрим автоматный язык  $L = \{(ab)^n \mid n \geq 0\} \cup \{a(ab)^n \mid n \geq 0\}$ .

Положим  $p = 3$ . Тогда для любого слова  $w \in L$  длины не меньше  $p$  найдутся слова  $x, y, z \in \Sigma^*$ , соответствующие утверждению леммы 3.3.1. Действительно, если  $w = abu$  для некоторого слова  $u$ , то положим  $x = \varepsilon$ ,  $y = ab$ ,  $z = u$ ; иначе  $w = aabu$  и можно положить  $x = a$ ,  $y = ab$ ,  $z = u$ .

**Упражнение 3.3.3.** Является ли автоматным язык  $\{a^m b a^m \mid m \geq 0\}$ ?

**Упражнение 3.3.4.** Является ли автоматным язык  $\{a^n \mid \text{существует такое число } p \geq n, \text{ что } p \text{ простое и } p + 2 \text{ простое}\}$ ?

**Упражнение 3.3.5.** Является ли автоматным язык  $\{a^k b^m a^n \mid k \neq n \text{ или } m = 0\}$ ?

**Упражнение 3.3.6.** Является ли автоматным язык  $\{(aab)^n a (aba)^n \mid n \geq 0\}$ ?

**Упражнение 3.3.7.** Является ли автоматным язык  $\{uaav \mid u \in \{a, b\}^*, v \in \{a, b\}^*, |u|_b \geq |v|_a\}$ ?

**Упражнение 3.3.8.** Является ли автоматным язык  $\{uav \mid u \in \{a, b\}^*, v \in \{a, b\}^*, |u|_b \geq |v|_a\}$ ?

**Упражнение 3.3.9.** Является ли автоматным язык  $\{a^k w b^k \mid k \geq 0, w \in \{a, b\}^*, |w|_a \geq 3\}$ ?

**Упражнение 3.3.10.** Является ли автоматным язык, порождаемый грамматикой  $S \rightarrow aSa$ ,  
 $S \rightarrow bSa$ ,  
 $S \rightarrow bSb$ ,  
 $S \rightarrow \varepsilon$ ?

### 3.4. Примеры неавтономных языков

**Пример 3.4.1.** Рассмотрим язык  $L = \{ab^n a^n \mid n \geq 0\}$  над алфавитом  $\Sigma = \{a, b\}$ . Утверждение леммы 3.3.1 не выполняется ни для какого натурального числа  $p$ .

Действительно, если  $w = ab^p a^p$ , то  $x = ab^k$ ,  $y = b^m$ ,  $z = b^{p-k-m} a^p$  для некоторых  $k \geq 0$  и  $m \geq 1$  или  $x = \varepsilon$ ,  $y = ab^l$ ,  $z = b^{p-l} a^p$  для некоторого  $l \geq 0$ . В обоих случаях  $x y z \notin L$ . Таким образом, язык  $L$  не является автоматным.

**Упражнение 3.4.2.** Пусть  $\Sigma = \{a, b, c\}$ . При каких словах  $u \in \{a, b\}^*$  и  $v \in \{a, b\}^*$  язык  $\{u^m c v^m \mid m > 0\}$  является автоматным?

**Замечание 3.4.3.** Условие, сформулированное в лемме 3.3.1, является необходимым для автоматности, но не достаточным.

**Пример 3.4.4.** Пусть  $\Sigma = \{a, b\}$ . Рассмотрим язык  $L = \{a^k b^m a^n \mid k=0 \text{ или } m=n\}$ . Положим  $p = 1$ . Тогда для любого слова  $w \in L$  длины не меньше  $p$  найдутся слова  $x, y, z \in \Sigma^*$ , соответствующие утверждению леммы 3.3.1. Тем не менее язык  $L$  не является автоматным, так как

$$L \cap \{ab^m a^n \mid m \geq 0, n \geq 0\} = \{ab^n a^n \mid n \geq 0\}.$$

**Лемма 3.4.5\*.** Пусть  $L$  - автоматный язык над алфавитом  $\Sigma$ . Тогда найдется такое положительное целое число  $p$ , что для любого слова  $w \in L$  можно подобрать слова  $x, y, z \in \Sigma^*$ , для которых верно  $xyz = w$ ,  $|y| \geq \lceil |w|/p \rceil$  и  $xy^i z \in L$  для всех  $i \geq 0$ . Здесь  $\lceil m \rceil$  означает целую часть числа  $m$ .

*Доказательство.* Пусть  $L$  распознается конечным автоматом  $\langle Q, \Sigma, \Delta, I, F \rangle$ , содержащим только переходы с метками длины единица. Положим  $p = |Q|$ . Пусть слово  $w$  является меткой успешного пути  $\langle q_0, e_1, q_1, e_2, \dots, q_n \rangle$ . Обозначим  $l = \lceil |w|/p \rceil$ . Если  $l = 0$ , то положим  $x = \varepsilon$  и  $y = \varepsilon$ . Пусть  $l > 0$ . Согласно принципу Дирихле найдутся такие натуральные числа  $j$  и  $k$ , что  $0 \leq j < k \leq p$  и  $q_{j-1} = q_{k-1}$ . Выберем слова  $x, y$  и  $z$  так, что  $|x| = j-1$ ,  $|y| = k-1-j+1$  и  $xyz = w$ .

**Упражнение 3.4.6.** Является ли автоматным язык  $\{a^m b^n \mid m \neq 1 \text{ или } n \text{ простое}\}$ ?

**Упражнение 3.4.7.** Является ли автоматным язык  $\{u^R v \mid u \in \{a, b\}^+, v \in \{a, b\}^*\}$ ?

**Упражнение 3.4.8.** Является ли автоматным язык

$\{w \in \{a, b\}^* \mid \text{множества } \{x \in \{a, b\}^* \mid xaab \sqsubset w\} \text{ и } \{x \in \{a, b\}^* \mid xbba \sqsubset w\} \text{ равномощны}\}$ ?

**Упражнение 3.4.9.** Является ли автоматным язык

$\{w \in \{a, b\}^* \mid \text{множества } \{x \in \{a, b\}^* \mid xaab \sqsubset w\} \text{ и } \{x \in \{a, b\}^* \mid xba \sqsubset w\} \text{ равномощны}\}$ ?

**Упражнение 3.4.10.** Является ли автоматным язык  $\{a^k b^n a^n \mid k \geq 1, n \geq 0\} \cup \{b^k a b^m a^n \mid k \geq 0, m \neq n\}$ ?

**Упражнение 3.4.11.** Является ли автоматным язык, порождаемый грамматикой

$$F \rightarrow abb,$$

$$F \rightarrow abbbFFF,$$

$$F \rightarrow abbbbFFF?$$

## Математическая теория формальных языков

### 4. Лекция: Дополнительные свойства автоматных языков: версия для печати и PDA

В данной лекции доказывается замкнутость класса всех автоматных языков относительно взятия гомоморфного образа и относительно взятия полного гомоморфного прообраза, определяются понятия побуквенного гомоморфизма и локального языка. Приведены практические примеры и предоставлены упражнения для самостоятельного решения

Эта лекция содержит дополнительные результаты, не используемые в дальнейшем изложении. В начале лекции доказывается замкнутость класса всех автоматных языков относительно взятия гомоморфного образа и относительно взятия полного гомоморфного прообраза.

В разделе 4.2\* определяются понятия побуквенного гомоморфизма и локального языка и доказывается еще один критерий автоматности: среди языков, не содержащих пустого слова, автоматными являются в точности образы локальных языков при побуквенных гомоморфизмах.

В последнем разделе этой лекции устанавливается числовой критерий автоматности для языков над однобуквенным алфавитом (в терминах арифметических прогрессий) и доказывается связанное с длинами слов необходимое условие автоматности (для произвольного алфавита).

#### 4.1. Гомоморфизмы и автоматные языки

**Теорема 4.1.1.** Для любого гомоморфизма  $h: \Sigma_1^* \rightarrow \Sigma_2^*$  и автоматного языка  $L \subseteq \Sigma_1^*$  язык  $h(L)$  является автоматным.

*Доказательство.* Пусть исходный язык  $L$  задан конечным автоматом

$M = \langle Q, \Sigma_1, \Delta, I, F \rangle$ . Положим  
 $\Delta' = \{ \langle p, h(x), q \rangle \mid \langle p, x, q \rangle \in \Delta \}$ .

Тогда язык  $h(L)$  распознается конечным автоматом  $\langle Q, \Sigma_2, \Delta', I, F \rangle$ .

**Теорема 4.1.2.** Для любого гомоморфизма  $h: \Sigma_1^* \rightarrow \Sigma_2^*$  и автоматного языка  $L \subseteq \Sigma_2^*$  язык  $h^{-1}(L)$  является автоматным.

*Доказательство.* Без ограничения общности можно предположить, что исходный язык  $L$  задан конечным автоматом  $M = \langle Q, \Sigma_2, \Delta, I, F \rangle$ , где  $\Delta$  не содержит переходов с метками длины больше единицы. Положим

$\Delta' = \{ \langle p, a, q \rangle \mid a \in \Sigma_1 \text{ и существует путь из } p \text{ в } q \text{ с меткой } h(a) \}$ .

Язык  $h^{-1}(L)$  распознается конечным автоматом  $\langle Q, \Sigma_1, \Delta', I, F \rangle$ .

**Упражнение 4.1.3.** Существует ли такой автоматный язык  $L$  над алфавитом  $\{a, b\}$ , что язык  $\{a^{|w|_b} \mid w \in L\}$  не является автоматным?

**Упражнение 4.1.4.** Существует ли такой автоматный язык  $L$  над алфавитом  $\{a, b\}$ , что язык  $\{w \in \{a, b\}^* \mid a^{|w|_a} \in L\}$  не является автоматным?

**Упражнение 4.1.5.** Существует ли такой автоматный язык  $L$  над алфавитом  $\{a, b\}$ , что

язык  $\{w \in \{a, b\}^* \mid (\exists x \in L) |x| = |w|\}$  не является автоматным?

## 4.2\*. Локальные языки

**Определение 4.2.1.** Гомоморфизм  $h: \Sigma_1^* \rightarrow \Sigma_2^*$  называется *побуквенным* (length-preserving), если  $|h(a)| = 1$  для каждого  $a \in \Sigma_1$ .

**Замечание 4.2.2.** Гомоморфизм  $h: \Sigma_1^* \rightarrow \Sigma_2^*$  является побуквенным тогда и только тогда, когда  $|h(w)| = |w|$  для каждого слова  $w \in \Sigma_1^*$ .

**Определение 4.2.3.** Язык  $L \subseteq \Sigma^*$  называется *локальным*, если существуют такие языки  $L_1 \subseteq \Sigma^*$ ,  $L_2 \subseteq \Sigma^*$ ,  $L_3 \subseteq \Sigma^*$ , что

1. языки  $L_1$  и  $L_2$  содержат только однобуквенные слова;
2. язык  $L_3$  содержит только двухбуквенные слова;
3.  $L = ((L_1 \cdot \Sigma^*) \cap (\Sigma^* \cdot L_2)) - (\Sigma^* \cdot L_3 \cdot \Sigma^*)$ .

**Лемма 4.2.4.** Каждый локальный язык является автоматным.

Очевидно, что языки  $L_1$ ,  $L_2$  и  $L_3$  в определении 4.2.3 являются конечными. Остается применить замечание 2.1.19 и теоремы 3.1.1 и 3.2.1 (напомним, что разность языков выражается через пересечение и дополнение).

**Теорема 4.2.5.** Пусть  $L$  - язык над алфавитом  $\Sigma$  и  $L$  не содержит пустого слова. Язык  $L$  является автоматным тогда и только тогда, когда существуют такие алфавит  $\Sigma_0$ , локальный язык  $L_0 \subseteq \Sigma_0^*$  и побуквенный гомоморфизм  $h: \Sigma_0^* \rightarrow \Sigma^*$ , что  $L = h(L_0)$ .

*Доказательство.* Достаточность следует из леммы 4.2.4 и теоремы 4.1.1.

Для доказательства необходимости рассмотрим конечный автомат  $\langle Q, \Sigma, \Delta, I, F \rangle$  с однобуквенными переходами, задающий язык  $L$ . В качестве алфавита  $\Sigma_0$  возьмем множество  $\Delta$ . Положим

$$L_1 = \{\langle p, a, q \rangle \mid \langle p, a, q \rangle \in \Delta, p \in I\},$$

$$L_2 = \{\langle p, a, q \rangle \mid \langle p, a, q \rangle \in \Delta, q \in F\},$$

$$L_3 = \{\langle p_1, a_1, q_1 \rangle \langle p_2, a_2, q_2 \rangle \mid \langle p_1, a_1, q_1 \rangle \in \Delta, \langle p_2, a_2, q_2 \rangle \in \Delta, q_1 \neq p_2\}$$

и  $h(\langle p, a, q \rangle) = a$  для каждого  $\langle p, a, q \rangle \in \Delta$ .

**Пример 4.2.6.** Пусть  $\Sigma = \{a, b, c\}$ . Рассмотрим конечный автомат  $M_2$  из примера 3.1.3 и обозначим  $L = L(M_2)$ . Применим конструкцию из доказательства теоремы 4.2.5 к

языку  $L$ . Для удобства заменим  $\langle 4, c, 4 \rangle$  на  $d$ ,  $\langle 4, a, 5 \rangle$  на  $e$  и  $\langle 5, c, 5 \rangle$  на  $f$ . Получим алфавит  $\Sigma_0 = \{d, e, f\}$  и локальный язык

$$L_0 = ((\{d, e\} \cdot \Sigma_0^*) \cap (\Sigma_0^* \cdot \{e, f\})) - (\Sigma_0^* \cdot \{df, ed, ee, fd, fe\} \cdot \Sigma_0^*).$$

Можно доказать, что

$$L_0 = \{d^m e f^n \mid m \geq 0, n \geq 0\}.$$

Побуквенный гомоморфизм  $h$  задается равенствами  $h(d) = c$ ,  $h(e) = a$  и  $h(f) = c$ .

Легко проверить, что действительно  $L = h(L_0)$ .

**Упражнение 4.2.7.** Пусть  $\Sigma = \{a, b, c, d\}$ . Существует ли такой побуквенный гомоморфизм  $h: \Sigma^* \rightarrow \Sigma^*$ , что  $h(abc) = bac$  и  $h(da) = da$ ?

**Упражнение 4.2.8.** Является ли локальным язык

$$\{(ab)^n \mid n \geq 1\}$$

над алфавитом  $\Sigma = \{a, b\}$ ?

**Упражнение 4.2.9.** Является ли локальным язык

$$\{aau \mid u \in \{a, b\}^*\}$$

над алфавитом  $\Sigma = \{a, b\}$ ?

**Упражнение 4.2.10.** Является ли локальным язык

$$\{bua \mid u \in \{a, b, c\}^*\}$$

над алфавитом  $\Sigma = \{a, b, c\}$ ?

### 4.3. Длины слов в автоматных языках

**Определение 4.3.1.** Пусть  $A \subseteq \mathbb{N}$ ,  $m \in \mathbb{N}$  и  $m > 0$ . Множество  $A$  называется *заключительно периодическим* (ultimately periodic) с периодом  $m$ , если выполнено условие

$$(\exists n_0 \in \mathbb{N}) (\forall n \geq n_0) (n \in A \leftrightarrow n + m \in A).$$

**Лемма 4.3.2.** Пусть  $A \subseteq \mathbb{N}$ . Тогда равносильны следующие утверждения:

1. множество  $A$  является *заключительно периодическим*;
2. найдутся такие положительное целое число  $m$  и конечные множества  $M \subseteq \mathbb{N}$  и  $K \subseteq \{0, 1, \dots, m-1\}$ , что  $A = \{k \in \mathbb{N} \mid (k \bmod m) \in K\} - M$ ;
3. множество  $A$  является объединением конечного семейства арифметических прогрессий.

**Теорема 4.3.3.** Язык  $L$  над однобуквенным алфавитом  $\{a\}$  является автоматным тогда и только тогда, когда множество  $\{k \in \mathbb{N} \mid a^k \in L\}$  является *заключительно периодическим*.

*Доказательство.* Для доказательства необходимости достаточно рассмотреть детерминированный конечный автомат, распознающий язык  $L$ .

**Теорема 4.3.4.** Если язык  $L$  является автоматным, то множество  $\{|w| \mid w \in L\}$  является *заключительно периодическим*.

*Доказательство.* Рассмотрим конечный автомат, распознающий язык  $L$ . Заменяем все символы в метках переходов на символ  $a$ . Осталось применить теорему 4.3.3 к полученному автоматному языку над однобуквенным алфавитом  $\{a\}$ .

**Упражнение 4.3.5.** Существует ли такой автоматный язык  $L$  над алфавитом  $\{a\}$ , что



язык  $\{a^n \mid a^{n^2} \in L\}$  не является автоматным?

**Упражнение 4.3.6.** Существует ли такой автоматный язык  $L$  над алфавитом  $\{a\}$ , что язык  $\{a^n \mid a^{2^n} \in L\}$  не является автоматным?

**Упражнение 4.3.7.** Существует ли такой автоматный язык  $L_1$  над алфавитом  $\Sigma$ , что язык

$L_2 = \{w \in \Sigma^* \mid wx \in L_1, |x| = 2^{|w|} \text{ для некоторого слова } x \in \Sigma^*\}$   
не является автоматным?

**Упражнение 4.3.8.** Существует ли такой автоматный язык  $L$  над алфавитом  $\{a, b\}$ , что язык  $\{a^{2^n} \mid a^n \notin L\}$  не является автоматным?

**Упражнение 4.3.9.** Существует ли такой автоматный язык  $L$  над алфавитом  $\{a, b\}$ , что язык  $\{a^n \mid a^{n^3} \notin L\}$  не является автоматным?

## Математическая теория формальных языков

### 5. Лекция: Регулярные выражения: версия для печати и PDA

В данной лекции рассматривается наиболее удобный и компактный способ конечного описания формального языка - регулярные выражения, - который находит практическое применение во многих компьютерных приложениях, таких как текстовые редакторы, интерпретаторы командной строки и автоматические генераторы лексических анализаторов. Приведены практические примеры и предоставлены упражнения для самостоятельного решения

До сих пор мы рассматривали два способа конечного описания формального языка: грамматики и автоматы. Третий способ, часто наиболее удобный и компактный, - регулярные выражения. В них используются символы, обозначающие итерацию, конкатенацию и объединение языков. Например, для обозначения итерации традиционно используется символ "звездочка".

Регулярные выражения находят практическое применение во многих компьютерных приложениях, таких как текстовые редакторы и интерпретаторы командной строки. В автоматических генераторах лексических анализаторов принято использовать именно регулярные выражения в качестве формализма, с помощью которого задаются классы однотипных лексем (например, класс идентификаторов, класс десятичных констант). В языках программирования Perl, Python и др. имеется встроенная поддержка регулярных выражений.

В начале лекции даются необходимые определения. Затем в разделе 5.2\* приводятся некоторые тождества регулярных выражений, такие как ассоциативность конкатенации, дистрибутивность конкатенации относительно объединения и др. В разделе 5.3 доказывается, что регулярные выражения задают в точности класс автоматных языков. В разделе 5.4\* формулируется теорема о классификации автоматных языков по их сложности, измеряемой звездной высотой (необходимой глубиной вложенности итераций при задании данного языка регулярным выражением).

### 5.1. Определение регулярного выражения

**Определение 5.1.1.** Регулярное выражение над алфавитом  $\Sigma$  определяется рекурсивно следующим образом:  $\emptyset$  является регулярным выражением;  $1$  является регулярным выражением; если  $a \in \Sigma$ , то  $a$  является регулярным выражением; если  $e$  и  $f$  являются регулярными выражениями, то  $(e+f)$ ,  $(e \cdot f)$  и  $e^*$  тоже являются регулярными выражениями.

Для экономии скобок будем считать, что операция  $*$  связывает сильнее (то есть имеет более высокий приоритет), чем умножение, а умножение связывает сильнее, чем сложение. Вместо  $e \cdot f$  часто пишут просто  $ef$ .

**Пример 5.1.2.** Пусть  $\Sigma = \{a, b\}$ . Тогда  $((a \cdot b)^* \cdot (1+a))$  является регулярным выражением над алфавитом  $\Sigma$ .

**Определение 5.1.3.** Каждое регулярное выражение  $e$  над алфавитом  $\Sigma$  задает (denotes, represents) некоторый язык над алфавитом  $\Sigma$  (обозначение  $L(e)$ ), определяемое рекурсивно следующим образом:

$$\begin{aligned}
L(a) &= \{a\}, \text{ если } a \in \Sigma, \\
L(0) &= \emptyset, \\
L(1) &= \{\varepsilon\}, \\
L(e+f) &= L(e) \cup L(f), \\
L(e \cdot f) &= L(e) \cdot L(f), \\
L(e^*) &= L(e)^*.
\end{aligned}$$

Заметим, что в правой части последнего выражения символом  $*$  обозначена итерация языка (см. определение 1.2.7).

Вместо  $L(e)$  часто пишут просто  $e$ .

**Пример 5.1.4.** Пусть  $\Sigma = \{a, b\}$ . Согласно определению  $L((ab)^* \cdot (1+a)) = \{(ab)^n \mid n \geq 0\} \cup \{(ab)^n a \mid n \geq 0\}$ .

**Определение 5.1.5.** Язык  $L$  называется *регулярным* если он задается некоторым регулярным выражением.

**Определение 5.1.6.** Пусть  $e$  - регулярное выражение. Тогда  $e^+ = e^*e$ .

**Упражнение 5.1.7.** Упростить регулярное выражение  $((a+bc)^*)^*$

**Упражнение 5.1.8.** Найти праволинейную грамматику для языка  $ab^*a$

**Упражнение 5.1.9.** Найти праволинейную грамматику для языка  $((a+b)a)^*$ .

**Упражнение 5.1.10.** Найти полный детерминированный конечный автомат для языка  $(a+b)^*(aab+abaa+abb)(a+b)^*$ .

**Упражнение 5.1.11.** Найти полный детерминированный конечный автомат для языка  $(a+b)^*(a(b+1)abb+baa)(a+b)^*$ .

**Упражнение 5.1.12.** Найти полный детерминированный конечный автомат для языка  $(b+c)((ab)^*c+(ba)^*)^*$ .

**Упражнение 5.1.13.** Найти полный детерминированный конечный автомат для языка  $(abab)^+(aba)^*$ .

**Упражнение 5.1.14.** Найти полный детерминированный конечный автомат для языка  $(c+(a+b+c)(b+a(b+c)^*a))^*(a+b+c)$ .

## 5.2\*. Свойства регулярных выражений

**Лемма 5.2.1.** Регулярные выражения образуют ассоциативное полукольцо с операциями  $(0, +, 1, \cdot)$ , то есть для любых регулярных выражений  $e, f$  и  $g$  выполняются следующие тождества:

1.  $e+f = f+e$ ;
2.  $e+0 = e$ ;
3.  $(e+f)+g = e+(f+g)$ ;

4.  $e \cdot 1 = e$ ;
5.  $1 \cdot e = e$ ;
6.  $(e \cdot f) \cdot g = e \cdot (f \cdot g)$ ;
7.  $e \cdot (f + g) = e \cdot f + e \cdot g$ ;
8.  $(f + g) \cdot e = f \cdot e + g \cdot e$ ;
9.  $e \cdot 0 = 0$ ;
10.  $0 \cdot e = 0$ .

Равенство понимается как равенство языков, задаваемых регулярными выражениями.

**Лемма 5.2.2.** Для любых регулярных выражений  $e$  и  $f$  выполняются следующие тождества:

1.  $e + e = e$ ;
2.  $(1 + e + ee + \dots + e^{n-1})(e^n)^* = e^*$  для любого  $n \geq 1$ ;
3.  $(e^* f)^* e^* = (e + f)^*$ ;
4.  $1 + e(fe)^* f = (ef)^*$ .

**Лемма 5.2.3.** Для любых регулярных выражений  $e$ ,  $f$  и  $g$ , если  $e = ef + g$  и  $\varepsilon \notin L(f)$ , то  $e = gf^*$ .

**Упражнение 5.2.4.** Упростить регулярное выражение  $0^*$ .

**Упражнение 5.2.5.** Упростить регулярное выражение  $(a + b + ab)^*$ .

**Упражнение 5.2.6.** Упростить регулярное выражение  $(a^* b)^* + (b^* a)^*$ .

**Упражнение 5.2.7.** Упростить регулярное выражение  $((b^* a)^* b + 1)b^*$ .

**Упражнение 5.2.8.** Упростить регулярное выражение  $((ab + aab)^* a^*)^*$ .

**Упражнение 5.2.9.** Упростить регулярное выражение  $(abbaab + abbaaba)^*$ .

**Упражнение 5.2.10.** Упростить регулярное выражение  $(a + b)^* (a(a + b)^* a + b(a + b)^* b)$ .

**Упражнение 5.2.11.** Упростить регулярное выражение  $(eb^*(1 + c(d + ab^*c)^*a)b^*f)^* eb^*c(d + ab^*c)^*$ .

## 5.3. Теорема Клини

**Определение 5.3.1.** Назовем *обобщенным конечным автоматом* аналог конечного автомата, где переходы помечены не словами, а регулярными выражениями. *Метка пути* такого автомата - произведение регулярных выражений на переходах данного пути. Слово  $w$  *допускается* обобщенным конечным автоматом, если оно принадлежит языку, задаваемому меткой некоторого успешного пути.

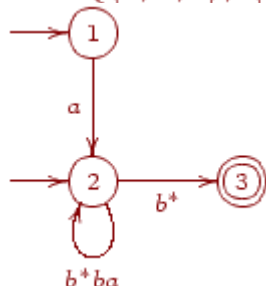
**Замечание 5.3.2.** Каждый конечный автомат можно преобразовать в обобщенный конечный автомат, допускающий те же слова. Для этого достаточно заменить всюду в метках переходов пустое слово на 1, а каждое непустое слово - на произведение его букв.

**Замечание 5.3.3.** Если к обобщенному конечному автомату добавить переход с меткой

0, то множество допускаемых этим автоматом слов не изменится.

**Пример 5.3.4.** Пусть  $\Sigma = \{a, b, c\}$ . Обобщенный конечный автомат  $\langle Q, \Sigma, \Delta, I, F \rangle$ , где  $Q = \{1, 2, 3\}$ ,  $I = \{1, 2\}$ ,  $F = \{3\}$ ,

$\Delta = \{\langle 1, a, 2 \rangle, \langle 2, b^*ba, 2 \rangle, \langle 2, b^*, 3 \rangle\}$ ,



допускает все слова в алфавите  $\Sigma$ , кроме слов, содержащих подслово  $aa$ .

**Теорема 5.3.5 (теорема Клини).** Язык  $L$  является регулярным тогда и только тогда, когда он является автоматным.

*Доказательство.* Пусть  $e$  - регулярное выражение. Индукцией по построению  $e$  легко показать, что задаваемый им язык является автоматным (см. теорему 3.1.1).

Обратно, пусть язык  $L$  распознается некоторым (недетерминированным) конечным автоматом с одним начальным состоянием и одним заключительным состоянием.

Существует эквивалентный ему обобщенный конечный автомат  $\langle Q, \Sigma, \Delta, \{q_1\}, \{q_2\} \rangle$ ,

где  $q_1 \neq q_2$ . Если есть несколько переходов с общим началом и общим концом (такие переходы называются *параллельными*), заменим их на один переход, используя операцию  $+$ .

Устраним по очереди все состояния, кроме  $q_1$  и  $q_2$ . При устранении состояния  $q$  нужно

для каждого перехода вида  $\langle p_1, f_1, q \rangle$ , где  $p_1 \neq q$ , и для каждого перехода вида

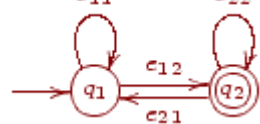
$\langle q, f_2, p_2 \rangle$ , где  $p_2 \neq q$ , добавить переход  $\langle p_1, f_1 g^* f_2, p_2 \rangle$ , где регулярное выражение  $g$  - метка перехода из  $q$  в  $q$  (если нет перехода из  $q$  в  $q$ , то надо добавить

переход  $\langle p_1, f_1 f_2, p_2 \rangle$ ), и снова всюду заменить параллельные переходы на один переход, используя операцию  $+$ .

После устранения всех состояний, кроме  $q_1$  и  $q_2$ , получится обобщенный конечный

автомат  $\langle \{q_1, q_2\}, \Sigma, \Delta', \{q_1\}, \{q_2\} \rangle$ , где

$\Delta' = \{\langle q_1, e_{11}, q_1 \rangle, \langle q_1, e_{12}, q_2 \rangle, \langle q_2, e_{21}, q_1 \rangle, \langle q_2, e_{22}, q_2 \rangle\}$ .



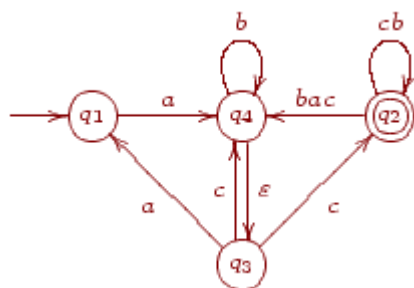
Очевидно, что  $L = L(e_{11}^* e_{12} (e_{22} + e_{21} e_{11}^* e_{12})^*)$ .

**Пример 5.3.6.** Рассмотрим язык, распознаваемый конечным автоматом

$M = \langle \{q_1, q_2, q_3, q_4\}, \Sigma, \Delta, \{q_1\}, \{q_2\} \rangle$ ,

где  $\Sigma = \{a, b, c\}$  и

$\Delta = \{\langle q_1, a, q_4 \rangle, \langle q_2, cb, q_2 \rangle, \langle q_2, bac, q_4 \rangle, \langle q_3, a, q_1 \rangle, \langle q_3, c, q_2 \rangle, \langle q_3, c, q_4 \rangle, \langle q_4, \varepsilon, q_3 \rangle, \langle q_4, b, q_4 \rangle\}$ .

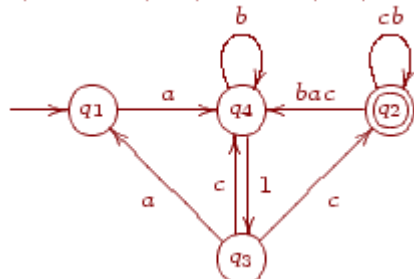


Тот же язык порождается обобщенным конечным автоматом

$$M_1 = \langle \{q_1, q_2, q_3, q_4\}, \Sigma, \Delta_1, \{q_1\}, \{q_2\} \rangle,$$

где

$$\Delta_1 = \{ \langle q_1, a, q_4 \rangle, \langle q_2, cb, q_2 \rangle, \langle q_2, bac, q_4 \rangle, \langle q_3, a, q_1 \rangle, \langle q_3, c, q_2 \rangle, \langle q_3, c, q_4 \rangle, \langle q_4, 1, q_3 \rangle, \langle q_4, b, q_4 \rangle \}.$$

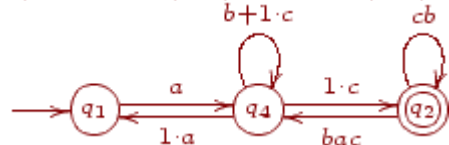


После устранения состояния  $q_3$  получается обобщенный конечный автомат

$$M_2 = \langle \{q_1, q_2, q_4\}, \Sigma, \Delta_2, \{q_1\}, \{q_2\} \rangle,$$

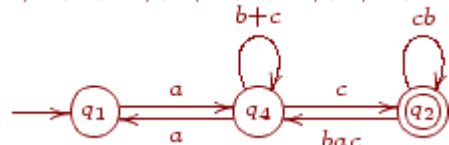
где

$$\Delta_2 = \{ \langle q_1, a, q_4 \rangle, \langle q_2, cb, q_2 \rangle, \langle q_2, bac, q_4 \rangle, \langle q_4, 1 \cdot a, q_1 \rangle, \langle q_4, 1 \cdot c, q_2 \rangle, \langle q_4, b + 1 \cdot c, q_4 \rangle \}.$$



Можно упростить регулярные выражения и получить

$$\Delta'_2 = \{ \langle q_1, a, q_4 \rangle, \langle q_2, cb, q_2 \rangle, \langle q_2, bac, q_4 \rangle, \langle q_4, a, q_1 \rangle, \langle q_4, c, q_2 \rangle, \langle q_4, b + c, q_4 \rangle \}.$$

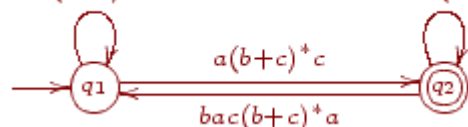


После устранения состояния  $q_4$  и упрощения регулярных выражений получается обобщенный конечный автомат

$$M_3 = \langle \{q_1, q_2\}, \Sigma, \Delta_3, \{q_1\}, \{q_2\} \rangle,$$

где

$$\Delta_3 = \{ \langle q_1, a(b+c)^*a, q_1 \rangle, \langle q_1, a(b+c)^*c, q_2 \rangle, \langle q_2, bac(b+c)^*a, q_1 \rangle, \langle q_2, cb + bac(b+c)^*c, q_2 \rangle \}.$$



Следовательно, язык  $L(M)$  задается регулярным выражением

$$(a(b+c)^*a)^*a(b+c)^*c \cdot (cb + bac(b+c)^*c + bac(b+c)^*a(a(b+c)^*a)^*a(b+c)^*c)^*.$$

Упростив это регулярное выражение, получим

$$L(M) = L(a(aa+b+c+c(cb)^*bac)^*c(cb)^*).$$

**Упражнение 5.3.7.** Найти регулярное выражение для языка, порождаемого грамматикой

$$S \rightarrow bT,$$

$$S \rightarrow aU,$$

$$T \rightarrow cR,$$

$$T \rightarrow eU,$$

$$R \rightarrow dT,$$

$$U \rightarrow \varepsilon.$$

**Упражнение 5.3.8.** Найти регулярное выражение для языка

$$\{w \in \{a, b\}^* \mid (|w|_a - |w|_b) \neq 0\}$$

**Упражнение 5.3.9.** Найти регулярное выражение для языка  $L_1 \cap L_2 \cap L_3$ , где  $L_1 = (aaab+c+d)^*$ ,  $L_2 = (a^*ba^*ba^*bc+d)^*$ ,  $L_3 = ((a+b)^*c(a+b)^*cd)^*$

**Упражнение 5.3.10** Найти регулярное выражение для дополнения языка  $(a+b)^*bbb(a+b)^*$  в алфавите  $\{a, b\}$ .

**Упражнение 5.3.11** Найти регулярное выражение для дополнения языка  $(ab+ba)^*(1+a+b)$  в алфавите  $\{a, b\}$ .

**Упражнение 5.3.12** Найти регулярное выражение для дополнения языка  $(a+b)^*(aab+abaa+abb)(a+b)^*$  в алфавите  $\{a, b\}$ .

**Упражнение 5.3.13** Найти регулярное выражение для дополнения языка  $(aa(ab)^*bb(ab)^*)^*$  в алфавите  $\{a, b\}$ .

## 5.4\*. Звездная высота

**Определение 5.4.1.** Звездная высота (star-height) регулярного выражения (обозначение  $sh(e)$ ) определяется рекурсивно следующим образом:

$$sh(a) = 0,$$

$$sh(0) = 0,$$

$$sh(1) = 1,$$

$$sh(e+f) = \max(sh(e), sh(f)),$$

$$sh(e \cdot f) = \max(sh(e), sh(f)),$$

$$sh(e^*) = 1 + sh(e).$$

**Пример 5.4.2.** Пусть  $\Sigma = \{a, b, c\}$ . Тогда

$$sh((a^*+b^*+ab)^*(ab^*c)^*) = 2.$$

**Определение 5.4.3.** Звездной высотой регулярного языка  $L$  (обозначение  $sh(L)$ ) называется минимум звездных высот регулярных выражений, задающих этот язык.

**Замечание 5.4.4.** Регулярный язык  $L$  является конечным тогда и только тогда, когда  $sh(L) = 0$ .

**Теорема 5.4.5.** Пусть  $|\Sigma| \geq 2$ . Тогда для любого  $n \in \mathbb{N}$  существует такой регулярный язык  $L \subseteq \Sigma^*$ , что  $sh(L) = n$ .

Доказательство можно найти в книге Саломаа А. Жемчужины теории формальных языков. - М.: Мир, 1986 с.41-46.

**Пример 5.4.6.** Пусть  $\Sigma = \{a, b\}$  и  $L = \{w \in \Sigma^* \mid (|w|_a - |w|_b) \div 4\}$ .

Тогда  $sh(L) = 2$ . Действительно, язык  $L$  задается регулярным выражением  $(ab+ba+(aa+bb)(ab+ba)^*(aa+bb))^*$  и не задается никаким регулярным выражением меньшей звездной высоты.

**Замечание 5.4.7.** Неизвестно, верен ли аналог теоремы 5.4.5 для *обобщенных регулярных выражений*, в которых, помимо итерации, конкатенации и объединения, разрешена операция дополнения.

**Упражнение 5.4.8.** Уменьшить звездную высоту регулярного выражения  $(a^*+b^*+ab)^*$ .

**Упражнение 5.4.9.** Уменьшить звездную высоту регулярного выражения  $(c(a^*b)^*)^*$ .

**Упражнение 5.4.10.** Уменьшить звездную высоту регулярного выражения  $(a(ab)^*b)^*$ .

**Упражнение 5.4.11.** Существует ли такой регулярный язык  $L \subseteq \{a\}^*$ , что  $sh(L) = 2$ ?



## Математическая теория формальных языков

### 6. Лекция: Синтаксические моноиды: версия для печати и PDA

В данной лекции рассматривается понятие синтаксического моноида и приводится доказательство еще одного критерия автоматности формального языка, который можно сформулировать в терминах классов эквивалентности слов по взаимозаменяемости. Приведены примеры практической реализации критерия и предоставлены упражнения для самостоятельного решения

Основная цель данной лекции - доказать еще один критерий автоматности формального языка. Этот критерий можно сформулировать в терминах классов эквивалентности слов по взаимозаменяемости (однако формальные определения будут даны без использования понятия класса эквивалентности). Слова  $x$  и  $y$  считаются взаимозаменяемыми (относительно языка  $L$ ), если при замене в любом слове из языка  $L$  подслова, совпадающего с  $x$ , на  $y$  снова получится слово из языка  $L$  и наоборот. В разделе 6.3 фактически доказывается, что язык  $L$  является автоматным тогда и только тогда, когда соответствующее отношение взаимозаменяемости разбивает множество всех слов рассматриваемого алфавита на конечное число классов эквивалентности.

Но сначала мы докажем аналогичный результат для отношения взаимозаменяемости не подслов, а префиксов (раздел 6.1). Соответствующие классы эквивалентности слов позволяют построить минимальный детерминированный конечный автомат для заданного языка. Известен и другой метод нахождения минимального детерминированного конечного автомата (раздел 6.2), но этот метод можно применять только тогда, когда уже имеется какой-нибудь детерминированный конечный автомат, распознающий данный язык. В конце лекции доказывается, что классы эквивалентности по взаимозаменяемости относительно автоматного языка сами являются автоматными языками.

#### 6.1. Множества правых контекстов

**Определение 6.1.1.** Пусть  $L \subseteq \Sigma^*$  и  $y \in \Sigma^*$ . Тогда множество правых контекстов слова  $y$  относительно языка  $L$  определяется следующим образом:

$$C_L^{(r)}(y) = \{z \in \Sigma^* \mid yz \in L\}.$$

**Пример 6.1.2.** Пусть  $\Sigma = \{a, b\}$  и  $L = \{a^n b a^n \mid n \geq 0\}$ . Тогда

1.  $C_L^{(r)}(a^i) = \{a^k b a^{k+i} \mid k \geq 0\}$
2. если  $i \geq j$ , то  $C_L^{(r)}(a^i b a^j) = \{a^{i-j}\}$ ;
3. если  $i < j$ , то  $C_L^{(r)}(a^i b a^j) = \emptyset$ ;
4. если  $|y|_b > 1$ , то  $C_L^{(r)}(y) = \emptyset$ .

**Определение 6.1.3.** Для любого состояния  $p$  полного детерминированного конечного автомата  $\langle Q, \Sigma, \Delta, I, F \rangle$  и любого слова  $w$  обозначим через  $\Delta_w(p)$  такое состояние  $q$ , что существует путь из  $p$  в  $q$  с меткой  $w$  (в силу полноты и детерминированности такое состояние существует и единственно).

**Лемма 6.1.4.** Если язык  $L$  распознается полным детерминированным конечным автоматом  $\langle Q, \Sigma, \Delta, I, F \rangle$ , то

$$|\{C_L^{(r)}(y) \mid y \in \Sigma^*\}| \leq |Q|.$$

**Доказательство.** Пусть  $I = \{s\}$ . Введем обозначение

$$J = \{C_L^{(r)}(y) \mid y \in \Sigma^*\}.$$

Определим функцию  $f: J \rightarrow Q$ , положив  $f(A)$  равным  $\Delta_y(s)$ , где  $y$  - некоторое слово, для которого выполнено условие  $C_L^{(r)}(y) = A$  (если существует несколько таких слов  $y$ , то можно использовать, например, первое среди них в лексикографическом порядке).

Заметим, что для любых слов  $u$  и  $v$ , если  $C_L^{(r)}(u) \neq C_L^{(r)}(v)$ , то  $\Delta_u(s) \neq \Delta_v(s)$ .

Следовательно, функция  $f$  является инъективной. Но тогда  $|J| \leq |Q|$ .

**Пример 6.1.5.** Рассмотрим язык  $L$ , порождаемый полным детерминированным

конечным автоматом  $\langle Q, \Sigma, \Delta, I, F \rangle$  из примера 2.6.4. Тогда

$$\{C_L^{(r)}(y) \mid y \in \Sigma^*\} = \{C_L^{(r)}(\varepsilon), C_L^{(r)}(a), C_L^{(r)}(b)\},$$

$$C_L^{(r)}(\varepsilon) = \{(ab)^n \mid n \geq 0\} \cup \{(ab)^n a \mid n \geq 0\},$$

$$C_L^{(r)}(a) = \{b(ab)^n \mid n \geq 0\} \cup \{(ba)^n \mid n \geq 0\},$$

$$C_L^{(r)}(b) = \emptyset.$$

**Лемма 6.1.6.** Если  $L \subseteq \Sigma^*$  и множество  $\{C_L^{(r)}(y) \mid y \in \Sigma^*\}$  конечно, то язык  $L$  является автоматным.

**Доказательство.** Язык  $L$  распознается полным детерминированным конечным

автоматом  $\langle Q, \Sigma, \Delta, I, F \rangle$ , где

$$Q = \{C_L^{(r)}(y) \mid y \in \Sigma^*\},$$

$$I = \{C_L^{(r)}(\varepsilon)\},$$

$$F = \{C_L^{(r)}(y) \mid y \in L\},$$

$$\Delta = \{\langle C_L^{(r)}(y), a, C_L^{(r)}(ya) \rangle \mid y \in \Sigma^*, a \in \Sigma\}.$$

**Пример 6.1.7.** Пусть  $\Sigma = \{a, b\}$ . Рассмотрим автоматный язык  $L = a^+b^*$ . Обозначим

$$q_\varepsilon = C_L^{(r)}(\varepsilon),$$

$$q_a = C_L^{(r)}(a),$$

$$q_b = C_L^{(r)}(b) = \emptyset,$$

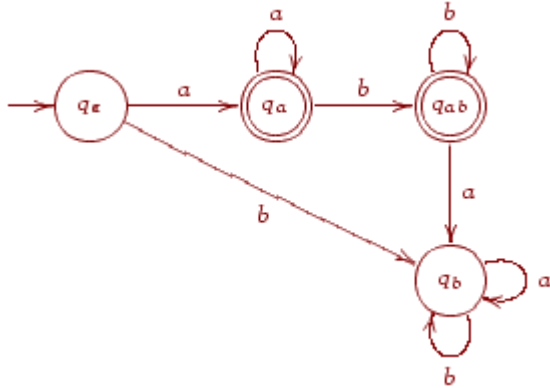
$$q_{ab} = C_L^{(r)}(ab).$$

Тогда  $\{C_L^{(r)}(y) \mid y \in \Sigma^*\} = \{q_\varepsilon, q_a, q_b, q_{ab}\}$ . Язык  $L$  распознается полным

детерминированным конечным автоматом  $\langle Q, \Sigma, \Delta, I, F \rangle$ , где  $Q = \{q_\varepsilon, q_a, q_b, q_{ab}\}$ ,

$$I = \{q_\varepsilon\}, F = \{q_a, q_{ab}\},$$

$$\Delta = \{ \langle q_\varepsilon, a, q_a \rangle, \langle q_\varepsilon, b, q_b \rangle, \langle q_a, a, q_a \rangle, \langle q_a, b, q_{ab} \rangle, \langle q_b, a, q_b \rangle, \langle q_b, b, q_b \rangle, \langle q_{ab}, a, q_b \rangle, \langle q_{ab}, b, q_{ab} \rangle \}.$$



**Теорема 6.1.8.** Язык  $L \subseteq \Sigma^*$  является автоматным тогда и только тогда, когда множество  $\{C_L^{(r)}(y) \mid y \in \Sigma^*\}$  конечно.

*Доказательство.* Необходимость доказана в лемме 6.1.4, достаточность - в лемме 6.1.6.

**Замечание 6.1.9.** В силу леммы 6.1.4 полный детерминированный конечный автомат, построенный в доказательстве леммы 6.1.6, является минимальным (по количеству состояний) среди всех полных детерминированных конечных автоматов, распознающих заданный язык. Можно доказать, что любой минимальный полный детерминированный конечный автомат, распознающий заданный язык, изоморфен этому автомату.

**Упражнение 6.1.10.** Пусть  $\Sigma = \{a, b\}$  и  $L = \{w \in \Sigma^* \mid |w|_a = 1, |w|_b = 2\}$ .

Сколько элементов содержит множество  $\{C_L^{(r)}(y) \mid y \in \Sigma^*\}$  ?

**Упражнение 6.1.11.** Найти минимальный полный детерминированный конечный автомат для языка  $\{ab, abb\}^*$ .

**Упражнение 6.1.12.** Найти минимальный полный детерминированный конечный автомат для языка

$$\{ab^{2n} \mid n \geq 0\} \cup \{cb^{2n+1} \mid n \geq 0\}.$$

**Упражнение 6.1.13.** Найти минимальный полный детерминированный конечный автомат для языка

$$\{ab^{2n} \mid n \geq 0\} \cup \{b^{4n} \mid n \geq 0\}.$$

**Упражнение 6.1.14.** Найти минимальный полный детерминированный конечный автомат для языка

$$\{a^m ub^n \mid m \geq 0, n \geq 0, |u| \geq 3\}.$$

## 6.2. Минимизация детерминированных конечных автоматов

**Определение 6.2.1.** Говорят, что слово  $w$  различает состояния  $p$  и  $q$  полного детерминированного конечного автомата  $\langle Q, \Sigma, \Delta, I, F \rangle$ , если одно из состояний  $\Delta_w(p)$ ,  $\Delta_w(q)$  заключительное, а другое не является заключительным.

**Определение 6.2.2.** Состояния  $p$  и  $q$  полного детерминированного конечного автомата  $\langle Q, \Sigma, \Delta, I, F \rangle$  называются *различимыми* (distinguishable), если существует слово  $w$ , которое их различает.

**Замечание 6.2.3.** Пусть полный детерминированный конечный автомат

$\langle Q, \Sigma, \Delta, \{q_s\}, F \rangle$  задает язык  $L$ . Рассмотрим произвольные слова  $u \in \Sigma^*$  и  $v \in \Sigma^*$ .

Состояния  $\Delta_u(q_s)$  и  $\Delta_v(q_s)$  различимы тогда и только тогда, когда

$$C_L^{(i)}(u) \neq C_L^{(i)}(v).$$

**Теорема 6.2.4.** Существует быстрый алгоритм, позволяющий по произвольному детерминированному конечному автомату находить минимальный (по количеству состояний) автомат среди детерминированных конечных автоматов, эквивалентных исходному автомату.

*Доказательство.* Без ограничения общности можно предположить, что дан полный

детерминированный конечный автомат  $\langle Q, \Sigma, \Delta, I, F \rangle$ , все состояния которого достижимы из начального состояния (для обеспечения полноты достаточно добавить одно состояние). Для каждого натурального числа  $i$  определим на множестве  $Q$  отношение эквивалентности  $\equiv_i$ :

$$p \equiv_0 q \iff (p \in F \text{ и } q \in F) \text{ или } (p \notin F \text{ и } q \notin F),$$

$$p \equiv_{i+1} q \iff p \equiv_i q \text{ и } \Delta_a(p) \equiv_i \Delta_a(q) \text{ для каждого } a \in \Sigma.$$

Легко видеть, что  $p \equiv_i q$  тогда и только тогда, когда никакое слово длины не больше  $i$

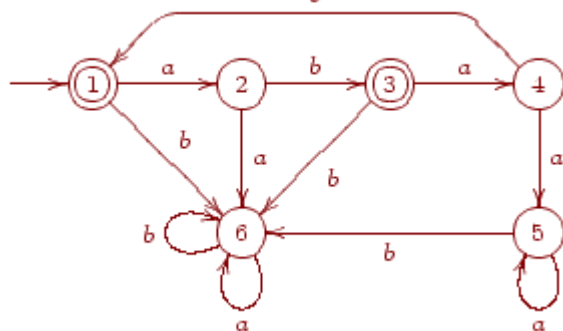
не различает  $p$  и  $q$ . Обозначим  $n = |Q|$ . Легко проверить, что для любого  $k \geq n$

отношение  $\equiv_k$  совпадает с отношением  $\equiv_{n-1}$ . Используя классы эквивалентности

отношения  $\equiv_{n-1}$  как состояния, можно построить минимальный полный детерминированный конечный автомат. Удалив из него бесполезное состояние (из которого не достижимо ни одно заключительное состояние), если такое имеется, получим искомый минимальный детерминированный конечный автомат.

**Пример 6.2.5.** Рассмотрим полный детерминированный конечный автомат

$\langle \{1, 2, 3, 4, 5, 6\}, \{a, b\}, \Delta, \{1\}, \{1, 3\} \rangle$ , где  
 $\Delta = \{ \langle 1, a, 2 \rangle, \langle 2, b, 3 \rangle, \langle 3, a, 4 \rangle, \langle 4, b, 1 \rangle, \langle 4, a, 5 \rangle, \langle 5, a, 5 \rangle, \langle 1, b, 6 \rangle, \langle 2, a, 6 \rangle, \langle 3, b, 6 \rangle, \langle 5, b, 6 \rangle, \langle 6, a, 6 \rangle, \langle 6, b, 6 \rangle \}$ .



Он эквивалентен минимальному детерминированному конечному автомату

$\langle \{\{1, 3\}, \{2, 4\}\}, \{a, b\}, \Delta', \{\{1, 3\}\}, \{\{1, 3\}\} \rangle$ , где  
 $\Delta' = \{ \langle \{1, 3\}, a, \{2, 4\} \rangle, \langle \{2, 4\}, b, \{1, 3\} \rangle \}$ .



**Замечание 6.2.6.** Неизвестно, существует ли быстрый (полиномиальный) алгоритм,

позволяющий по произвольному конечному автомату находить минимальный автомат среди всех (не обязательно детерминированных) конечных автоматов, эквивалентных исходному автомату.

**Упражнение 6.2.7.** Найти минимальный полный детерминированный конечный автомат для языка, порождаемого грамматикой

$$\begin{array}{ll} C \rightarrow aD, & H \rightarrow aC, \\ C \rightarrow bE, & H \rightarrow bE, \\ D \rightarrow aE, & D \rightarrow \varepsilon, \\ D \rightarrow bH, & H \rightarrow \varepsilon. \\ E \rightarrow aD, & \\ E \rightarrow bC, & \end{array}$$

**Упражнение 6.2.8.** Найти минимальный полный детерминированный конечный автомат для языка

$$(1 + (a+b)^*b)(a+b)^*.$$

**Упражнение 6.2.9.** Найти минимальный полный детерминированный конечный автомат для языка

$$(a+b)^*(aaba + (aabb + bbaa)abbb)(a+b)^*.$$

**Упражнение 6.2.10.** Равны ли регулярные выражения

$$(aa+b+ab)^* \quad \text{и} \quad ((a+b)^*b+1)(aa)^*?$$

**Упражнение 6.2.11.** Равны ли регулярные выражения

$$(a+b)^*aa(a+b)^*bb(a+b)^* \quad \text{и} \quad (ab+b)^*aa(a+b)^*bb(a+ba)^*?$$

**Упражнение 6.2.12.** Равны ли регулярные выражения

$$((ba+bb)(aa+ab)^*)^*a \quad \text{и} \quad b(aa+ab+ba+bb)^*(aa+ba)?$$

**Упражнение 6.2.13.** Равны ли регулярные выражения

$$(ca+cb+cc+(a+b^+(a+c))^+c)(a+b+c)^*$$

и

$$c(a+b+bc)^++(a+b+c)^*(ac+cc)(a+b+bc)^*?$$

### 6.3. Множества двусторонних контекстов

**Определение 6.3.1.** Пусть  $L \subseteq \Sigma^*$  и  $y \in \Sigma^*$ . Тогда *множество контекстов* (*множество двусторонних контекстов*) слова  $y$  относительно языка  $L$  определяется следующим образом:

$$C_L(y) = \{ \langle x, z \rangle \in \Sigma^* \times \Sigma^* \mid xyz \in L \}.$$

**Пример 6.3.2.** Пусть  $\Sigma = \{a, b\}$  и  $L = \{a^nba^n \mid n \geq 0\}$ . Тогда

$$\begin{aligned}
C_L(a^i) &= \{\langle a^m, a^k b a^{k+i+m} \rangle \mid m \geq 0, k \geq 0\} \cup \\
&\quad \cup \{\langle a^{k+i+m} b a^k, a^m \rangle \mid k \geq 0, m \geq 0\}, \\
C_L(a^i b a^j) &= \{\langle a^k, a^m \rangle \mid k \geq 0, m \geq 0, k+i = m+j\}, \\
C_L^{(r)}(y) &= \emptyset, \text{ если } |y|_b > 1.
\end{aligned}$$

**Лемма 6.3.3.** Если  $C_L(u_1) = C_L(u_2)$ , то  $C_L^{(r)}(u_1) = C_L^{(r)}(u_2)$ .

*Доказательство.* Из определений следует, что

$$C_L^{(r)}(y) = \{z \in \Sigma^* \mid \langle \varepsilon, z \rangle \in C_L(y)\}.$$

**Лемма 6.3.4.** Если  $C_L(u_1) = C_L(u_2)$ , то  $C_L(u_1 v) = C_L(u_2 v)$  и  $C_L(v u_1) = C_L(v u_2)$ .

*Доказательство.* Пусть  $C_L(u_1) = C_L(u_2)$  и  $\langle x, z \rangle \in C_L(u_1 v)$ . Тогда  $x u_1 v z \in L$ . Следовательно,  $\langle x, v z \rangle \in C_L(u_1)$ . Далее, получаем, что  $\langle x, v z \rangle \in C_L(u_2)$ ,  $x u_2 v z \in L$  и  $\langle x, z \rangle \in C_L(u_2 v)$ . Второе равенство доказывается аналогично.

**Лемма 6.3.5.** Если  $C_L(u_1) = C_L(u_2)$  и  $C_L(v_1) = C_L(v_2)$ , то  $C_L(u_1 v_1) = C_L(u_2 v_2)$ .

**Определение 6.3.6.** Пусть  $L \subseteq \Sigma^*$ . Тогда множество  $\text{Synt}(L) = \{C_L(y) \mid y \in \Sigma^*\}$  называется *синтаксическим моноидом* (syntactic monoid) языка  $L$ .

**Определение 6.3.7\*.** Полугруппой (semigroup)  $\langle M, \star \rangle$  называется непустое множество  $M$  с ассоциативной бинарной операцией  $\star$ .

**Определение 6.3.8\*.** Пусть  $\langle M, \star \rangle$  - полугруппа. Элемент  $e \in M$  называется *единицей* (unit), если  $e \star x = x = x \star e$  для каждого  $x \in M$ .

**Определение 6.3.9\*.** Моноид  $\langle M, \star, e \rangle$  - это полугруппа  $\langle M, \star \rangle$  с единицей  $e \in M$ .

**Теорема 6.3.10\*.** Определим бинарную операцию на  $\text{Synt}(L)$  следующим образом:  $C_L(u) \star C_L(v) = C_L(uv)$ .

Тогда  $\langle \text{Synt}(L), \star, C_L(\varepsilon) \rangle$  является моноидом.

**Теорема 6.3.11.** Синтаксический моноид  $\text{Synt}(L)$  конечен тогда и только тогда, когда язык  $L$  является автоматным.

*Доказательство.* Пусть множество  $\text{Synt}(L)$  конечно. Согласно лемме 6.3.3 множество

$$\{C_L^{(r)}(y) \mid y \in \Sigma^*\}$$

тоже конечно. В силу леммы 6.1.6 язык  $L$  является автоматным.

Обратно, пусть язык  $L$  распознается некоторым конечным автоматом

$M = \langle Q, \Sigma, \Delta, I, F \rangle$ , не содержащим переходов с метками длины больше единицы.

Поставим в соответствие каждому слову  $y$  множество  $\text{Tr}_M(y) \subseteq Q \times Q$ , определенное следующим образом:

$\text{Tr}_M(y) = \{ \langle p, q \rangle \in Q \times Q \mid \text{существует путь из } p \text{ в } q \text{ с меткой } y \}$ .

Легко проверить, что если  $\text{Tr}_M(y_1) = \text{Tr}_M(y_2)$ , то  $C_L(y_1) = C_L(y_2)$ .

Следовательно,  $|\text{Synt}(L)| \leq 2^{n^2}$ , где  $n = |Q|$ .

**Пример 6.3.12.** Рассмотрим конечный автомат  $M$  из примера 2.1.14. Тогда

1.  $\text{Tr}_M(\varepsilon) = \{ \langle 1, 1 \rangle, \langle 2, 2 \rangle \}$ ;
2. если  $n \geq 1$ , то  $\text{Tr}_M((ab)^n) = \{ \langle 1, 1 \rangle \}$ ;
3. если  $n \geq 1$ , то  $\text{Tr}_M((ba)^n) = \{ \langle 2, 2 \rangle \}$ ;
4. если  $n \geq 0$ , то  $\text{Tr}_M((ab)^n a) = \{ \langle 1, 2 \rangle \}$ ;
5. если  $n \geq 0$ , то  $\text{Tr}_M(b(ab)^n) = \{ \langle 2, 1 \rangle \}$ ;
6.  $\text{Tr}_M(xaay) = \emptyset$ ;
7.  $\text{Tr}_M(xbb y) = \emptyset$ .

**Лемма 6.3.13.** Пусть  $L \subseteq \Sigma^*$ ,  $n \in \mathbb{N}$  и для каждого слова  $y \in \Sigma^*$  длины  $n$  найдется такое слово  $x \in \Sigma^*$ , что  $|x| < n$  и  $C_L(x) = C_L(y)$ . Тогда  $\text{Synt}(L) = \{ C_L(y) \mid y \in \Sigma^*, |y| < n \}$ .

*Доказательство.* Индукцией по  $k \geq n$  можно доказать, что для каждого слова  $y \in \Sigma^*$  длины  $k$  найдется такое слово  $x \in \Sigma^*$ , что  $|x| < n$  и  $C_L(x) = C_L(y)$ . В шаге индукции используется лемма 6.3.5.

**Упражнение 6.3.14.** Сколько элементов в синтаксическом моноиде языка  $a^+b$  над алфавитом  $\{a, b\}$ ?

**Упражнение 6.3.15.** Сколько элементов в синтаксическом моноиде языка  $b^+a^+$  над алфавитом  $\{a, b\}$ ?

**Упражнение 6.3.16.** Сколько элементов в синтаксическом моноиде языка  $(aa+b)^*$  над алфавитом  $\{a, b\}$ ?

**Упражнение 6.3.17.** Сколько элементов в синтаксическом моноиде языка  $(ab)^*(ba)^{*+}a^*$  над алфавитом  $\{a, b\}$ ?

**Упражнение 6.3.18.** Сколько элементов в синтаксическом моноиде языка  $a(b+c)^*a(a+b+c)^*+b(a+c)^*b(a+b+c)^*+c(a+b)^*c(a+b+c)^*$  над алфавитом  $\{a, b, c\}$ ?

## 6.4\*. Классы эквивалентности слов

**Лемма 6.4.1.** Пусть  $L \subseteq \Sigma^*$  и  $y \in \Sigma^*$ . Тогда

$$C_{\Sigma^+ - L}^{(i)}(y) = \Sigma^* - C_L^{(i)}(y),$$

$$C_{\Sigma^+ - L}(y) = (\Sigma^* \times \Sigma^*) - C_L(y).$$



**Определение 6.4.2.** Пусть  $L \subseteq \Sigma^*$  и  $x \in \Sigma^*$ . Обозначим через  $[x]_L^{(r)}$  язык

$\{y \in \Sigma^* \mid C_L^{(r)}(y) = C_L^{(r)}(x)\}$ . Обозначим через  $[x]_L$  язык  $\{y \in \Sigma^* \mid C_L(y) = C_L(x)\}$ .

**Пример 6.4.3.** Пусть  $L \subseteq \Sigma^*$ . Множества вида  $[x]_L^{(r)}$  образуют разбиение множества  $\Sigma^*$  на классы эквивалентности. Множества вида  $[x]_L$  образуют разбиение множества  $\Sigma^*$  на классы эквивалентности.

**Пример 6.4.4.** Пусть  $\Sigma = \{a, b\}$  и  $L = \{a^n b a^n \mid n \geq 0\}$ . Тогда  $[baa]_L^{(r)} = \{a^i b a^j \mid 0 \leq i < j\} \cup \{y \in \Sigma^* \mid |y|_b > 1\}$ ,  
 $[baa]_L = \{a^i b a^{i+2} \mid i \geq 0\}$ .

**Лемма 6.4.5.** Если язык  $L \subseteq \Sigma^*$  является автоматным, то для каждого слова  $x \in \Sigma^*$  языки  $[x]_L^{(r)}$  и  $[x]_L$  являются автоматными.

*Доказательство.* Пусть язык  $L$  распознается конечным автоматом

$M = \langle Q, \Sigma, \Delta, I, F \rangle$ , не содержащим переходов с метками длины больше единицы. Будем использовать обозначение  $\text{Tr}_M$  из доказательства теоремы 6.3.11. При любой

фиксированной паре  $\langle p, q \rangle \in Q \times Q$  язык  $\mathcal{A}_{p,q,M} = \{y \in \Sigma^* \mid \langle p, q \rangle \in \text{Tr}_M(y)\}$  является автоматным (он распознается конечным автоматом  $\langle Q, \Sigma, \Delta, \{p\}, \{q\} \rangle$ ). Для каждого слова  $z \in \Sigma^*$  язык  $\mathcal{B}_{z,M} = \{y \in \Sigma^* \mid \text{Tr}_M(y) = \text{Tr}_M(z)\}$  является автоматным, так как он представим в виде пересечения конечного семейства автоматных языков:

$$\mathcal{B}_{z,M} = \left( \bigcap_{\langle p,q \rangle \in \text{Tr}_M(z)} \mathcal{A}_{p,q,M} \right) \cap \left( \bigcap_{\langle p,q \rangle \notin \text{Tr}_M(z)} (\Sigma^* - \mathcal{A}_{p,q,M}) \right).$$

Каждый из языков  $[x]_L$  и  $[x]_L^{(r)}$  является объединением конечного семейства автоматных языков:

$$[x]_L = \bigcup_{z \in [x]_L} \mathcal{B}_{z,M}, \quad [x]_L^{(r)} = \bigcup_{z \in [x]_L^{(r)}} \mathcal{B}_{z,M}.$$

**Замечание 6.4.6.** Из теоремы 6.1.8 вытекает, что если язык  $L$  автоматный, то существует лишь конечное число различных множеств  $[x]_L^{(r)}$ . Аналогичное утверждение верно для множеств  $[x]_L$  (см. теорему 6.3.11).

**Пример 6.4.7.** Рассмотрим язык  $L = \{(ab)^n \mid n \geq 0\} \cup \{(ab)^n a \mid n \geq 0\}$  над алфавитом  $\Sigma = \{a, b\}$ . Тогда



$$\begin{aligned}
[\varepsilon]_L^{(r)} &= (ab)^*; \\
[a]_L^{(r)} &= (ab)^*a; \\
[b]_L^{(r)} &= b(a+b)^* + (a+b)^*(aa+bb)(a+b)^*; \\
[\varepsilon]_L &= 1; \\
[ab]_L &= (ab)^+; \\
[ba]_L &= (ba)^+; \\
[a]_L &= (ab)^*a; \\
[b]_L &= b(ab)^*; \\
[aa]_L &= (a+b)^*(aa+bb)(a+b)^*.
\end{aligned}$$

**Упражнение 6.4.8.** Существуют ли такие языки  $L_1 \subseteq \Sigma^*$  и  $L_2 \subseteq \Sigma^*$ , что язык  $L_1$  является автоматным, но язык

$$L_3 = \{x \in \Sigma^* \mid xy \in L_1 \text{ для некоторого слова } y \in L_2\}$$

не является автоматным?

**Упражнение 6.4.9.** Существуют ли такие языки  $L_1 \subseteq \Sigma^*$  и  $L_2 \subseteq \Sigma^*$ , что язык  $L_1$  является автоматным, но язык

$$L_3 = \{x \in \Sigma^* \mid xy \in L_1 \text{ для всех слов } y \in L_2\}$$

не является автоматным?

**Упражнение 6.4.10.** Существует ли такой автоматный язык  $L \subseteq \Sigma^*$ , что язык  $\text{Max}(L) = \{x \in L \mid xy \notin L \text{ для всех слов } y \in \Sigma^+\}$  не является автоматным?

**Теорема 6.4.11.** Язык  $L \subseteq \Sigma^*$  является автоматным тогда и только тогда, когда существует такое отношение эквивалентности  $R \subseteq \Sigma^* \times \Sigma^*$ , что  $R$  разбивает  $\Sigma^*$  на конечное множество классов эквивалентности,  $L$  является объединением некоторых из этих классов эквивалентности и для любых  $x \in \Sigma^*$ ,  $y \in \Sigma^*$ ,  $z \in \Sigma^*$  из  $xRy$  следует  $xzRyz$ .

**Замечание 6.4.12.** Теоремы 6.1.8 и 6.4.11 образуют теорему Майхилла-Нерода.

## Математическая теория формальных языков

### 7. Лекция: Неоднозначность в контекстно-свободных грамматиках: версия для печати и PDA

В данной лекции рассматривается класс контекстно-свободных языков, которые находят применение в разнообразных программных продуктах, таких как компиляторы, средства форматирования исходного кода, средства статического анализа программ, синтаксические редакторы, системы верстки, программы просмотра форматированного текста, поисковые системы. Приведены практические примеры и упражнения для самостоятельного решения

Перейдем к более богатому классу языков - к контекстно-свободным языкам. Они находят применение в разнообразных программных продуктах, таких как компиляторы, средства форматирования исходного кода, средства статического анализа программ, синтаксические редакторы, системы верстки, программы просмотра форматированного текста поисковые системы.

Начнем рассмотрение контекстно-свободных языков с введения понятия деревьев вывода, что позволит определить важное для компьютерных приложений понятие однозначности контекстно-свободной грамматики. Чтобы не вдаваться в детали определения изоморфизма ориентированных деревьев, будем использовать в определении однозначности понятие левостороннего вывода (раздел 7.2). Соответствие деревьев вывода и левосторонних (а также правосторонних) выводов понадобится также в [лекции 13](#).

Из класса всех контекстно-свободных языков можно выделить подкласс тех языков, для которых существует хотя бы одна однозначная грамматика. В разделе 7.3\* доказывается, что все праволинейные (то есть автоматные) языки принадлежат этому подклассу.

В последнем разделе этой лекции приводятся важные конкретные примеры однозначных контекстно-свободных грамматик, моделирующих системы правильно вложенных скобок и польскую (префиксную) запись выражений.

#### 7.1. Деревья вывода

**Определение 7.1.1.** Выводам в контекстно-свободной грамматике соответствуют так называемые *деревья вывода* (*деревья разбора*, derivation tree, parse tree) - некоторые упорядоченные деревья, вершины которых помечены символами алфавита  $N \cup \Sigma$ . Корень дерева отвечает начальному символу. Каждому символу слова  $w_1$ , на которое заменяется начальный символ на первом шаге вывода, ставится в соответствие вершина дерева, и к ней проводится дуга из корня. Полученные таким образом непосредственные потомки корня упорядочены согласно порядку их меток в слове  $w_1$ . Для тех из полученных вершин, которые помечены символами из множества  $N$ , делается аналогичное построение, и т. д.

*Кроной* (yield) дерева вывода называется слово, записанное в вершинах, помеченных символами из алфавита  $\Sigma$ .

**Пример 7.1.2.** Рассмотрим контекстно-свободную грамматику

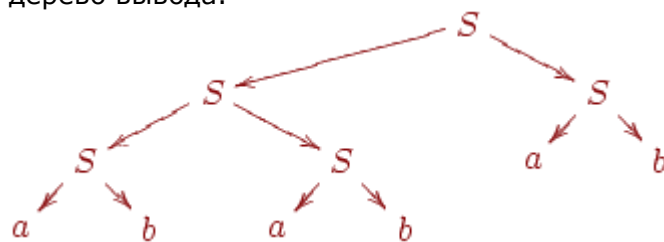
$$S \rightarrow SS,$$

$$S \rightarrow ab,$$

$$S \rightarrow aSb.$$

Выводу  $S \Rightarrow SS \Rightarrow Sab \Rightarrow SSab \Rightarrow abSab \Rightarrow ababab$  соответствует следующее

дерево вывода.



Крона этого дерева вывода - ababab.

**Упражнение 7.1.3.** Перечислить все деревья вывода в грамматике

$S \rightarrow a,$

$S \rightarrow bR,$

$R \rightarrow cTT,$

$T \rightarrow a,$

$T \rightarrow ba.$

**Упражнение 7.1.4.** Существует ли праволинейная грамматика без  $\epsilon$ -правил, в которой некоторое слово имеет бесконечно много выводов?

## 7.2. Однозначные контекстно-свободные грамматики

**Определение 7.2.1.** Вывод в контекстно-свободной грамматике называется *левосторонним* (*левым*, leftmost derivation), если на каждом шаге вывода заменяется самое левое из всех вхождений вспомогательных символов (то есть каждый шаг вывода имеет вид  $uA\theta \Rightarrow u\beta\theta$ , где  $(A \rightarrow \beta) \in P$ ,  $u \in \Sigma^*$  и  $\theta \in (N \cup \Sigma)^*$ ). Иногда в левосторонних выводах вместо  $\Rightarrow$  пишут  $\xRightarrow{\text{lm}}$ . *Правосторонний* (*правый*) вывод определяется аналогично. В правосторонних выводах вместо  $\Rightarrow$  пишут  $\xRightarrow{\text{rm}}$ .

**Пример 7.2.2.** Вывод

$S \Rightarrow SS \Rightarrow Sab \Rightarrow SSab \Rightarrow abSab \Rightarrow ababab$   
из примера 7.2.1 не является левосторонним.

**Лемма 7.2.3.** Для каждого слова, выводимого в контекстно-свободной грамматике, существует левосторонний вывод.

**Лемма 7.2.4.** Пусть  $G$  - контекстно-свободная грамматика над алфавитом  $\Sigma$ . Пусть  $w \in \Sigma^*$ . Тогда существует взаимно-однозначное соответствие между левосторонними выводами слова  $w$  в грамматике  $G$  и деревьями вывода в грамматике  $G$ , кроной которых является  $w$ .

**Пример 7.2.5.** Рассмотрим дерево вывода из примера 7.1.2. Ему соответствует левосторонний вывод

$S \xRightarrow{\text{lm}} SS \xRightarrow{\text{lm}} SSS \xRightarrow{\text{lm}} abSS \xRightarrow{\text{lm}} ababS \xRightarrow{\text{lm}} ababab.$

**Определение 7.2.6.** Контекстно-свободная грамматика называется *неоднозначной* (ambiguous), если существует слово, которое имеет два или более левосторонних вывода (устаревший термин - *неопределенная* грамматика). В противном случае контекстно-свободная грамматика называется *однозначной* (unambiguous).

**Пример 7.2.7.** Контекстно-свободная грамматика из примера 7.2.1 неоднозначна. Слово ababab имеет два левосторонних вывода:

$$S \Rightarrow_{lm} SS \Rightarrow_{lm} SSS \Rightarrow_{lm} abSS \Rightarrow_{lm} ababS \Rightarrow_{lm} ababab$$

и

$$S \Rightarrow_{lm} SS \Rightarrow_{lm} abS \Rightarrow_{lm} abSS \Rightarrow_{lm} ababS \Rightarrow_{lm} ababab.$$

**Пример 7.2.8.** Пусть  $\Sigma = \{p, \#, \neg, \wedge, \vee\}$ . Контекстно-свободная грамматика

$$\begin{aligned} S &\rightarrow S \vee D, & C &\rightarrow \neg C, \\ S &\rightarrow D, & C &\rightarrow (S), \\ D &\rightarrow D \wedge C, & C &\rightarrow V, \\ D &\rightarrow C, & V &\rightarrow V \#, \\ & & V &\rightarrow p \end{aligned}$$

порождает множество всех булевых формул, составленных из переменных  $p, p\#, p\#\#, \dots$ , с помощью скобок и операторов  $\neg, \wedge$  и  $\vee$ . Эта грамматика является однозначной.

**Определение 7.2.9.** Контекстно-свободный язык называется *существенно неоднозначным* (inherently ambiguous), если каждая контекстно-свободная грамматика, порождающая этот язык, является неоднозначной.

**Пример 7.2.10.** Язык, порождаемый контекстно-свободной грамматикой из примера 7.2.1, не является существенно неоднозначным. Он порождается однозначной грамматикой

$$\begin{aligned} S &\rightarrow TS, \\ S &\rightarrow T, \\ T &\rightarrow ab, \\ T &\rightarrow aSb. \end{aligned}$$

**Пример 7.2.11.** Пусть  $\Sigma = \{a, b, c\}$ . Контекстно-свободный язык  $\{a^k b^m c^n \mid k = m \text{ или } m = n\}$  является существенно неоднозначным. Доказательство этого факта можно найти в [АхоУль, с. 234-236].

**Упражнение 7.2.12.** Однозначна ли контекстно-свободная грамматика

$$\begin{aligned} K &\rightarrow a, \\ K &\rightarrow b, \\ K &\rightarrow K + K, \\ K &\rightarrow K - K \end{aligned}$$

с алфавитом  $\Sigma = \{a, b, +, -\}$ ?

**Упражнение 7.2.13.** Однозначна ли контекстно-свободная грамматика

$$\begin{aligned} K &\rightarrow a, \\ K &\rightarrow b, \\ K &\rightarrow +KK, \\ K &\rightarrow -KK \end{aligned}$$

с алфавитом  $\Sigma = \{a, b, +, -\}$ ?

**Упражнение 7.2.14.** Однозначна ли контекстно-свободная грамматика

$$K \rightarrow a,$$

$$K \rightarrow b,$$

$$K \rightarrow (K+K),$$

$$K \rightarrow (K-K)$$

с алфавитом  $\Sigma = \{a, b, +, -, ), (\}$  ?

**Упражнение 7.2.15.** Однозначна ли контекстно-свободная грамматика

$$E \rightarrow EcE,$$

$$E \rightarrow EdE,$$

$$E \rightarrow aEb,$$

$$E \rightarrow i?$$

**Упражнение 7.2.16.** Найти однозначную контекстно-свободную грамматику, эквивалентную грамматике

$$S \rightarrow aS,$$

$$S \rightarrow aSb,$$

$$S \rightarrow c.$$

**Упражнение 7.2.17.** Найти однозначную контекстно-свободную грамматику, эквивалентную грамматике

$$S \rightarrow aSaaaa,$$

$$S \rightarrow aSaa,$$

$$S \rightarrow aaSa,$$

$$S \rightarrow b.$$

### 7.3\*. Однозначные праволинейные грамматики

**Теорема 7.3.1.** Каждый праволинейный язык порождается некоторой однозначной праволинейной грамматикой. Другими словами, ни один праволинейный язык не является существенно неоднозначным.

*Доказательство.* Согласно теоремам 2.4.3 и 2.7.1 исходный язык распознается некоторым детерминированным конечным автоматом. Применив к нему конструкцию из доказательства теоремы 2.4.1, получим однозначную праволинейную грамматику.

**Замечание 7.3.2.** Этот факт можно получить также из более общей теоремы 12.2.6 с учетом теоремы 12.2.1.

**Упражнение 7.3.3.** Найти однозначную праволинейную грамматику, порождающую язык  $a^* ((a+b) (a+b))^* b^*$ .

**Упражнение 7.3.4.** Найти однозначную праволинейную грамматику, эквивалентную грамматике

$$S \rightarrow TaaaT,$$

$$S \rightarrow TbbbT,$$

$$T \rightarrow aT,$$

$$T \rightarrow bT,$$

$$T \rightarrow \varepsilon.$$

**Упражнение 7.3.5.** Найти однозначную праволинейную грамматику, эквивалентную грамматике

$$S \rightarrow aS, \quad T \rightarrow abZ,$$

$$S \rightarrow bS, \quad T \rightarrow bbZ,$$

$$S \rightarrow aaT, \quad Z \rightarrow aZ,$$

$$S \rightarrow abT, \quad Z \rightarrow bZ,$$

$$Z \rightarrow \varepsilon.$$

**Упражнение 7.3.6.** Существует ли праволинейная грамматика в нормальной форме с  $n$  вспомогательными символами, не эквивалентная ни одной однозначной праволинейной грамматике с количеством вспомогательных символов  $2^n - 1$ ?

## 7.4. Языки Дика и Лукасевича

**Определение 7.4.1.** Языком Дика (Dyck language) над  $2n$  буквами называется контекстно-свободный язык над алфавитом  $\{a_1, b_1, a_2, b_2, \dots, a_n, b_n\}$ , порождаемый грамматикой

$$S \rightarrow \varepsilon$$

$$S \rightarrow a_1 S b_1 S$$

$$\vdots$$

$$S \rightarrow a_n S b_n S.$$

**Замечание 7.4.2.** Словами этого языка являются последовательности правильно вложенных скобок  $n$  типов (если считать символ  $a_i$  левой скобкой, а символ  $b_i$  соответствующей правой скобкой).

**Теорема 7.4.3.** Слово  $w$  над алфавитом  $\{a, b\}$  выводится в грамматике

$$S \rightarrow \varepsilon,$$

$$S \rightarrow aSbS$$

тогда и только тогда, когда

$$|w|_a = |w|_b$$

и для всех слов  $x \sqsubset w$  выполняется неравенство

$$|x|_a \geq |x|_b.$$

**Теорема 7.4.4.** При любом положительном целом  $n$  грамматика из определения 7.4.1 является однозначной.

**Определение 7.4.5.** Языком Лукасевича (Lukasiewicz language) над  $n + 1$  буквами называется контекстно-свободный язык над алфавитом  $\{a_0, a_1, \dots, a_n\}$ , порождаемый грамматикой

$$\begin{aligned}
S &\rightarrow a_0, \\
S &\rightarrow a_1 S, \\
S &\rightarrow a_2 S S, \\
&\vdots \\
S &\rightarrow a_n S^n.
\end{aligned}$$

**Теорема 7.4.6.** Слово  $w$  над алфавитом  $\{a_0, a_1, \dots, a_n\}$  принадлежит языку Лукасевича над  $n + 1$  буквами тогда и только тогда, когда

$$\sum_{i=0}^n (i-1) |w|_{a_i} = -1$$

и для всех слов  $x \sqsubset w$ , кроме  $x = w$ , выполняется неравенство

$$\sum_{i=0}^n (i-1) |x|_{a_i} \geq 0.$$

**Теорема 7.4.7.** При любом  $n \in \mathbb{N}$  грамматика из определения 7.4.5 является однозначной.

**Упражнение 7.4.8.** Принадлежит ли слово  $aababb$  языку, порождаемому грамматикой

$$\begin{aligned}
S &\rightarrow a S b S, \\
S &\rightarrow \varepsilon?
\end{aligned}$$

**Упражнение 7.4.9.** Принадлежит ли слово  $cacbcbbacacccaaaaacabcaa$  языку, порождаемому грамматикой

$$\begin{aligned}
S &\rightarrow a, \\
S &\rightarrow b S, \\
S &\rightarrow c S S?
\end{aligned}$$

**Упражнение 7.4.10.** Эквивалентны ли грамматика

$$\begin{aligned}
T &\rightarrow a, \\
T &\rightarrow T T, \\
T &\rightarrow T T b?
\end{aligned}$$

и грамматика

$$\begin{aligned}
R &\rightarrow a, \\
R &\rightarrow R a, \\
R &\rightarrow R R b?
\end{aligned}$$

**Упражнение 7.4.11.** Описать язык, порождаемый грамматикой

$$\begin{aligned}
F &\rightarrow ab, \\
F &\rightarrow a F b, \\
F &\rightarrow F F.
\end{aligned}$$

**Упражнение 7.4.12.** Описать язык, порождаемый грамматикой

$$F \rightarrow a,$$

$$F \rightarrow bF,$$

$$F \rightarrow cFF,$$

$$F \rightarrow dFFFF.$$

**Упражнение 7.4.13.** Найти однозначную контекстно-свободную грамматику, порождающую язык  $\{w \in \{a, b\}^* \mid |w|_a \geq |w|_b\}$ .

© 2003-2008 INTUIT.ru. Все права защищены.



## Математическая теория формальных языков

### 8. Лекция: Нормальные формы контекстно-свободных грамматик: версия для печати и PDA

В данной лекции рассматривается доказательство того, что каждая контекстно-свободная грамматика эквивалентна некоторой контекстно-свободной грамматике специального вида, а именно грамматике в нормальной форме Хомского. Этот факт используется в доказательствах многих теорем о контекстно-свободных языках и является очень важным. Приведены также практические примеры и самостоятельные упражнения

Основная цель этой лекции - доказать, что каждая контекстно-свободная грамматика эквивалентна некоторой контекстно-свободной грамматике специального вида, а именно грамматике в нормальной форме Хомского (раздел 8.3). Этот факт используется дальше в доказательствах многих теорем о контекстно-свободных языках. Два вспомогательных результата, на которые опирается приведение грамматик к нормальной форме Хомского, выделены в отдельные разделы в начале лекции.

В конце лекции доказывается, что каждую контекстно-свободную грамматику можно также привести к нормальной форме Грейбах.

#### 8.1. Устранение бесполезных символов

**Определение 8.1.1.** Пусть дана порождающая грамматика  $G = \langle N, \Sigma, P, S \rangle$ . Символ  $A \in N$  называется *полезным* (useful), если существуют такие слова  $\alpha \in (N \cup \Sigma)^*$ ,  $\beta \in (N \cup \Sigma)^*$  и  $w \in \Sigma^*$ , что  $S \xRightarrow{*} \alpha A \beta$  и  $\alpha A \beta \xRightarrow{*} w$ . Символ  $A \in N$  называется *бесполезным* (useless), если он не является полезным. Символ  $A \in N$  называется *порождающим* (generating), если существует такое слово  $w \in \Sigma^*$ , что  $A \xRightarrow{*} w$ . Символ  $A \in N$  называется *достижимым* если существуют такие слова  $\alpha \in (N \cup \Sigma)^*$  и  $\beta \in (N \cup \Sigma)^*$ , что  $S \xRightarrow{*} \alpha A \beta$ .

**Лемма 8.1.2.** Пусть дана контекстно-свободная грамматика  $G = \langle N, \Sigma, P, S \rangle$ , в которой все символы из  $N$  являются порождающими. Пусть  $N'$  - множество всех достижимых символов грамматики  $G$ , а  $P'$  - множество тех правил из  $P$ , которые не содержат ни одного символа из множества  $N - N'$ . Тогда в контекстно-свободной грамматике  $\langle N', \Sigma, P', S \rangle$  все символы из  $N'$  являются порождающими. \end{lemma}

**Теорема 8.1.3.** Пусть дана контекстно-свободная грамматика  $G = \langle N, \Sigma, P, S \rangle$  и  $L(G) \neq \emptyset$ . Тогда существуют такие множества  $N' \subseteq N$  и  $P' \subseteq P$ , что в контекстно-свободной грамматике  $\langle N', \Sigma, P', S \rangle$  нет бесполезных символов и она эквивалентна исходной грамматике.

*Доказательство.* На первом этапе удалим все непорождающие символы (удалим также каждое правило, содержащее хотя бы один такой символ). На втором этапе из полученной грамматики удалим все недостижимые символы (и правила, их содержащие). Согласно 8.1.2 на втором этапе ни один порождающий символ не может стать непорождающим.

**Пример 8.1.4.** Рассмотрим контекстно-свободную грамматику  $G$  с правилами

$S \rightarrow UX,$	$W \rightarrow YZY,$	$Y \rightarrow YY$
$S \rightarrow VZ,$	$W \rightarrow aab,$	$Y \rightarrow aU$
$T \rightarrow aa,$	$X \rightarrow Xa,$	$Y \rightarrow \varepsilon$
$T \rightarrow bb,$	$X \rightarrow Xb,$	$Z \rightarrow W$
$U \rightarrow aUa,$	$X \rightarrow \varepsilon,$	$Z \rightarrow b.$
$U \rightarrow bUb,$		
$V \rightarrow aTb,$		
$V \rightarrow bTa,$		

Удалив четыре правила, содержащие непорождающий символ  $U$ , получим грамматику  $G_1$ .

В ней символ  $x$  является недостижимым. Удалив три правила, содержащие  $x$ , получим грамматику  $G_2$  с правилами

$S \rightarrow VZ,$	$W \rightarrow YZY,$	$Y \rightarrow YY,$
$T \rightarrow aa,$	$W \rightarrow aab,$	$Y \rightarrow \varepsilon,$
$T \rightarrow bb,$		$Z \rightarrow W,$
$V \rightarrow aTb,$		$Z \rightarrow b.$
$V \rightarrow bTa,$		

Очевидно, что  $L(G) = L(G_2)$  и грамматика  $G_2$  не содержит бесполезных символов.

**Упражнение 8.1.5.** Найти контекстно-свободную грамматику без бесполезных символов, эквивалентную грамматике

$S \rightarrow SRT,$   
 $S \rightarrow c,$   
 $R \rightarrow aRa,$   
 $R \rightarrow b,$   
 $T \rightarrow aT.$

## 8.2. Устранение эпсилон-правил

**Теорема 8.2.1.** Пусть язык  $L$  является контекстно-свободным. Тогда язык  $L - \{\varepsilon\}$  порождается некоторой контекстно-свободной грамматикой без  $\varepsilon$ -правил.

*Доказательство.* Пусть дана контекстно-свободная грамматика  $G = \langle N, \Sigma, P, S \rangle$ , порождающая язык  $L$ . Проведем серию преобразований множества  $P$ .

Если для каких-то  $A \in N$ ,  $B \in N$ ,  $\alpha \in (N \cup \Sigma)^*$  и  $\beta \in (N \cup \Sigma)^*$  множество  $P$  содержит правила  $B \rightarrow \alpha A \beta$  и  $A \rightarrow \varepsilon$ , но не содержит правила  $B \rightarrow \alpha \beta$ , то добавим это правило в  $P$ . Повторяем эту процедуру, пока возможно.

Теперь исключим из множества  $P$  все правила вида  $A \rightarrow \varepsilon$ . Полученная грамматика порождает язык  $L - \{\varepsilon\}$ .

**Пример 8.2.2.** Рассмотрим язык  $L$ , порождаемый грамматикой

$$S \rightarrow \varepsilon,$$

$$S \rightarrow aSbS.$$

Язык  $L - \{\varepsilon\}$  порождается грамматикой

$$S \rightarrow aSbS,$$

$$S \rightarrow abS,$$

$$S \rightarrow aSb,$$

$$S \rightarrow ab.$$

**Упражнение 8.2.3.** Найти контекстно-свободную грамматику без  $\varepsilon$ -правил, эквивалентную грамматике

$$D \rightarrow Sc,$$

$$S \rightarrow SS,$$

$$S \rightarrow \varepsilon,$$

$$S \rightarrow aSb.$$

### 8.3. Нормальная форма Хомского

**Определение 8.3.1.** Грамматика в нормальной форме Хомского (грамматика в бинарной нормальной форме, квадратичная грамматика, grammar in Chomsky normal form) - контекстно-свободная грамматика  $\langle N, \Sigma, P, S \rangle$ , в которой каждое правило имеет один из следующих трех видов:  $S \rightarrow \varepsilon$ ,  $A \rightarrow a$ ,  $A \rightarrow BC$ , где  $A \in N$ ,  $B \in N - \{S\}$ ,  $C \in N - \{S\}$ ,  $a \in \Sigma$ .

**Пример 8.3.2.** Грамматика

$$S \rightarrow RR, \quad A \rightarrow a,$$

$$S \rightarrow AB, \quad B \rightarrow RB,$$

$$R \rightarrow RR, \quad B \rightarrow b$$

$$R \rightarrow AB,$$

является грамматикой в нормальной форме Хомского.

**Теорема 8.3.3.** Каждая контекстно-свободная грамматика эквивалентна некоторой грамматике в нормальной форме Хомского.

**Доказательство.** Пусть дана контекстно-свободная грамматика  $G = \langle N, \Sigma, P, S \rangle$ . Проведем ряд преобразований этой грамматики так, что порождаемый ею язык остается неизменным.

Если правая часть какого-нибудь правила содержит символ  $S$ , то заменим грамматику

$$\langle N, \Sigma, P, S \rangle \text{ на грамматику } \langle N \cup \{S_0\}, \Sigma, P \cup \{S_0 \rightarrow S\}, S_0 \rangle,$$

где  $S_0$  - новый символ, не принадлежащий множеству  $N \cup \Sigma$ .

Заменим во всех правилах каждый терминальный символ  $a$  на новый нетерминальный символ  $T_a$  и добавим к множеству  $P$  правила  $T_a \rightarrow a$  для всех  $a \in \Sigma$ .

Устраним правила вида  $A \rightarrow \alpha$ , где  $|\alpha| > 2$ , заменив каждое из них на ряд более

коротких правил (при этом добавляются новые нетерминальные символы).

Теперь устраним все правила вида  $A \rightarrow \varepsilon$ , где  $A$  не является начальным символом. Это можно сделать так же, как в доказательстве теоремы 8.2.1.

Если для каких-то  $A \in N$ ,  $B \in N$  и  $\alpha \in (N \cup \Sigma)^*$  множество  $P$  содержит правила  $A \rightarrow B$  и  $B \rightarrow \alpha$ , но не содержит правила  $A \rightarrow \alpha$ , то добавим это правило в  $P$ . Повторяем эту процедуру, пока возможно. После этого исключим из множества  $P$  все правила вида  $A \rightarrow B$ .

**Пример 8.3.4.** Грамматика

$$S \rightarrow \varepsilon,$$

$$S \rightarrow aUbU,$$

$$U \rightarrow S,$$

$$U \rightarrow ba$$

эквивалентна следующей грамматике в нормальной форме Хомского:

$$S_0 \rightarrow \varepsilon,$$

$$C \rightarrow BU,$$

$$S_0 \rightarrow AD,$$

$$C \rightarrow b,$$

$$D \rightarrow UC,$$

$$U \rightarrow BA,$$

$$D \rightarrow BU,$$

$$U \rightarrow AD,$$

$$D \rightarrow b,$$

$$A \rightarrow a,$$

$$B \rightarrow b.$$

**Теорема 8.3.5.** Если контекстно-свободный язык не содержит пустого слова, то он порождается некоторой грамматикой, в которой каждое правило имеет один из следующих двух видов:  $A \rightarrow a$ ,  $A \rightarrow BC$ , где  $A \in N$ ,  $B \in N - \{S\}$ ,  $C \in N - \{S\}$ ,  $a \in \Sigma$ .

**Упражнение 8.3.6.** Найти контекстно-свободную грамматику в нормальной форме Хомского, эквивалентную грамматике

$$F \rightarrow ab,$$

$$F \rightarrow aFb,$$

$$F \rightarrow FF.$$

## 8.4\*. Нормальная форма Грейбах

**Определение 8.4.1.** Грамматика в нормальной форме Грейбах (grammar in Greibach normal form) - контекстно-свободная грамматика  $\langle N, \Sigma, P, S \rangle$ , в которой каждое правило имеет один из следующих четырех видов:  $A \rightarrow \varepsilon$ ,  $A \rightarrow a$ ,  $A \rightarrow aB$ ,  $A \rightarrow aBC$ , где  $A \in N$ ,  $B \in N$ ,  $C \in N$ ,  $a \in \Sigma$ .

**Пример 8.4.2.** Грамматика

$$S \rightarrow aST,$$

$$S \rightarrow aT,$$

$$T \rightarrow bS,$$

$$T \rightarrow b$$

является грамматикой в нормальной форме Грейбах.

**Замечание 8.4.3.** Некоторые авторы разрешают в грамматиках в нормальной форме Грейбах использовать также правила вида  $A \rightarrow a\beta$ , где  $A \in N$ ,  $a \in \Sigma$ ,  $\beta \in N^*$  (в определении 8.4.1 разрешены, только если  $|\beta| \leq 2$ ).

**Теорема 8.4.4.** Каждая контекстно-свободная грамматика эквивалентна некоторой грамматике в нормальной форме Грейбах.

*Доказательство.* Докажем теорему для контекстно-свободных языков, не содержащих пустого слова. Согласно теореме 8.3.5 исходный язык порождается некоторой

грамматикой  $G = \langle N, \Sigma, P, S \rangle$ , в которой каждое правило имеет вид  $A \rightarrow a$  или  $A \rightarrow BC$ , где  $A \in N$ ,  $B \in N - \{S\}$ ,  $C \in N - \{S\}$ ,  $a \in \Sigma$ .

Введем  $|N|^2$  новых вспомогательных символов, соответствующих упорядоченным парам из множества  $N \times N$ . Новый символ, соответствующий паре  $\langle A, B \rangle$ , будем обозначать  $(A \setminus B)$ . Построим грамматику "почти в нормальной форме Грейбах"

$$\begin{aligned} \bar{G} = \langle \bar{N}, \Sigma, \bar{P}, S \rangle, \text{ положив } \bar{N} = N \cup \{(A \setminus B) \mid A \in N, B \in N\} \text{ и} \\ \bar{P} = \{(A \setminus A) \rightarrow \varepsilon \mid A \in N\} \cup \\ \cup \{(C \setminus E) \rightarrow A(A \setminus D)(B \setminus E) \mid (B \rightarrow CD) \in P, A \in N, E \in N\} \cup \\ \cup \{A \rightarrow a \mid (A \rightarrow a) \in P\} \cup \\ \cup \{S \rightarrow a(A \setminus S) \mid (A \rightarrow a) \in P\}. \end{aligned}$$

Если в этой грамматике заменить

$$\begin{aligned} \{(C \setminus E) \rightarrow A(A \setminus D)(B \setminus E) \mid (B \rightarrow CD) \in P, A \in N, E \in N\} \cup \\ \cup \{A \rightarrow a \mid (A \rightarrow a) \in P\} \end{aligned}$$

на

$$\{(C \setminus E) \rightarrow a(A \setminus D)(B \setminus E) \mid (B \rightarrow CD) \in P, (A \rightarrow a) \in P, E \in N\},$$

получим эквивалентную ей грамматику в нормальной форме Грейбах. Осталось лишь доказать, что  $L(G) = L(\bar{G})$ .

Сначала проверим индукцией по длине слова  $\gamma \in N^*$ , что если  $F \xrightarrow{*}_G C\gamma$ , то

$$\begin{aligned} (C \setminus E) \xrightarrow{*}_G \gamma(F \setminus E) \text{ для любого } E \in N. \text{ Чтобы провести шаг индукции, допустим, что} \\ (B \rightarrow CD) \in P \text{ и} \\ F \xrightarrow{*}_G B\beta \Rightarrow CD\beta \xrightarrow{*}_G CA\alpha\beta, \end{aligned}$$

где  $D \xrightarrow{*}_G A\alpha$  и  $\gamma = A\alpha\beta$ . По предположению индукции имеем  $(B \setminus E) \xrightarrow{*}_G \beta(F \setminus E)$  и  $(A \setminus D) \xrightarrow{*}_G \alpha(D \setminus D)$ . Подключая эти выводы к правилу  $(C \setminus E) \Rightarrow A(A \setminus D)(B \setminus E)$  и используя  $(D \setminus D) \Rightarrow \varepsilon$ , получаем искомый вывод  $(C \setminus E) \xrightarrow{*}_G A\alpha\beta(F \setminus E)$ .

Докажем теперь, что для любого  $\gamma \in N^*$  равносильны утверждения  $F \xrightarrow{*}_G C\gamma$  и

$$(C \setminus F) \xrightarrow{*}_G \gamma. \text{ В одну сторону это следует из только что доказанного. Доказательство}$$

того, что если  $(C \setminus F) \xrightarrow{*}_G \gamma$ , то  $F \xrightarrow{*}_G C\gamma$ , проведем индукцией по длине слова

$\gamma \in N^*$ . Чтобы провести шаг индукции, допустим, что  $(B \rightarrow CD) \in P$ ,  
 $(C \setminus F) \xRightarrow{*}_G A(A \setminus D)(B \setminus F)$ ,  $(A \setminus D) \xRightarrow{*}_G \alpha$ ,  $(B \setminus F) \xRightarrow{*}_G \beta$  и  $\gamma = A\alpha\beta$ . По  
 предположению индукции  $D \xRightarrow{*}_G A\alpha$  и  $F \xRightarrow{*}_G B\beta$ . Получаем искомый вывод  
 $F \xRightarrow{*}_G B\beta \xRightarrow{*}_G CD\beta \xRightarrow{*}_G CA\alpha\beta$ .

Теперь убедимся, что  $L(G) = L(\overline{G})$ . Рассмотрим произвольное слово  $a_0a_1 \dots a_m$ , где  
 $m \geq 0$  и  $a_i \in \Sigma$  для всех  $i \leq m$ . Пусть

$$S \xRightarrow{*}_G A_0A_1 \dots A_m \xRightarrow{*}_G a_0a_1 \dots a_m,$$

где  $A_i \in N$  для всех  $i \leq m$ . Тогда

$$S \xRightarrow{*}_G a_0(A_0 \setminus S) \xRightarrow{*}_G a_0A_1 \dots A_m \xRightarrow{*}_G a_0a_1 \dots a_m.$$

Обратно, пусть

$$S \xRightarrow{*}_G a_0(A_0 \setminus S) \xRightarrow{*}_G a_0A_1 \dots A_m \xRightarrow{*}_G a_0a_1 \dots a_m,$$

где  $A_i \in N$  для всех  $i \leq m$ . Тогда

$$S \xRightarrow{*}_G A_0A_1 \dots A_m \xRightarrow{*}_G a_0a_1 \dots a_m.$$

**Пример 8.4.5.** Грамматика

$$\begin{array}{ll} S \rightarrow RT, & U \rightarrow VT, \\ T \rightarrow b, & V \rightarrow RT, \\ T \rightarrow UR, & R \rightarrow a \end{array}$$

эквивалентна следующей грамматике в нормальной форме Грейбах:

$$\begin{array}{ll} S \rightarrow aC, & E \rightarrow aDF, \\ C \rightarrow aD, & E \rightarrow bF, \\ C \rightarrow b, & F \rightarrow a. \\ D \rightarrow aDE, & \\ D \rightarrow bE, & \end{array}$$

Здесь C, D, E и F соответствуют символам  $(A \setminus S)$ ,  $(A \setminus T)$ ,  $(V \setminus T)$  и  $(U \setminus T)$  из доказательства теоремы 8.4.4 (удален 21 бесполезный символ).

**Теорема 8.4.6.** Пусть язык  $L$  контекстно-свободный. Тогда язык  $L - \{\epsilon\}$  порождается некоторой грамматикой в нормальной форме Грейбах без  $\epsilon$ -правил.

**Пример 8.4.7.** Грамматика

$$\begin{array}{l} S \rightarrow aR, \\ R \rightarrow bRT, \\ R \rightarrow \epsilon, \\ T \rightarrow cSR, \\ T \rightarrow \epsilon \end{array}$$

эквивалентна следующей грамматике в нормальной форме Грейбах без  $\epsilon$ -правил:

$$\begin{array}{lll}
 S \rightarrow aR, & R \rightarrow bRT, & T \rightarrow cSR, \\
 S \rightarrow a, & R \rightarrow bT, & T \rightarrow cS. \\
 & R \rightarrow bR, & \\
 & R \rightarrow b, & 
 \end{array}$$

**Упражнение 8.4.8.** Найти контекстно-свободную грамматику в нормальной форме Грейбах, эквивалентную грамматике

$$\begin{array}{l}
 F \rightarrow ab, \\
 F \rightarrow aFb, \\
 F \rightarrow FF.
 \end{array}$$

**Упражнение 8.4.9.** Найти контекстно-свободную грамматику в нормальной форме Грейбах, эквивалентную грамматике

$$\begin{array}{l}
 S \rightarrow AB, \\
 B \rightarrow AB, \\
 A \rightarrow BB, \\
 A \rightarrow a, \\
 B \rightarrow b.
 \end{array}$$

**Упражнение 8.4.10.** Найти контекстно-свободную грамматику в нормальной форме Грейбах, эквивалентную грамматике

$$\begin{array}{l}
 K \rightarrow Fb, \\
 F \rightarrow \varepsilon, \\
 F \rightarrow aFbF.
 \end{array}$$

**Упражнение 8.4.11.** Найти контекстно-свободную грамматику в нормальной форме Грейбах, эквивалентную грамматике

$$\begin{array}{l}
 S \rightarrow aT, \\
 T \rightarrow aTTa, \\
 T \rightarrow b.
 \end{array}$$

## Математическая теория формальных языков

### 9. Лекция: Основные свойства контекстно-свободных языков: версия для печати и PDA

В данной лекции рассматриваются основные свойства контекстно-свободных языков, приведены некоторые теоремы для класса линейных языков, а также доказательство того, что пересечение контекстно-свободного языка с автоматным языком является контекстно-свободным. Также приводятся практические примеры и упражнения для самостоятельной проработки

В [лекции 2](#) были доказаны лемма о разрастании и свойства замкнутости класса автоматных языков. Некоторые из этих теорем имеют аналоги для класса контекстно-свободных языков (разделы 9.1 и 9.4), но этот класс не замкнут относительно дополнения и пересечения (раздел 9.5).

Лемма о разрастании для контекстно-свободных языков формализует явление "периодичности" в этих языках. Для полноты картины в разделах 9.2\* и 9.3 приведены некоторые аналогичные теоремы для класса линейных языков, хотя ни в теории, ни в практических приложениях класс линейных языков значительной роли не играет.

В разделе 9.6 доказывается, что пересечение контекстно-свободного языка с автоматным языком является контекстно-свободным. В сочетании с леммой о разрастании этот факт дает удобное средство, позволяющее во многих задачах доказать, что заданный язык не является контекстно-свободным. Еще одно необходимое условие контекстной свободности сформулировано в разделе 9.7\*.

#### 9.1. Лемма о разрастании для контекстно-свободных языков

**Лемма 9.1.1 (pumping lemma, лемма о разрастании, лемма о накачке, лемма-насос)** Пусть  $L$  - контекстно-свободный язык над алфавитом  $\Sigma$ . Тогда найдется такое положительное целое число  $p$ , что для любого слова  $w \in L$  длины не меньше  $p$  можно подобрать слова  $u, v, x, y, z \in \Sigma^*$ , для которых верно  $uvxyz = w$ ,  $vy \neq \epsilon$  (то есть  $v \neq \epsilon$  или  $y \neq \epsilon$ ),  $|vxy| \leq p$  и  $uv^i xy^i z \in L$  для всех  $i \in \mathbb{N}$ .

*Доказательство.* Пусть язык  $L$  порождается грамматикой в нормальной форме Хомского

$G = \langle N, \Sigma, P, S \rangle$ . Индукцией по  $k$  легко доказать, что для любого дерева вывода в грамматике  $G$  длина кроны дерева не превышает  $2^{k-2}$ , где  $k$  - количество вершин в самом длинном пути, начинающемся в корне дерева и заканчивающемся в некоторой вершине, помеченной символом из  $\Sigma$ .

Положим  $p = 2^{|N|}$ . Пусть  $w \in L$  и  $|w| \geq p$ . Зафиксируем некоторое дерево вывода с кроной  $w$  в грамматике  $G$ . Рассмотрим самый длинный путь в этом дереве. Этот путь содержит не менее  $|N| + 2$  вершин. Среди них найдутся две вершины с одинаковыми метками, причем их можно выбрать среди последних  $|N| + 2$  вершин рассматриваемого пути. Выберем слова  $u, v, x, y$  и  $z$  так, что  $uvxyz = w$ , поддерево с корнем в одной из найденных вершин имеет крону  $x$  и поддерево с корнем в другой найденной вершине имеет крону  $vxy$ .

Из того что  $G$  - грамматика в нормальной форме Хомского, заключаем, что  $vxy \neq x$ .

Неравенство  $|vxy| \leq 2^{|N|}$  следует из того, что самый длинный путь в соответствующем слову  $vxy$  поддереве содержит не более  $|N| + 2$  вершин. Для каждого  $i \in \mathbb{N}$  можно построить дерево вывода с кроной  $uv^i xy^i z$ , комбинируя части исходного



дерева вывода.

**Пример 9.1.2.** Рассмотрим язык  $L = \{a^n b^n c^n \mid n \geq 0\}$  над алфавитом  $\{a, b, c\}$ . Утверждение леммы 9.1.1 не выполняется ни для какого натурального числа  $p$ .

Действительно, если  $uvxyz = a^p b^p c^p$ ,  $|vy| > 0$  и  $|vxy| \leq p$ , то  $|vy|_a = 0$  или  $|vy|_c = 0$ . Следовательно,  $|uvvxyyz|_a = p$  или  $|uvvxyyz|_c = p$ . Так как  $|uvvxyyz| > 3p$ , то

$uvvxyyz \notin L$ . Из этого можно заключить, что язык  $L$  не является контекстно-свободным.

**Теорема 9.1.3.** Каждый контекстно-свободный язык над однобуквенным алфавитом является автоматным.

*Доказательство.* Пусть дан контекстно-свободный язык  $L$  над алфавитом  $\{a\}$ . Согласно лемме 9.1.1 найдется такое натуральное число  $p$ , что множество  $\{k \in \mathbb{N} \mid a^k \in L\}$  является объединением некоторого семейства арифметических прогрессий, причем у каждой прогрессии первый член и шаг не больше числа  $p$ . Так как существует лишь конечное число прогрессий натуральных чисел с таким ограничением, рассматриваемое семейство конечно. Следовательно, язык  $L$  является автоматным (используем пример 2.1.18).

**Упражнение 9.1.4.** Является ли контекстно-свободным язык  $\{ww \mid w \in \{a, b\}^*\}$ ?

**Упражнение 9.1.5.** Является ли контекстно-свободным язык

$$\{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}?$$

**Упражнение 9.1.6.** Является ли контекстно-свободным язык

$$\{a^{k+m} b^m a^k b^m \mid k \geq 0, m \geq 0\}?$$

**Упражнение 9.1.7.** Является ли контекстно-свободным язык  $\{a^m \mid m \text{ простое}\}$ ?

**Упражнение 9.1.8.** Является ли контекстно-свободным язык

$$\{a^n b^n a^n \mid n \geq 0\} \cup \{a^k b^l a^m \mid k \geq 0, l \geq 0, m \leq 52\}?$$

**Упражнение 9.1.9.** Является ли контекстно-свободным язык

$$\{w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c\}?$$

**Упражнение 9.1.10.** Является ли контекстно-свободным язык

$$\{a^n b w b a^m \mid w \in \{a, b\}^*, |w|_a = n + m\}?$$

**Упражнение 9.1.11.** Является ли контекстно-свободным язык

$$\{uvw \mid u \in \{a, b\}^*, v \in \{c, d\}^*, w \in \{a, b\}^*, |u|_a = |w|_b, |v|_c = |v|_d\}?$$

**Упражнение 9.1.12.** Является ли контекстно-свободным язык

$$\{uvw \mid u \in \{a, b\}^*, v \in \{c, d\}^*, w \in \{a, b\}^*, |u|_a = |v|_c, |v|_d = |w|_b\}?$$

**Упражнение 9.1.13.** Является ли контекстно-свободным язык  $\{a^k b^m c^n \mid k < \max(m, n)\}$ ?

**Упражнение 9.1.14.** Является ли контекстно-свободным язык  $\{a^k b^m c^n \mid k > \max(m, n)\}$ ?

$(m, n) \}$ ?

**Упражнение 9.1.15.** Является ли контекстно-свободным язык

$$\{a^n b^n a^n b^n \mid n \geq 0\} \cup \{a^{2k} b^{2k} a^m b^m \mid k \geq 0, m \geq 0\} \cup \{a^k b^{2l} b a a^{2l} b^k \mid k \geq 0, m \geq 0\}?$$

**Упражнение 9.1.16.** Какому классу принадлежит язык, порождаемый грамматикой

$$S \rightarrow TcV,$$

$$T \rightarrow bV,$$

$$V \rightarrow aVa,$$

$$V \rightarrow \epsilon,$$

$$ac \rightarrow a?$$

**Упражнение 9.1.17.** Существуют ли такие контекстно-свободные языки  $L_1 \subseteq \{a, b\}^*$  и  $L_2 \subseteq \{b, c\}^*$ , что язык  $L_1 - L_2$  не является контекстно-свободным?

## 9.2\*. Лемма о разрастании для линейных языков

**Определение 9.2.1.** Линейная грамматика в нормальной форме - это такая линейная грамматика, в которой каждое правило имеет вид  $A \rightarrow \epsilon$ ,  $A \rightarrow a$ ,  $A \rightarrow aB$  или  $A \rightarrow Ba$ , где  $A \in N$ ,  $B \in N$ ,  $a \in \Sigma$ .

**Теорема 9.2.2.** Каждая линейная грамматика эквивалентна некоторой линейной грамматике в нормальной форме.

**Теорема 9.2.3.** Если линейный язык не содержит пустого слова, то он порождается некоторой линейной грамматикой в нормальной форме без  $\epsilon$ -правил.

**Теорема 9.2.4.** Язык  $L$  является линейным тогда и только тогда, когда язык  $L - \{\epsilon\}$  является линейным.  $\end{theorem}$

**Лемма 9.2.5.** Пусть  $L$  - линейный язык над алфавитом  $\Sigma$ . Тогда найдется такое положительное целое число  $p$ , что для любого слова  $w \in L$  длины не меньше  $p$  можно подобрать слова  $u, v, x, y, z \in \Sigma^*$ , для которых верно  $uvxyz = w$ ,  $vy \neq \epsilon$  (то есть  $v \neq \epsilon$  или  $y \neq \epsilon$ ),  $|uv| + |yz| \leq p$  и  $uv^i xy^i z \in L$  для всех  $i \in \mathbb{N}$ .

**Доказательство.** Пусть язык  $L - \{\epsilon\}$  порождается линейной грамматикой в нормальной форме  $G = \langle N, \Sigma, P, S \rangle$  без  $\epsilon$ -правил. Положим  $p = |N| + 1$ . Пусть  $w \in L$  и  $|w| \geq p$ . Зафиксируем некоторое дерево вывода слова  $w$  в грамматике  $G$ . Рассмотрим самый длинный путь в этом дереве (он начинается с корня и заканчивается некоторым листом, помеченным символом из  $\Sigma$ ). Этот путь содержит не менее  $|N| + 1$  вершин, помеченных элементами  $N$ . Среди первых  $|N| + 1$  вершин рассматриваемого пути найдутся две вершины с одинаковыми метками. Выберем слова  $u, v, x, y$  и  $z$  так, что  $uvxyz = w$ , поддерево с корнем в одной из найденных вершин имеет крону  $x$  и поддерево с корнем в другой найденной вершине имеет крону  $vyx$ .

**Пример 9.2.6.** Рассмотрим язык  $L = \{a^m b^m a^n b^n \mid m \geq 0, n \geq 0\}$  над алфавитом

$\{a, b\}$ . Утверждение леммы 9.2.5 не выполняется ни для какого натурального числа  $p$ . Следовательно, язык  $L$  не является линейным.

**Упражнение 9.2.7.** Какому классу принадлежит язык  $\{a^n b^n (cd)^m \mid n \geq 1, m \geq 1\}$ ?

**Упражнение 9.2.8.** Какому классу принадлежит язык  $\{a^k b^l c^m d^n \mid k < n, l > m\}$ ?

**Упражнение 9.2.9.** Какому классу принадлежит язык  $\{a^k b^l c^m a^n b^n c^k \mid k \geq 1, l < m, n \geq 3\}$ ?

### 9.3. Свойства замкнутости класса линейных языков

**Пример 9.3.1.** Пусть  $\Sigma = \{a, b, c\}$ . Язык  $\{ucuc^R \mid u \in \{a, b\}^*\}$  является линейным, так как он порождается грамматикой

$$S \rightarrow aSa,$$

$$S \rightarrow bSb,$$

$$S \rightarrow c.$$

**Пример 9.3.2.** Рассмотрим алфавит  $\Sigma = \{a, b, c\}$ . Язык

$$\{a^m b^n c^k \mid 0 \leq m < n, k \geq 0\}$$

является линейным, так как он порождается грамматикой

$$S \rightarrow Sc,$$

$$T \rightarrow aTb,$$

$$S \rightarrow T,$$

$$T \rightarrow Tb,$$

$$T \rightarrow b.$$

**Теорема 9.3.3.** Если  $L_1$  и  $L_2$  - линейные языки над алфавитом  $\Sigma$ , то  $L_1 \cup L_2$  тоже линейный язык.

*Доказательство.* Пусть язык  $L_1$  порождается линейной грамматикой  $\langle N_1, \Sigma, P_1, S_1 \rangle$  и

$L_2$  порождается линейной грамматикой  $\langle N_2, \Sigma, P_2, S_2 \rangle$ , где  $N_1 \cap N_2 = \emptyset$ . Тогда

$L_1 \cup L_2$  порождается грамматикой

$$\langle N_1 \cup N_2 \cup \{T\}, \Sigma, P_1 \cup P_2 \cup \{T \rightarrow S_1, T \rightarrow S_2\}, T \rangle,$$

где  $T \notin N_1 \cup N_2 \cup \Sigma$ .

**Пример 9.3.4.** Рассмотрим алфавит  $\Sigma = \{a, b, c\}$ . Язык

$$L = \Sigma^* - \{a^n b^n c^n \mid n \geq 0\}$$

является линейным, поскольку

$$L = L_1 \cup L_2 \cup (\Sigma^* - L_3),$$

где языки

$$L_1 = \{a^m b^n c^k \mid m \neq n, k \geq 0\}, \quad L_2 = \{a^m b^n c^k \mid n \neq k, m \geq 0\}$$

являются линейными, а язык

$$L_3 = \{a^m b^n c^k \mid m \geq 0, n \geq 0, k \geq 0\}$$

является автоматным, и можно применить теоремы 9.3.3, 3.2.1, 2.4.1 и лемму 1.5.13.

**Упражнение 9.3.5.** Пусть  $\Sigma = \{a, b, c\}$ . Является ли линейным язык  $\Sigma^* - \{ucu \mid u \in \{a, b\}^*\}$ ?

**Упражнение 9.3.6.** Пусть  $\Sigma = \{a, b, c\}$ . Является ли линейным язык  $\Sigma^* - \{ucu^R \mid u \in \{a, b\}^*\}$ ?

**Упражнение 9.3.7.** Найти линейную грамматику, порождающую язык  $\{a, b, c\}^* - L_1$ , где  $L_1$  порождается грамматикой

$$F \rightarrow baFa aa,$$

$$F \rightarrow baaFb,$$

$$F \rightarrow baaaa,$$

$$F \rightarrow baacb.$$

## 9.4. Свойства замкнутости класса контекстно-свободных языков

**Теорема 9.4.1.** Если  $L$  - контекстно-свободный язык, то  $L^*$  тоже контекстно-свободный язык.

*Доказательство.* Пусть язык  $L$  порождается контекстно-свободной грамматикой

$$\langle N, \Sigma, P, S \rangle.$$
 Тогда язык  $L^*$  порождается грамматикой  $\langle N \cup \{T\}, \Sigma, P \cup \{T \rightarrow ST, T \rightarrow \epsilon\}, T \rangle$ ,

где  $T \notin N \cup \Sigma$ .

**Теорема 9.4.2.** Если  $L_1$  и  $L_2$  - контекстно-свободные языки над алфавитом  $\Sigma$ , то  $L_1 \cdot L_2$  тоже контекстно-свободный язык.

*Доказательство.* Пусть язык  $L_1$  порождается контекстно-свободной грамматикой

$$\langle N_1, \Sigma, P_1, S_1 \rangle$$
 и  $L_2$  порождается контекстно-свободной грамматикой

$$\langle N_2, \Sigma, P_2, S_2 \rangle,$$
 где  $N_1 \cap N_2 = \emptyset$ . Тогда  $L_1 \cdot L_2$  порождается грамматикой  $\langle N_1 \cup N_2 \cup \{T\}, \Sigma, P_1 \cup P_2 \cup \{T \rightarrow S_1 S_2\}, T \rangle$ ,

где  $T \notin N_1 \cup N_2 \cup \Sigma$ .

**Теорема 9.4.3.** Если  $L_1$  и  $L_2$  - контекстно-свободные языки над алфавитом  $\Sigma$ , то  $L_1 \cup L_2$  тоже контекстно-свободный язык.

*Доказательство.* Пусть язык  $L_1$  порождается контекстно-свободной грамматикой

$$\langle N_1, \Sigma, P_1, S_1 \rangle$$
 и  $L_2$  порождается контекстно-свободной грамматикой

$$\langle N_2, \Sigma, P_2, S_2 \rangle,$$
 где  $N_1 \cap N_2 = \emptyset$ . Тогда  $L_1 \cup L_2$  порождается грамматикой  $\langle N_1 \cup N_2 \cup \{T\}, \Sigma, P_1 \cup P_2 \cup \{T \rightarrow S_1, T \rightarrow S_2\}, T \rangle$ ,

где  $T \notin N_1 \cup N_2 \cup \Sigma$ .

**Теорема 9.4.4.** Если  $L$  - контекстно-свободный язык, то  $L^R$  тоже контекстно-свободный язык.

**Упражнение 9.4.5.** Является ли контекстно-свободным язык

$$\{w \in \{a, b, c\}^* \mid |w|_a \neq |w|_b \text{ или } |w|_b \neq |w|_c\} ?$$

**Упражнение 9.4.6.** Найти контекстно-свободную грамматику для языка  $L_1 \cup L_2$ , где  $L_1$  порождается грамматикой

$$F \rightarrow a,$$

$$F \rightarrow bF,$$

$$F \rightarrow cFF,$$

а язык  $L_2$  порождается грамматикой

$$F \rightarrow aM, \quad M \rightarrow aF,$$

$$F \rightarrow cM, \quad M \rightarrow bF,$$

$$M \rightarrow \varepsilon.$$

## 9.5. Пересечение и дополнение контекстно-свободных языков

**Теорема 9.5.1.** Неверно, что для любых контекстно-свободных языков  $L_1$  и  $L_2$  язык

$L_1 \cap L_2$  тоже контекстно-свободный.

*Доказательство.* Положим  $L_1 = \{a^m b^m c^n \mid m \geq 0, n \geq 0\}$  и

$L_2 = \{a^m b^n c^n \mid m \geq 0, n \geq 0\}$ . В примере 9.2.1 было доказано, что язык  $L_1 \cap L_2$  не является контекстно-свободным.

**Теорема 9.5.2.** Неверно, что для любого контекстно-свободного языка  $L \subseteq \Sigma^*$  язык  $\Sigma^* - L$  тоже контекстно-свободный.

*Доказательство.* Положим  $L = \Sigma^* - \{a^n b^n c^n \mid n \geq 0\}$ , где  $\Sigma = \{a, b, c\}$ . В примере 9.3.4 было доказано, что язык  $L$  является линейным (и следовательно, контекстно-свободным).

**Упражнение 9.5.3.** Является ли контекстно-свободным язык  $\{a\}^* - \{a^{n^2} \mid n \geq 0\}$  ?

**Упражнение 9.5.4.** Является ли контекстно-свободным язык

$$\{a, b\}^* - \{ww \mid w \in \{a, b\}^*\} ?$$

**Упражнение 9.5.5.** Существует ли такой линейный язык  $L$  над алфавитом  $\{a, b\}$ , что язык  $L^R \cap L$  не является контекстно-свободным?

## 9.6. Пересечение контекстно-свободного языка с автоматным языком

**Теорема 9.6.1.** Если  $L_1$  - контекстно-свободный язык и  $L_2$  - автоматный язык, то язык

$L_1 \cap L_2$  является контекстно-свободным.

*Доказательство.* Пусть  $\langle N, \Sigma, P, S \rangle$  - контекстно-свободная грамматика, порождающая язык  $L_1$ . Без ограничения общности можно считать, что множество  $P$  содержит только правила вида  $A \rightarrow a$  и  $A \rightarrow \alpha$ , где  $A \in N$ ,  $a \in \Sigma$  и  $\alpha \in N^*$  (см. теорему 8.3.3).

Пусть  $\langle Q, \Sigma, \Delta, I, F \rangle$  - конечный автомат, распознающий язык  $L_2$ . Без ограничения общности можно считать, что для каждого перехода  $\langle p, x, q \rangle \in \Delta$  выполняется равенство  $|x| = 1$  (см. лемму 2.3.3).

Построим контекстно-свободную грамматику  $\langle \bar{N}, \Sigma, \bar{P}, \bar{S} \rangle$ , порождающую язык

$L_1 \cap L_2$ . Положим

$$\bar{N} = \{\bar{S}\} \cup (Q \times N \times Q),$$

$$\bar{P} = \{\bar{S} \rightarrow \langle p, S, q \rangle \mid p \in I, q \in F\} \cup$$

$$\cup \{\langle p, A, q \rangle \rightarrow a \mid \langle p, a, q \rangle \in \Delta, (A \rightarrow a) \in P\} \cup$$

$$\cup \{\langle p_0, A, p_n \rangle \rightarrow \langle p_0, B_1, p_1 \rangle \dots \langle p_{n-1}, B_n, p_n \rangle \mid \\ (A \rightarrow B_1 \dots B_n) \in P, p_0 \in Q, \dots, p_n \in Q\},$$

где  $\bar{S}$  - новый символ (не принадлежащий множеству  $Q \times N \times Q$ ).

**Пример 9.6.2.** Пусть  $\Sigma = \{a, b, c\}$ . Рассмотрим контекстно-свободный язык  $L_1$ , порождаемый грамматикой

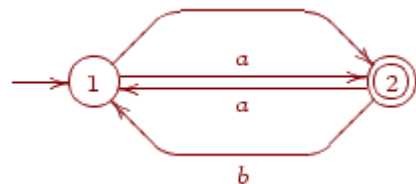
$$S \rightarrow a,$$

$$S \rightarrow bS,$$

$$S \rightarrow cSS,$$

и автоматный язык  $L_2$ , распознаваемый конечным автоматом  $M = \langle Q, \Sigma, \Delta, I, F \rangle$ , где  $Q = \{1, 2\}$ ,  $I = \{1\}$ ,  $F = \{2\}$ ,

$$\Delta = \{\langle 1, a, 2 \rangle, \langle 1, c, 2 \rangle, \langle 2, a, 1 \rangle, \langle 2, b, 1 \rangle\}.$$



Тогда язык  $L_1 \cap L_2$  порождается контекстно-свободной грамматикой

$$\bar{S} \rightarrow S_{12},$$

$$S_{11} \rightarrow cS_{21}S_{11},$$

$$S_{12} \rightarrow a,$$

$$S_{11} \rightarrow cS_{22}S_{21},$$

$$S_{21} \rightarrow a,$$

$$S_{12} \rightarrow cS_{21}S_{12},$$

$$S_{21} \rightarrow bS_{11},$$

$$S_{12} \rightarrow cS_{22}S_{22}.$$

$$S_{22} \rightarrow bS_{12},$$

Здесь  $S_{11}$ ,  $S_{12}$ ,  $S_{21}$  и  $S_{22}$  соответствуют символам  $\langle 1, S, 1 \rangle$ ,  $\langle 1, S, 2 \rangle$ ,  $\langle 2, S, 1 \rangle$  и  $\langle 2, S, 2 \rangle$  из доказательства теоремы 9.6.1.

**Теорема 9.6.3\*.** Если  $L_1$  - линейный язык и  $L_2$  - автоматный язык, то язык  $L_1 \cap L_2$  является линейным.

**Пример 9.6.4\*.** Пусть  $\Sigma = \{a, b\}$ . Рассмотрим линейный язык  $L_1$ , порождаемый грамматикой

$$S \rightarrow aaSaa,$$

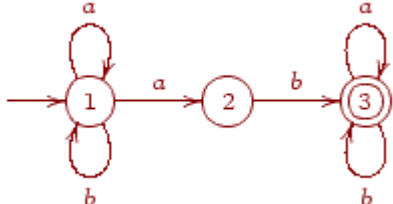
$$S \rightarrow bSb,$$

$$S \rightarrow \varepsilon,$$

и автоматный язык  $L_2$ , распознаваемый конечным автоматом  $M = \langle Q, \Sigma, \Delta, I, F \rangle$ , где

$$Q = \{1, 2, 3\}, I = \{1\}, F = \{3\},$$

$$\Delta = \{\langle 1, a, 1 \rangle, \langle 1, b, 1 \rangle, \langle 1, a, 2 \rangle, \langle 2, b, 3 \rangle, \langle 3, a, 3 \rangle, \langle 3, b, 3 \rangle\}.$$



Тогда язык  $L_1 \cap L_2$  порождается контекстно-свободной грамматикой

$$\bar{S} \rightarrow S_{13},$$

$$S_{11} \rightarrow bS_{11}b,$$

$$S_{11} \rightarrow aaS_{11}aa,$$

$$S_{13} \rightarrow bS_{13}b,$$

$$S_{12} \rightarrow aaS_{11}aa,$$

$$S_{13} \rightarrow bS_{12}b,$$

$$S_{13} \rightarrow aaS_{13}aa,$$

$$S_{23} \rightarrow bS_{33}b,$$

$$S_{13} \rightarrow aaS_{23}aa,$$

$$S_{33} \rightarrow bS_{33}b,$$

$$S_{33} \rightarrow aaS_{33}aa,$$

$$S_{11} \rightarrow \varepsilon,$$

$$S_{33} \rightarrow \varepsilon.$$

Эту грамматику можно упростить, заменив  $S_{11}$  и  $S_{33}$  на один символ.

**Упражнение 9.6.5.** Найти контекстно-свободную грамматику для языка  $L_1 \cap L_2$ , где  $L_1$  порождается грамматикой

$$F \rightarrow aFa,$$

$$F \rightarrow bFb,$$

$$F \rightarrow \varepsilon,$$

а язык  $L_2$  порождается грамматикой

$$K \rightarrow aM,$$

$$M \rightarrow aK,$$

$$K \rightarrow bK,$$

$$M \rightarrow bK,$$

$$M \rightarrow \varepsilon.$$

**Упражнение 9.6.6.** Найти контекстно-свободную грамматику для языка  $L_1 \cap L_2$ , где  $L_1$  порождается грамматикой

$$F \rightarrow aFa,$$

$$F \rightarrow bFb,$$

$$F \rightarrow \varepsilon,$$

а язык  $L_2$  порождается грамматикой

$$K \rightarrow aM,$$

$$M \rightarrow aK,$$

$$K \rightarrow bM,$$

$$K \rightarrow \varepsilon.$$

$$K \rightarrow bK,$$

**Упражнение 9.6.7.** Является ли контекстно-свободным язык

$$\{a^n b^n a^k b^n \mid n \geq 0, k \geq 0\}$$

**Упражнение 9.6.8.** Является ли контекстно-свободным язык

$$\{(ab)^{n^2} \mid n \geq 0\} \cup \{aaw \mid w \in \{a, b\}^*\}$$

**Упражнение 9.6.9.** Существует ли над алфавитом  $\{a, b\}$  такой линейный язык  $L$ , что язык  $\{ubv \mid u \in L, uv \in L\}$  не является контекстно-свободным?

## 9.7\*. Теорема Парика

**Замечание 9.1.7.** В этом разделе предполагается, что зафиксирован некоторый линейный порядок на алфавите  $\Sigma$ . Пусть  $\Sigma = \{a_1, \dots, a_n\}$ .

**Определение 9.7.2.** Через  $\Psi_\Sigma$  будем обозначать функцию из  $\Sigma^*$  в  $\mathbb{N}^n$ , определенную следующим образом:  $\Psi_\Sigma(w) = \langle |w|_{a_1}, \dots, |w|_{a_n} \rangle$ . Аналогично, каждому языку  $L \subseteq \Sigma^*$  ставится в соответствие множество  $\Psi_\Sigma(L) \subseteq \mathbb{N}^n$ , определенное следующим образом:  
 $\Psi_\Sigma(L) = \{\Psi_\Sigma(w) \mid w \in L\}$ .

**Пример 9.7.3.** Пусть  $\Sigma = \{a_1, a_2\}$  и  $L = \{a_1, a_1 a_2 a_2, a_2 a_2 a_1\}$ . Тогда  
 $\Psi_\Sigma(L) = \{\langle 1, 0 \rangle, \langle 1, 2 \rangle\}$ .

**Определение 9.7.4.** Пусть  $B \subseteq \mathbb{N}^n$  и  $P \subseteq \mathbb{N}^n$ . Тогда через  $L(B, P)$  обозначается множество

$$\{b + p_1 + \dots + p_k \mid b \in B, k \geq 0, p_1, \dots, p_k \in P\}.$$

При этом множество  $B$  называется *системой предпериодов* множества  $L(B, P)$ . Множество  $P$  называется *системой периодов* множества  $L(B, P)$ .

**Определение 9.7.5.** Множество  $A \subseteq \mathbb{N}^n$  называется *линейным* (linear), если  $A = L(B, P)$  для некоторых конечных множеств  $B$  и  $P$ .

**Определение 9.7.6.** Множество  $A \subseteq \mathbb{N}^n$  называется *полулинейным* (semilinear), если оно является объединением конечного числа линейных множеств.

**Теорема 9.7.7 (Теорема Парика).** Если язык  $L \subseteq \Sigma^*$  является контекстно-свободным, то множество  $\Psi_\Sigma(L)$  является полулинейным.

Доказательство можно найти в [Гин, с. 207-211].

**Пример 9.7.8.** Пусть  $\Sigma = \{a, b\}$ . Рассмотрим язык

$L = \{a^m b^n \mid m > n \text{ или } m \text{ простое}\}$ . Можно проверить, что множество  $\Psi_\Sigma(L)$  не является полулинейным. Следовательно, язык  $L$  не является контекстно-свободным.

**Теорема 9.7.9.** Если множество  $A \subseteq \mathbb{N}^n$  является полулинейным, то существует такой автоматный язык  $L$ , что  $A = \Psi_\Sigma(L)$ .

*Доказательство.* Докажем это для произвольного линейного множества  $A = L(B, P)$  (на



полулинейные множества утверждение распространяется по теореме 3.1.1). Рассмотрим конечный автомат  $M = \langle Q, \Sigma, \Delta, I, F \rangle$ , где  $Q = \{1, 2\}$ ,  $I = \{1\}$ ,  $F = \{2\}$  и  $\Delta = \{ \langle 1, a_1^{k_1} \dots a_n^{k_n}, 2 \rangle \mid \langle k_1, \dots, k_n \rangle \in B \} \cup \{ \langle 2, a_1^{k_1} \dots a_n^{k_n}, 2 \rangle \mid \langle k_1, \dots, k_n \rangle \in P \}$ .

Очевидно, что  $\Psi_\Sigma(L(M)) = L(B, P)$ .

**Замечание 9.7.10.** Теорема 9.3.1 является следствием теорем 9.7.7 и 9.7.9.

**Упражнение 9.7.11.** Является ли контекстно-свободным язык

$$\{a^{m^2}b^n \mid m \geq 0, n \geq 0\} \cup \{a^mb^n \mid 0 \leq n < m\}$$

© 2003-2008 INTUIT.ru. Все права защищены.

## Математическая теория формальных языков

### 10. Лекция: Автоматы с магазинной памятью: версия для печати и PDA

В данной лекции рассматриваются автоматы с магазинной памятью, в которых помимо ограниченной памяти, хранящей текущее состояние, имеется потенциально бесконечная память, используемая как магазин. Даются необходимые определения, и доказывается, что автоматы с магазинной памятью распознают в точности контекстно-свободные языки. Приведены примеры практического использования автоматов с магазинной памятью и приведены упражнения для самостоятельной проработки

Подобно тому, как праволинейным грамматикам соответствуют конечные автоматы, контекстно-свободным грамматикам соответствуют автоматы с магазинной памятью (МП-автоматы). В таком автомате, помимо ограниченной памяти, хранящей текущее состояние, имеется потенциально бесконечная память, используемая как стек (магазин), то есть структура данных, где в каждый момент доступен только тот элемент, который был добавлен позже остальных присутствующих на данный момент элементов.

Для наглядности стек обычно изображают вертикально, так, что доступный элемент данных (вершина) находится наверху. Но при формальном определении конфигурации (мгновенного состояния) МП-автомата удобно считать все содержимое стека конечной последовательностью символов, то есть словом (в алфавите, в котором перечислены все возможные данные, "умещающиеся" в одной ячейке стека). Прежде чем определить конфигурацию, придется принять произвольное соглашение о том, в каком порядке записывать содержимое стека в этом слове. В этой книге считается, что вершина стека находится в начале слова (то есть слева). В разделе 10.1 даются необходимые определения. В разделе 10.2 доказывается, что МП-автоматы распознают в точности контекстно-свободные языки.

#### 10.1. Определение автомата с магазинной памятью

**Определение 10.1.1.** Автомат с магазинной памятью (МП-автомат, магазинный автомат, стековый автомат, pushdown automaton) - это шестерка

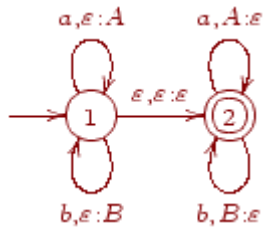
$M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ , где  $Q$ ,  $\Sigma$ ,  $\Gamma$  и  $\Delta$  - конечные множества,  $I \subseteq Q$ ,  $F \subseteq Q$  и  $\Delta \subseteq (Q \times \Sigma^* \times \Gamma^*) \times (Q \times \Gamma^*)$ .

Здесь  $Q$  - множество состояний,  $\Sigma$  - входной алфавит,  $\Gamma$  - алфавит магазинной памяти (stack alphabet),  $\Delta$  - множество переходов (transition relation), элементы  $I$  называются начальными состояниями, элементы  $F$  - заключительными или допускающими состояниями.

**Пример 10.1.2.** Пусть  $Q = \{1, 2\}$ ,  $\Sigma = \{a, b\}$ ,  $\Gamma = \{A, B\}$ ,  
 $\Delta = \{ \langle \langle 1, a, \varepsilon \rangle, \langle 1, A \rangle \rangle, \langle \langle 1, b, \varepsilon \rangle, \langle 1, B \rangle \rangle, \langle \langle 1, \varepsilon, \varepsilon \rangle, \langle 2, \varepsilon \rangle \rangle, \langle \langle 2, a, A \rangle, \langle 2, \varepsilon \rangle \rangle, \langle \langle 2, b, B \rangle, \langle 2, \varepsilon \rangle \rangle \}$ ,  
 $I = 1$ ,  $F = 2$ . Тогда  $\langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$  - МП-автомат.

**Замечание 10.1.3.** МП-автоматы можно изображать в виде диаграмм состояний. На диаграмме каждое состояние обозначается кружком, а переход - стрелкой. Каждое начальное состояние распознается по ведущей в него короткой стрелке. Каждое допускающее состояние отмечается на диаграмме двойным кружком. Стрелка с пометкой  $x, \beta : \gamma$ , ведущая из  $p$  в  $q$ , показывает, что  $\langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle$  является переходом данного МП-автомата.

**Пример 10.1.4.** Ниже приведена диаграмма МП-автомата из примера 10.1.2.



**Определение 10.1.5.** Конфигурацией МП-автомата называется любая тройка  $\langle q, w, \alpha \rangle$ , где  $q \in Q$ ,  $w \in \Sigma^*$ ,  $\alpha \in \Gamma^*$ .

**Определение 10.1.6.** Определим на множестве всех конфигураций МП-автомата  $M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$  бинарное отношение  $\vdash_M$  (такт работы) следующим образом. Если  $\langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta$ ,  $w \in \Sigma^*$  и  $\alpha \in \Gamma^*$ , то  $\langle p, xw, \beta\alpha \rangle \vdash_M \langle q, w, \gamma\alpha \rangle$ .

**Замечание 10.1.7** Обычно из контекста ясно, о каком МП-автомате идет речь. Тогда вместо  $\vdash_M$  будем писать  $\vdash$ .

**Пример 10.1.8.** Для МП-автомата из примера 10.1.2 выполняется  $\langle 1, abba, \varepsilon \rangle \vdash \langle 1, bba, A \rangle$  и  $\langle 1, abba, \varepsilon \rangle \vdash \langle 2, abba, \varepsilon \rangle$ .

**Определение 10.1.9.** Бинарное отношение  $\vdash^*$  определяется как рефлексивное, транзитивное замыкание отношения  $\vdash$ .

**Пример 10.1.10.** Для МП-автомата из примера 10.1.2 выполняется  $\langle 1, abba, \varepsilon \rangle \vdash^* \langle 1, ba, BA \rangle$  и  $\langle 1, abba, \varepsilon \rangle \vdash^* \langle 2, \varepsilon, \varepsilon \rangle$ .

**Лемма 10.1.11.** Если  $\langle p, x, \alpha_1 \rangle \vdash^* \langle q, \varepsilon, \alpha_2 \rangle$  и  $\langle q, y, \alpha_2\alpha_3 \rangle \vdash^* \langle r, \varepsilon, \alpha_4 \rangle$ , то  $\langle p, xy, \alpha_1\alpha_3 \rangle \vdash^* \langle r, \varepsilon, \alpha_4 \rangle$ .

*Доказательство.* Лемму легко доказать индукцией по количеству тактов в вычислительном процессе, ведущем из конфигурации  $\langle p, x, \alpha_1 \rangle$  в конфигурацию  $\langle q, \varepsilon, \alpha_2 \rangle$ .

**Определение 10.1.12.** Слово  $w \in \Sigma^*$  допускается МП-автоматом  $M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ , если найдутся такие состояния  $s \in I$  и  $q \in F$ , что  $\langle s, w, \varepsilon \rangle \vdash^* \langle q, \varepsilon, \varepsilon \rangle$ .

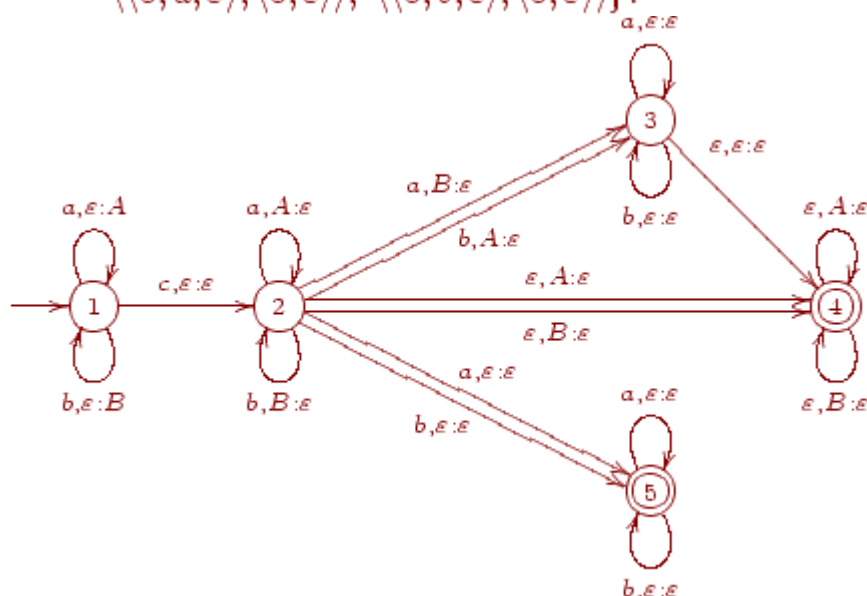
**Определение 10.1.13.** Язык, распознаваемый МП-автоматом, - множество всех слов, допускаемых этим МП-автоматом. Язык, распознаваемый МП-автоматом  $M$ , обозначается  $L(M)$ .

**Замечание 10.1.14.** Обычно в учебниках используется более узкое определение МП-автомата, но это не меняет класса языков, распознаваемых МП-автоматами.

**Пример 10.1.15.** Пусть  $M$  - МП-автомат из примера 10.1.2. Тогда  $L(M) = \{u^R \mid u \in \{a, b\}^*\}$ .

**Пример 10.1.16.** Пусть  $\Sigma = \{a, b, c\}$ . Рассмотрим МП-автомат

$M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ , где  $Q = \{1, 2, 3, 4, 5\}$ ,  $\Gamma = \{A, B\}$ ,  $I = \{1\}$ ,  $F = \{4, 5\}$  и  $\Delta = \{ \langle \langle 1, a, \varepsilon \rangle, \langle 1, A \rangle \rangle, \langle \langle 1, b, \varepsilon \rangle, \langle 1, B \rangle \rangle, \langle \langle 1, c, \varepsilon \rangle, \langle 2, \varepsilon \rangle \rangle, \langle \langle 2, a, A \rangle, \langle 2, \varepsilon \rangle \rangle, \langle \langle 2, b, B \rangle, \langle 2, \varepsilon \rangle \rangle, \langle \langle 2, a, B \rangle, \langle 3, \varepsilon \rangle \rangle, \langle \langle 2, b, A \rangle, \langle 3, \varepsilon \rangle \rangle, \langle \langle 2, \varepsilon, A \rangle, \langle 4, \varepsilon \rangle \rangle, \langle \langle 2, \varepsilon, B \rangle, \langle 4, \varepsilon \rangle \rangle, \langle \langle 2, b, \varepsilon \rangle, \langle 5, \varepsilon \rangle \rangle, \langle \langle 2, a, \varepsilon \rangle, \langle 5, \varepsilon \rangle \rangle, \langle \langle 3, a, \varepsilon \rangle, \langle 3, \varepsilon \rangle \rangle, \langle \langle 3, b, \varepsilon \rangle, \langle 3, \varepsilon \rangle \rangle, \langle \langle 3, \varepsilon, \varepsilon \rangle, \langle 4, \varepsilon \rangle \rangle, \langle \langle 4, \varepsilon, A \rangle, \langle 4, \varepsilon \rangle \rangle, \langle \langle 4, \varepsilon, B \rangle, \langle 4, \varepsilon \rangle \rangle, \langle \langle 5, a, \varepsilon \rangle, \langle 5, \varepsilon \rangle \rangle, \langle \langle 5, b, \varepsilon \rangle, \langle 5, \varepsilon \rangle \rangle \}$ .



Тогда  $L(M) = \{ucv \mid u \in \{a, b\}^*, v \in \{a, b\}^*, u \neq v^R\}$ .

**Определение 10.1.17.** Два МП-автомата эквивалентны, если они распознают один и тот же язык.

**Замечание 10.1.18.** В данном пособии не рассматриваются преобразователи с магазинной памятью (pushdown transducer) обобщение автоматов с магазинной памятью посредством добавления "выходного" потока (см. [7, 3.5] или [2, 3.1.4]).

**Замечание 10.1.19.** Некоторые авторы изменяют определение допускаемых слов следующим образом: слово  $w$  допускается МП-автоматом  $\langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ , если найдутся такие состояния  $s \in I$  и  $q \in F$  и последовательность  $\alpha \in \Gamma^*$ , что  $\langle s, w, \varepsilon \rangle \xrightarrow{*} \langle q, \varepsilon, \alpha \rangle$ . Такое определение не изменяет класса языков, распознаваемых МП-автоматами.

**Замечание 10.1.20.** Некоторые авторы добавляют в определение МП-автомата седьмую компоненту -  $z_0$ , называемую маркером магазинной памяти (start pushdown symbol), и изменяют определение допускаемых слов следующим образом: слово  $w$

допускается МП-автоматом  $\langle Q, \Sigma, \Gamma, \Delta, I, F, Z_0 \rangle$ , если найдутся такие состояния  $s \in I$  и  $q \in F$ , что  $\langle s, w, Z_0 \rangle \xrightarrow{*} \langle q, \varepsilon, \varepsilon \rangle$ . Такое определение не изменяет класса языков, распознаваемых МП-автоматами.

**Замечание 10.1.21.** Класс языков, распознаваемых МП-автоматами, не изменится также, если использовать следующую естественную комбинацию двух предыдущих определений: слово  $w$  допускается МП-автоматом  $\langle Q, \Sigma, \Gamma, \Delta, I, F, Z_0 \rangle$ , если найдутся такие состояния  $s \in I$  и  $q \in F$  и последовательность  $\alpha \in \Gamma^*$ , что  $\langle s, w, Z_0 \rangle \xrightarrow{*} \langle q, \varepsilon, \alpha \rangle$ .

**Замечание 10.1.22.** Некоторые авторы добавляют в определение МП-автомата маркер магазинной памяти  $Z_0$ , отбрасывают множество  $F$  и изменяют определение допускаемых слов следующим образом: слово  $w$  допускается МП-автоматом  $\langle Q, \Sigma, \Gamma, \Delta, I, Z_0 \rangle$ , если найдутся такие состояния  $s \in I$  и  $q \in Q$ , что  $\langle s, w, Z_0 \rangle \xrightarrow{*} \langle q, \varepsilon, \varepsilon \rangle$ . Это также не изменяет класса языков, распознаваемых МП-автоматами.

**Упражнение 10.1.23.** Найти МП-автомат, распознающий язык  $\{a^n b^n \mid n \geq 0\}$

**Упражнение 10.1.24.** Найти МП-автомат, распознающий язык  $\{a^i b^j c^k \mid i = j \text{ или } j = k\}$

**Упражнение 10.1.25.** Найти МП-автомат, распознающий язык  $\{w \in \{a, b\}^* \mid |w|_a = |w|_b\}$

## 10.2. Характеризация контекстно-свободных языков

**Теорема 10.2.1.** Если язык  $L$  является контекстно-свободным, то существует МП-автомат, распознающий этот язык.

*Доказательство.* Пусть язык  $L$  порождается контекстно-свободной грамматикой

$\langle N, \Sigma, P, S \rangle$ , в которой каждое правило имеет вид  $A \rightarrow w\beta$ , где  $A \in N$ ,  $w \in \Sigma^*$  и  $\beta \in N^*$  (в силу теоремы 8.8.3 такая грамматика существует). Положим  $\Gamma = N$ ,

$Q = 1, 2$ ,  $I = 1$ ,  $F = 2$  и

$\Delta = \{\langle 1, \varepsilon, \varepsilon \rangle, \langle 2, S \rangle\} \cup \{\langle 2, w, A \rangle, \langle 2, \beta \rangle \mid (A \rightarrow w\beta) \in P\}$ .

Можно доказать, что  $\langle 2, u, S \rangle \xrightarrow{*} \langle 2, \varepsilon, \alpha \rangle$  тогда и только тогда, когда существует

левосторонний вывод  $S \xrightarrow{\text{lm}}^* u\alpha$  (здесь  $u \in \Sigma^*$  и  $\alpha \in N^*$ ).

**Пример 10.2.2.** Пусть  $\Sigma = \{a, b, c, d\}$ . Контекстно-свободная грамматика

$S \rightarrow SS,$

$S \rightarrow a,$

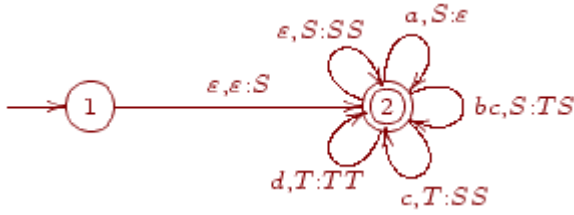
$S \rightarrow bcTS,$

$T \rightarrow cSS,$

$T \rightarrow dTT$

и МП-автомат  $\langle \{1, 2\}, \Sigma, \{S, T\}, \Delta, \{1\}, \{2\} \rangle$ , где

$\Delta = \{ \langle \langle 1, \varepsilon, \varepsilon \rangle, \langle 2, S \rangle \rangle, \langle \langle 2, \varepsilon, S \rangle, \langle 2, SS \rangle \rangle, \langle \langle 2, a, S \rangle, \langle 2, \varepsilon \rangle \rangle, \langle \langle 2, bc, S \rangle, \langle 2, TS \rangle \rangle, \langle \langle 2, c, T \rangle, \langle 2, SS \rangle \rangle, \langle \langle 2, d, T \rangle, \langle 2, TT \rangle \rangle \},$



задают один и тот же язык.

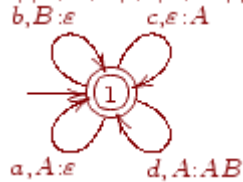
**Лемма 10.2.3.** Каждый МП-автомат эквивалентен некоторому МП-автомату

$\langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ , где  $|I| = 1$ ,  $|F| = 1$  и каждый переход  $\langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta$  удовлетворяет требованиям  $|x| \leq 1$  и  $|\beta| + |\gamma| = 1$ .

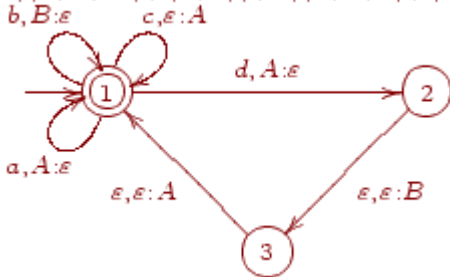
**Пример 10.2.4.** Рассмотрим МП-автомат  $M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ , где

$Q = I = F = \{1\}$ ,  $\Sigma = \{a, b, c, d\}$ ,  $\Gamma = \{A, B\}$ ,

$\Delta = \{ \langle \langle 1, a, A \rangle, \langle 1, \varepsilon \rangle \rangle, \langle \langle 1, b, B \rangle, \langle 1, \varepsilon \rangle \rangle, \langle \langle 1, c, \varepsilon \rangle, \langle 1, A \rangle \rangle, \langle \langle 1, d, A \rangle, \langle 1, AB \rangle \rangle \}.$



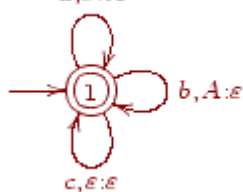
Он эквивалентен МП-автомату  $M' = \langle Q', \Sigma, \Gamma, \Delta', I, F \rangle$ , где  $Q' = \{1, 2, 3\}$  и  $\Delta' = \{ \langle \langle 1, a, A \rangle, \langle 1, \varepsilon \rangle \rangle, \langle \langle 1, b, B \rangle, \langle 1, \varepsilon \rangle \rangle, \langle \langle 1, c, \varepsilon \rangle, \langle 1, A \rangle \rangle, \langle \langle 1, d, A \rangle, \langle 2, \varepsilon \rangle \rangle, \langle \langle 2, \varepsilon, \varepsilon \rangle, \langle 3, B \rangle \rangle, \langle \langle 3, \varepsilon, \varepsilon \rangle, \langle 1, A \rangle \rangle \}.$



**Пример 10.2.5.** Рассмотрим МП-автомат  $M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ , где

$Q = I = F = \{1\}$ ,  $\Sigma = \{a, b, c\}$ ,  $\Gamma = \{A\}$ ,

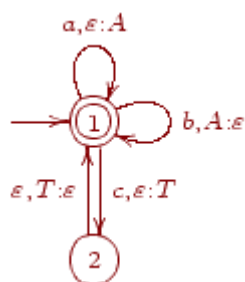
$\Delta = \{ \langle \langle 1, a, \varepsilon \rangle, \langle 1, A \rangle \rangle, \langle \langle 1, b, A \rangle, \langle 1, \varepsilon \rangle \rangle, \langle \langle 1, c, \varepsilon \rangle, \langle 1, \varepsilon \rangle \rangle \}.$



Он эквивалентен МП-автомату  $M' = \langle Q', \Sigma, \Gamma', \Delta', I, F \rangle$ , где  $Q' = \{1, 2\}$ ,

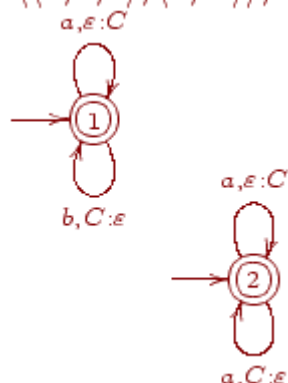
$\Gamma' = \{A, T\}$  и

$\Delta' = \{ \langle \langle 1, a, \varepsilon \rangle, \langle 1, A \rangle \rangle, \langle \langle 1, b, A \rangle, \langle 1, \varepsilon \rangle \rangle, \langle \langle 1, c, \varepsilon \rangle, \langle 2, T \rangle \rangle, \langle \langle 2, \varepsilon, T \rangle, \langle 1, \varepsilon \rangle \rangle \}.$



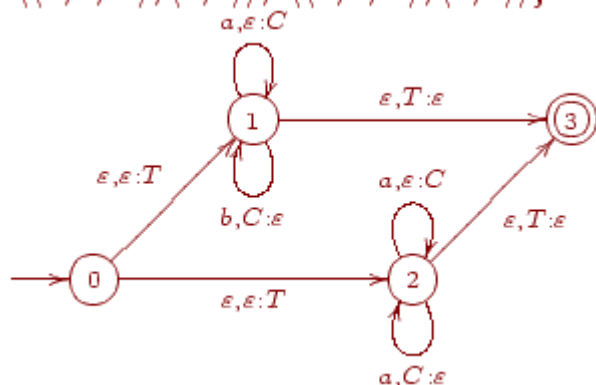
**Пример 10.2.6.** Рассмотрим МП-автомат  $M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ , где

$Q = I = F = \{1, 2\}$ ,  $\Sigma = \{a, b\}$ ,  $\Gamma = \{C\}$ ,  
 $\Delta = \{ \langle \langle 1, a, \varepsilon \rangle, \langle 1, C \rangle \rangle, \langle \langle 1, b, C \rangle, \langle 1, \varepsilon \rangle \rangle, \langle \langle 2, b, \varepsilon \rangle, \langle 2, C \rangle \rangle, \langle \langle 2, a, C \rangle, \langle 2, \varepsilon \rangle \rangle \}$ .



Он эквивалентен МП-автомату  $M' = \langle Q', \Sigma, \Gamma', \Delta', I', F' \rangle$ , где  $Q' = \{0, 1, 2, 3\}$ ,

$\Gamma' = \{A, T\}$ ,  $I' = \{0\}$ ,  $F' = \{3\}$ ,  
 $\Delta' = \Delta \cup \{ \langle \langle 0, \varepsilon, \varepsilon \rangle, \langle 1, T \rangle \rangle, \langle \langle 0, \varepsilon, \varepsilon \rangle, \langle 2, T \rangle \rangle, \langle \langle 1, \varepsilon, T \rangle, \langle 3, \varepsilon \rangle \rangle, \langle \langle 2, \varepsilon, T \rangle, \langle 3, \varepsilon \rangle \rangle \}$ .



**Теорема 10.2.7.** Если язык  $L$  распознается некоторым МП-автоматом, то  $L$  является контекстно-свободным.

*Доказательство.* Пусть язык  $L$  распознается МП-автоматом  $\langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ . Без ограничения общности можно считать, что  $I = \{q_s\}$ ,  $F = \{q_a\}$  и каждый переход  $\langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta$  удовлетворяет требованию  $|\beta| + |\gamma| = 1$ . Построим искомую контекстно-свободную грамматику  $\langle N, \Sigma, P, S \rangle$ , положив

$N = \{A_{p,q} \mid p \in Q, q \in Q\}$ ,  $S = A_{q_s, q_a}$  и

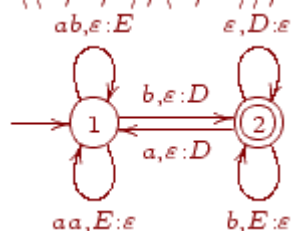
$$P = \{A_{p,p} \rightarrow \varepsilon \mid p \in Q\} \cup \\ \cup \{A_{p,t} \rightarrow xA_{q,r}yA_{s,t} \mid \langle \langle p, x, \varepsilon \rangle, \langle q, C \rangle \rangle \in \Delta, \\ \langle \langle r, y, C \rangle, \langle s, \varepsilon \rangle \rangle \in \Delta, C \in \Gamma, t \in Q\}.$$

Можно доказать, что  $\langle p, x, \varepsilon \rangle \vdash^* \langle q, \varepsilon, \varepsilon \rangle$  тогда и только тогда, когда  $A_{p,q} \xRightarrow{*} x$  (здесь  $x \in \Sigma^*$ ).

**Пример 10.2.8.** МП-автомат  $M = \langle \{1, 2\}, \Sigma, \Gamma, \Delta, \{1\}, \{2\} \rangle$ , где  $\Sigma = \{a, b\}$ ,

$$\Gamma = \{D, E\},$$

$$\Delta = \{ \langle \langle 1, ab, \varepsilon \rangle, \langle 1, E \rangle \rangle, \langle \langle 1, aa, E \rangle, \langle 1, \varepsilon \rangle \rangle, \langle \langle 1, b, \varepsilon \rangle, \langle 2, D \rangle \rangle, \\ \langle \langle 2, a, \varepsilon \rangle, \langle 1, D \rangle \rangle, \langle \langle 2, \varepsilon, D \rangle, \langle 2, \varepsilon \rangle \rangle, \langle \langle 2, b, E \rangle, \langle 2, \varepsilon \rangle \rangle \},$$



и контекстно-свободная грамматика

$$S \rightarrow abTaaS, \quad T \rightarrow abTaaT,$$

$$S \rightarrow abSbU, \quad T \rightarrow \varepsilon,$$

$$S \rightarrow bUU, \quad U \rightarrow aSU,$$

$$U \rightarrow \varepsilon$$

задают один и тот же язык. Здесь  $S$ ,  $T$  и  $U$  соответствуют символам  $A_{1,2}$ ,  $A_{1,1}$  и  $A_{2,2}$  из доказательства теоремы 10.2.7.

**Упражнение 10.2.9.** Найти МП-автомат, распознающий язык, порождаемый грамматикой

$$F \rightarrow aFb,$$

$$F \rightarrow aFa,$$

$$F \rightarrow a.$$

**Упражнение 10.2.10.** Найти контекстно-свободную грамматику, порождающую язык

$$\{w \in \{a, b\}^* \mid |w|_a = |w|_b\}$$

**Упражнение 10.2.10.** Найти контекстно-свободную грамматику, порождающую язык

$$\{w \in \{a, b\}^* \mid |w|_a \geq |w|_b\}$$

**Упражнение 10.2.10.** Найти контекстно-свободную грамматику, порождающую язык

$$\{w \in \{a, b\}^* \mid |w|_a = 2|w|_b\}$$

### 10.3\*. Автоматы с магазинной памятью с однобуквенными переходами

**Теорема 10.3.1.** Каждый МП-автомат эквивалентен некоторому МП-автомату

$$\langle Q, \Sigma, \Gamma, \Delta, I, F \rangle, \text{ где } |Q| = 2 \text{ и каждый переход } \langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta$$

удовлетворяет требованиям  $|x| = 1$ ,  $|\beta| \leq 1$  и  $|\gamma| \leq 2$ .



*Доказательство.* Пусть исходным МП-автоматом распознается контекстно-свободный язык  $L \subseteq \Sigma^*$ . Согласно теореме 8.4.6 язык  $L - \{\varepsilon\}$  порождается некоторой контекстно-свободной грамматикой  $\langle N, \Sigma, P, S \rangle$ , в которой каждое правило имеет вид  $A \rightarrow a\gamma$ , где  $A \in N$ ,  $a \in \Sigma$ ,  $\gamma \in N^*$  и  $|\gamma| \leq 2$ . Аналогично тому, как было сделано при доказательстве теоремы 10.2.1, положим  $\Gamma = N$ ,  $Q = \{1, 2\}$ ,  $I = \{1\}$ ,

$$F = \begin{cases} \{1, 2\}, & \text{если } \varepsilon \in L, \\ \{2\}, & \text{если } \varepsilon \notin L, \end{cases}$$

$$\Delta = \{ \langle \langle 2, a, A \rangle, \langle 2, \gamma \rangle \rangle \mid (A \rightarrow a\gamma) \in P \} \cup \{ \langle \langle 1, a, \varepsilon \rangle, \langle 2, \gamma \rangle \rangle \mid (S \rightarrow a\gamma) \in P \}.$$

**Теорема 10.3.2.** Каждый МП-автомат эквивалентен некоторому МП-автомату  $\langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ , в котором каждый переход  $\langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta$  удовлетворяет требованиям  $|x| = 1$ ,  $|\beta| \leq 1$  и  $|\gamma| \leq 1$ .

*Доказательство.* Пусть исходным МП-автоматом распознается контекстно-свободный язык  $L \subseteq \Sigma^*$ . Согласно теореме 8.4.6 язык  $L - \{\varepsilon\}$  порождается некоторой контекстно-свободной грамматикой  $G = \langle N, \Sigma, P, S \rangle$ , в которой каждое правило имеет один из следующих трех видов:  $A \rightarrow a$ ,  $A \rightarrow aB$ ,  $A \rightarrow aBC$ , где  $A \in N$ ,  $B \in N$ ,  $C \in N$ ,  $a \in \Sigma$ . Легко добиться того, чтобы в правилах грамматики  $G$  вспомогательные символы в правой части (то есть символы  $B$  и  $C$ ) были отличны от начального символа  $S$ .

Положим  $Q = N \cup \{\perp\}$ , где  $\perp \notin N$ . Далее, положим  $\Gamma = N$ ,

$$I = \{S\}, \quad F = \begin{cases} \{S, \perp\}, & \text{если } \varepsilon \in L, \\ \{\perp\}, & \text{если } \varepsilon \notin L, \end{cases}$$

$$\Delta = \{ \langle \langle A, a, \varepsilon \rangle, \langle B, C \rangle \rangle \mid (A \rightarrow aBC) \in P \} \cup$$

$$\cup \{ \langle \langle A, a, \varepsilon \rangle, \langle B, \varepsilon \rangle \rangle \mid (A \rightarrow aB) \in P \} \cup$$

$$\cup \{ \langle \langle A, a, D \rangle, \langle D, \varepsilon \rangle \rangle \mid (A \rightarrow a) \in P, D \in N - \{S\} \} \cup$$

$$\cup \{ \langle \langle A, a, \varepsilon \rangle, \langle \perp, \varepsilon \rangle \rangle \mid (A \rightarrow a) \in P \}.$$

**Упражнение 10.3.3.** Найти для языка, порождаемого грамматикой

$$S \rightarrow aTT,$$

$$S \rightarrow \varepsilon,$$

$$T \rightarrow bTU,$$

$$T \rightarrow c,$$

$$U \rightarrow aT,$$

МП-автомат, в котором каждый переход  $\langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta$  удовлетворяет требованиям  $|x| = 1$ ,  $|\beta| \leq 1$  и  $|\gamma| \leq 1$ .

**Упражнение 10.3.4.** Найти для языка, порождаемого грамматикой

$$S \rightarrow aTU,$$

$$T \rightarrow aUT,$$

$$T \rightarrow b,$$

$$U \rightarrow bT,$$

$$U \rightarrow a,$$

МП-автомат, в котором каждый переход  $\langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta$  удовлетворяет требованиям  $|x| = 1$ ,  $|\beta| \leq 1$  и  $|\gamma| \leq 1$ .

© 2003-2008 INTUIT.ru. Все права защищены.

## Математическая теория формальных языков

### 11. Лекция: Дополнительные свойства контекстно-свободных языков: версия для печати и PDA

В данной лекции рассматриваются свойства контекстно-свободных языков, которые удобно доказывать с привлечением автоматов с магазинной памятью, а также формулируются два критерия контекстной свободности, интересных в основном с теоретической точки зрения. Приведены практические примеры и упражнения для самостоятельного решения

В этой лекции излагаются те свойства контекстно-свободных языков, которые удобно доказывать с привлечением автоматов с магазинной памятью. В первых двух разделах приводятся некоторые свойства замкнутости класса контекстно-свободных языков (замкнутость относительно деления, взятия гомоморфного образа и полного гомоморфного прообраза). В конце лекции формулируются два критерия контекстной свободности, интересных в основном с теоретической точки зрения.

#### 11.1\*. Деление контекстно-свободных языков

**Теорема 11.1.1.** Пусть  $L_1$  - контекстно-свободный язык над алфавитом  $\Sigma$  и  $L_2$  - автоматный язык над алфавитом  $\Sigma$ . Тогда язык  $\{x \in \Sigma^* \mid (\exists y \in L_2) xy \in L_1\}$  является контекстно-свободным.

*Доказательство.* Пусть  $\langle Q_1, \Sigma, \Gamma_1, \Delta_1, I_1, F_1 \rangle$  - МП-автомат, распознающий язык  $L_1$ . Без ограничения общности можно считать, что для каждого перехода

$$\langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta_1 \text{ выполняется неравенство } |x| \leq 1.$$

Пусть  $\langle Q_2, \Sigma, \Delta_2, I_2, F_2 \rangle$  - конечный автомат, распознающий язык  $L_2$ . Без ограничения общности можно считать, что

$$Q_1 \cap (Q_1 \times Q_2) = \emptyset$$

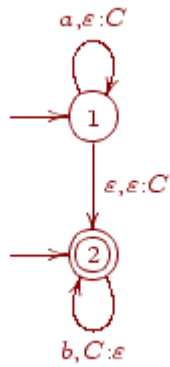
и для каждого перехода  $\langle p, x, q \rangle \in \Delta_2$  выполняется равенство  $|x| = 1$ .

Тогда язык  $L = \{x \in \Sigma^* \mid (\exists y \in L_2) xy \in L_1\}$  распознается МП-автоматом

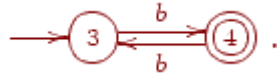
$\langle Q, \Sigma, \Gamma_1, \Delta, I, F \rangle$ , где  $Q = Q_1 \cup (Q_1 \times Q_2)$ ,  $I = I_1$ ,  $F = F_1 \times F_2$  и  $\Delta = \Delta_1 \cup$

$$\begin{aligned} & \cup \{ \langle \langle p_1, \varepsilon, \varepsilon \rangle, \langle \langle p_1, p_2 \rangle, \varepsilon \rangle \rangle \mid p_1 \in Q_1, p_2 \in I_2 \} \cup \\ & \cup \{ \langle \langle \langle p_1, p_2 \rangle, \varepsilon, \beta \rangle, \langle \langle q_1, q_2 \rangle, \gamma \rangle \rangle \mid \langle \langle p_1, a, \beta \rangle, \langle q_1, \gamma \rangle \rangle \in \Delta_1, \\ & \quad \langle p_2, a, q_2 \rangle \in \Delta_2 \text{ для некоторого } a \in \Sigma \} \cup \\ & \cup \{ \langle \langle \langle p_1, p_2 \rangle, \varepsilon, \beta \rangle, \langle \langle q_1, p_2 \rangle, \gamma \rangle \rangle \mid \langle \langle p_1, \varepsilon, \beta \rangle, \langle q_1, \gamma \rangle \rangle \in \Delta_1, p_2 \in Q_2 \}. \end{aligned}$$

**Пример 11.1.2.** Пусть  $\Sigma = \{a, b\}$ , язык  $L_1$  распознается МП-автоматом



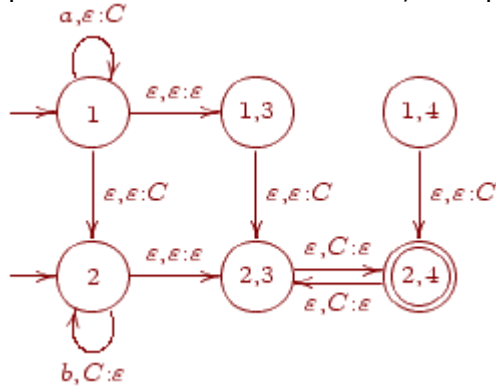
и язык  $L_2$  распознается конечным автоматом



Следуя доказательству теоремы 11.1.1, получаем, что язык

$$\{x \in \Sigma^* \mid (\exists y \in L_2) xy \in L_1\}$$

распознается МП-автоматом, изображенным ниже.



**Пример 11.1.3.** Пусть  $\Sigma = \{a, b\}$ ,  $L_1 = \{w \in \Sigma^* \mid |w|_a = |w|_b\}$  и

$L_2 = \{a^n \mid n \geq 2\}$ . Тогда

$$\{x \in \Sigma^* \mid (\exists y \in L_2) xy \in L_1\} = \{w \in \Sigma^* \mid |w|_a + 2 \leq |w|_b\}.$$

**Пример 11.1.4.** Пусть  $\Sigma = \{a, b\}$ ,  $L_1 = \{w \in \Sigma^* \mid w = w^R\}$  и

$L_2 = \{uaabbbv \mid u \in \Sigma^*, v \in \Sigma^*\}$ . Тогда

$$\{x \in \Sigma^* \mid (\exists y \in L_2) xy \in L_1\} = \Sigma^*.$$

**Замечание 11.1.5.** Пусть  $\Sigma = \{a, b, c\}$  и  $L \subseteq \Sigma^*$ . Язык  $L \cdot \{c\}$  является контекстно-свободным тогда и только тогда, когда язык  $L$  является контекстно-свободным.

**Упражнение 11.1.6.** Существует ли такой контекстно-свободный язык  $L \subseteq \Sigma^*$ , что язык  $\text{Sub}w$  не является контекстно-свободным?

**Упражнение 11.1.7.** Существует ли такой контекстно-свободный язык  $L$  над алфавитом  $\{a, b\}$ , что язык  $\{w \mid (\exists n \geq 0) wa^{2n} \in L\}$  не является контекстно-свободным?

**Упражнение 11.1.8.** Существует ли такой контекстно-свободный язык  $L$  над алфавитом  $\{a, b\}$ , что язык

$$\{w \in \{a, b\}^* \mid (\exists m \geq 0) (\exists n \geq 0) (ab)^m wb^{3n} \in L\}$$

не является контекстно-свободным?

## 11.2. Гомоморфизмы и контекстно-свободные языки

**Теорема 11.2.1.** Для любого гомоморфизма  $h: \Sigma_1^* \rightarrow \Sigma_2^*$  и контекстно-свободного языка  $L \subseteq \Sigma_1^*$  язык  $h(L)$  является контекстно-свободным.

*Доказательство.* Приведем здесь доказательство, использующее МП-автоматы, хотя эту теорему легко доказать и с помощью контекстно-свободных грамматик. Пусть язык  $L$  распознается МП-автоматом  $\langle Q, \Sigma_1, \Gamma, \Delta, I, F \rangle$ . Тогда язык  $h(L)$  распознается МП-автоматом  $\langle Q, \Sigma_2, \Gamma, \Delta', I, F \rangle$ , где  $\Delta' = \{ \langle \langle p, h(x), \beta \rangle, \langle q, \gamma \rangle \rangle \mid \langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta \}$ .

**Пример 11.2.2.** Пусть  $\Sigma_1 = \{a, b, c\}$  и  $\Sigma_2 = \{a, b\}$ . Рассмотрим контекстно-свободный язык  $L$ , порождаемый грамматикой

$$S \rightarrow aSSc,$$

$$S \rightarrow b,$$

и гомоморфизм  $h: \Sigma_1^* \rightarrow \Sigma_2^*$ , заданный равенствами  $h(a) = a$ ,  $h(b) = bba$  и  $h(c) = a$ . Тогда язык  $h(L)$  порождается контекстно-свободной грамматикой

$$S \rightarrow aSSa,$$

$$S \rightarrow bba,$$

**Упражнение 11.2.3.** Пусть гомоморфизм  $h: \{a, b, c\}^* \rightarrow \{a, b\}^*$  задан

соотношениями  $h(a) = b$ ,  $h(b) = \varepsilon$ ,  $h(c) = a$ . Рассмотрим язык  $L$ , порождаемый грамматикой

$$S \rightarrow a,$$

$$S \rightarrow bS,$$

$$S \rightarrow cSS.$$

Найти контекстно-свободную грамматику для языка  $h(L)$ .

**Теорема 11.2.4.** Для любого гомоморфизма  $h: \Sigma_1^* \rightarrow \Sigma_2^*$  и контекстно-свободного языка  $L \subseteq \Sigma_2^*$  язык  $h^{-1}(L)$  является контекстно-свободным.

*Доказательство.* Введем обозначение

$$\mathcal{A} = \{x \in \Sigma_2^* \mid x \sqsupseteq h(a) \text{ для некоторого } a \in \Sigma_1\}.$$

Пусть язык  $L$  распознается МП-автоматом  $\langle Q, \Sigma_2, \Gamma, \Delta, I, F \rangle$ . Без ограничения

общности можно считать, что для каждого перехода  $\langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta$

выполняется неравенство  $|x| \leq 1$ . Можно проверить, что в этом случае язык  $h^{-1}(L)$

распознается МП-автоматом  $\langle Q', \Sigma_1, \Gamma, \Delta', I', F' \rangle$ , где

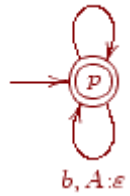
$$Q' = \mathcal{A} \times Q,$$

$$I' = \{\varepsilon\} \times I,$$

$$F' = \{\varepsilon\} \times F,$$

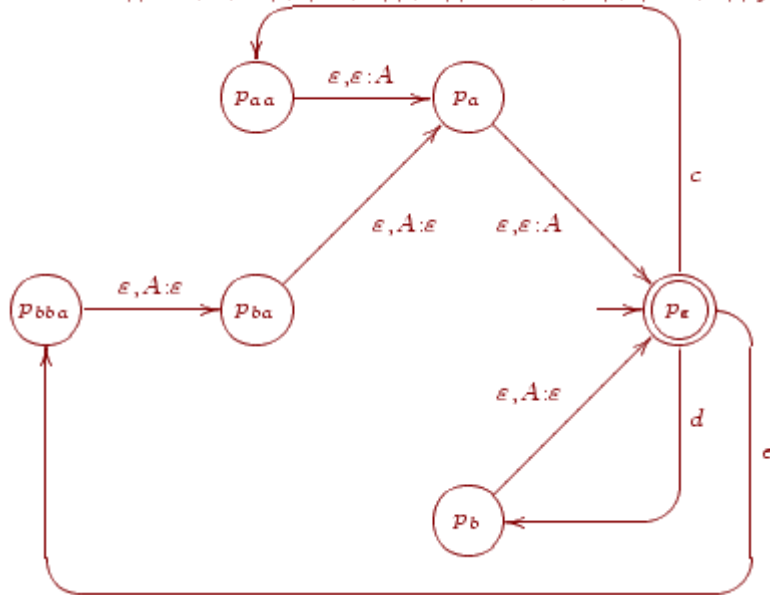
$$\Delta' = \{ \langle \langle \varepsilon, p \rangle, a, \varepsilon \rangle, \langle \langle h(a), p \rangle, \varepsilon \rangle \rangle \mid a \in \Sigma_1, p \in Q \} \cup \\ \cup \{ \langle \langle xw, p \rangle, \varepsilon, \beta \rangle, \langle \langle w, q \rangle, \gamma \rangle \rangle \mid \langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta, xw \in \mathcal{A} \}.$$

**Пример 11.2.5.** Пусть  $\Sigma_1 = \{c, d, e\}$  и  $\Sigma_2 = \{a, b\}$ . Рассмотрим гомоморфизм  $h: \Sigma_1^* \rightarrow \Sigma_2^*$ , заданный равенствами  $h(c) = aa$ ,  $h(d) = b$  и  $h(e) = bba$ . Пусть контекстно-свободный язык  $L$  распознается МП-автоматом  $\langle Q, \Sigma_2, \Gamma, \Delta, I, F \rangle$ , где  $Q = \{p\}$ ,  $\Gamma = \{A\}$ ,  $I = \{p\}$ ,  $F = \{p\}$ ,  $\Delta = \{ \langle \langle p, a, \varepsilon \rangle, \langle p, A \rangle \rangle, \langle \langle p, b, A \rangle, \langle p, \varepsilon \rangle \rangle \}$ .



Тогда язык  $h^{-1}(L)$  распознается МП-автоматом  $\langle Q', \Sigma_1, \Gamma, \Delta', I', F' \rangle$ , где

$Q' = \{p_\varepsilon, p_a, p_{aa}, p_b, p_{ba}, p_{bba}\}$ ,  $I' = \{p_\varepsilon\}$ ,  $F' = \{p_\varepsilon\}$  и  $\Delta' = \{ \langle \langle p_\varepsilon, c, \varepsilon \rangle, \langle p_{aa}, \varepsilon \rangle \rangle, \langle \langle p_\varepsilon, d, \varepsilon \rangle, \langle p_b, \varepsilon \rangle \rangle, \langle \langle p_\varepsilon, e, \varepsilon \rangle, \langle p_{bba}, \varepsilon \rangle \rangle, \langle \langle p_a, \varepsilon, \varepsilon \rangle, \langle p_\varepsilon, A \rangle \rangle, \langle \langle p_{aa}, \varepsilon, \varepsilon \rangle, \langle p_a, A \rangle \rangle, \langle \langle p_b, \varepsilon, A \rangle, \langle p_\varepsilon, \varepsilon \rangle \rangle, \langle \langle p_{ba}, \varepsilon, A \rangle, \langle p_a, \varepsilon \rangle \rangle, \langle \langle p_{bba}, \varepsilon, A \rangle, \langle p_{ba}, \varepsilon \rangle \rangle \}$ .



Язык  $h^{-1}(L)$  также порождается контекстно-свободной грамматикой

$$\begin{aligned} S &\rightarrow \varepsilon, & T &\rightarrow \varepsilon, \\ S &\rightarrow cT U d S, & T &\rightarrow cT U U, \\ & & U &\rightarrow dT, \\ & & U &\rightarrow eT. \end{aligned}$$

**Упражнение 11.2.6.** Пусть гомоморфизм  $h: \{a, b, c\}^* \rightarrow \{a, b\}^*$  задан соотношениями  $h(a) = ab$ ,  $h(b) = aaba$ ,  $h(c) = b$ . Рассмотрим язык  $L$ , порождаемый грамматикой

$$\begin{aligned} S &\rightarrow aS, \\ S &\rightarrow aSbS, \\ S &\rightarrow bSaS. \end{aligned}$$

Описать язык  $h^{-1}(L)$ .

**Упражнение 11.2.7.**  $h: \{c, d, e\}^* \rightarrow \{a, b\}^*$  задан соотношениями  $h(c) = a$ ,  $h(d) = ba$ ,  $h(e) = bb$ . Рассмотрим язык  $L$ , порождаемый грамматикой

$$B \rightarrow aBb, a,$$

$$B \rightarrow b.$$

Найти контекстно-свободную грамматику для языка  $h^{-1}(L)$ .

**Упражнение 11.2.8.** Пусть гомоморфизм  $h: \{c, d, e\}^* \rightarrow \{a, b\}^*$  задан соотношениями  $h(c) = bba$ ,  $h(d) = aa$ ,  $h(e) = b$ . Рассмотрим язык  $L$ , порождаемый грамматикой

$$S \rightarrow aSb, a,$$

$$S \rightarrow \varepsilon.$$

Найти контекстно-свободную грамматику для языка  $h^{-1}(L)$ .

**Упражнение 11.2.9.** Обозначим через  $L$  язык, порождаемый грамматикой

$$S \rightarrow aSb, a,$$

$$S \rightarrow ba.$$

Рассмотрим гомоморфизм  $h_1: \{a, b, c, d, e, f\}^* \rightarrow \{a, b, c\}^*$ , заданный соотношениями

$$h_1(a) = ac,$$

$$h_1(b) = bc,$$

$$h_1(c) = aa,$$

$$h_1(d) = ab,$$

$$h_1(e) = ba,$$

$$h_1(f) = bb,$$

и гомоморфизм  $h_2: \{a, b, c, d, e, f\}^* \rightarrow \{a, b\}^*$ , заданный соотношениями

$$h_2(a) = a,$$

$$h_2(b) = b,$$

$$h_2(c) = ab,$$

$$h_2(d) = aa,$$

$$h_2(e) = bb,$$

$$h_2(f) = ba.$$

Найти контекстно-свободную грамматику, порождающую язык

$$h_2(h_1^{-1}(L \cdot \{\varepsilon, c\})).$$

### 11.3. Представления контекстно-свободных языков посредством гомоморфизмов

**Теорема 11.3.1.** Рассмотрим алфавит  $\Sigma_0 = \{a_1, a_2, b_1, b_2, c\}$  и язык  $L_0 \subseteq \Sigma_0^*$ , порождаемый контекстно-свободной грамматикой  $G_0$ :

$$S \rightarrow Cb_1b_2b_1RC, \quad A \rightarrow a_1, \quad R \rightarrow \varepsilon,$$

$$C \rightarrow c, \quad A \rightarrow a_2, \quad R \rightarrow Ta_1Tb_1,$$

$$C \rightarrow cDC, \quad A \rightarrow b_1, \quad R \rightarrow Ta_2Tb_2,$$

$$D \rightarrow A, \quad A \rightarrow b_2, \quad T \rightarrow R,$$

$$D \rightarrow AD, \quad T \rightarrow RCC.$$

Произвольный язык  $L \subseteq \Sigma^*$  является контекстно-свободным тогда и только тогда, когда существует такой гомоморфизм  $h: \Sigma^* \rightarrow \Sigma_0^*$ , что  $L = h^{-1}(L_0)$  или

$$L = h^{-1}(L_0 \cup \{\varepsilon\}).$$

**Доказательство.** Достаточность следует из теоремы 11.2.4. Приведем теперь идею доказательства необходимости (полное доказательство можно найти в [Сал, с. 103-109]).

Пусть дан произвольный контекстно-свободный язык  $L$ . Согласно теореме 8.4.6 язык

$L - \{\varepsilon\}$  порождается некоторой контекстно-свободной грамматикой

$\langle \{A_1, \dots, A_n\}, \Sigma, P, A_1 \rangle$ , в которой каждое правило имеет один из следующих трех видов:  $A_i \rightarrow a$ ,  $A_i \rightarrow aA_j$ ,  $A_i \rightarrow aA_jA_k$ , где  $a \in \Sigma$ .

Определим вспомогательную функцию  $\bar{h}$ , ставящую в соответствие каждому символу из  $\Sigma$  конечный язык над алфавитом  $\Sigma_0$  следующим образом:

$$\begin{aligned} \bar{h}(a) = & \{b_1b_2^ib_1 \mid (A_i \rightarrow a) \in P\} \cup \\ & \cup \{b_1b_2^ib_1a_1a_2^ja_1 \mid (A_i \rightarrow aA_j) \in P\} \cup \\ & \cup \{b_1b_2^ib_1a_1a_2^ka_1a_1a_2^ja_1 \mid (A_i \rightarrow aA_jA_k) \in P\}. \end{aligned}$$

Искомый гомоморфизм  $h$  определяется следующим образом: если

$$\bar{h}(a) = \{w_1, w_2, \dots, w_k\},$$

положим

$$h(a) = cw_1cw_2c \dots cw_kc.$$

**Пример 11.3.2.** Пусть  $\Sigma = \{d, f, g\}$ . Рассмотрим язык  $L$ , порождаемый грамматикой

$$A_1 \rightarrow f,$$

$$A_1 \rightarrow dA_1A_2,$$

$$A_2 \rightarrow fA_3,$$

$$A_2 \rightarrow fA_2A_3,$$

$$A_3 \rightarrow g.$$

Тогда  $L = h^{-1}(L_0)$ , где гомоморфизм  $h$  задан равенствами

$$h(d) = cb_1b_2b_1a_1a_2a_2a_1a_2a_1c,$$

$$h(f) = cb_1b_2b_1cb_1b_2b_2b_1a_1a_2a_2a_1cb_1b_2b_2b_1a_1a_2a_2a_1a_2a_2a_1c,$$

$$h(g) = cb_1b_2b_2b_2b_1c.$$

Рассмотрим, например, слово  $df f g \in L$ . Проверим, что слово  $h(df f g)$  выводится в грамматике  $G_0$  из теоремы 11.3.1. Очевидно, что  $S \xRightarrow{*} cb_1b_2b_1Rc$ . С помощью последних пяти правил грамматики  $G_0$  можно вывести, что

$$R \xRightarrow{*} a_1a_2a_2a_1a_1a_2a_1CCb_1b_2b_1CCb_1b_2b_2b_1a_1a_2a_2a_2a_1CCb_1b_2b_2b_2b_1.$$

Осталось найти такие шесть выводимых из  $c$  слов  $w_1, \dots, w_6$ , что

$$\begin{aligned} h(df f g) = \\ = cb_1b_2b_1a_1a_2^2a_1a_1a_2a_1w_1w_2b_1b_2b_1w_3w_4b_1b_2^2b_1a_1a_2^3a_1w_5w_6b_1b_2^3b_1c. \end{aligned}$$

Подходят слова



$w_1 = c,$   
 $w_2 = c,$   
 $w_3 = cb_1b_2b_2b_1a_1a_2a_2a_1cb_1b_2b_2b_1a_1a_2a_2a_1a_1a_2a_2a_1c,$   
 $w_4 = cb_1b_2b_1c,$   
 $w_5 = cb_1b_2b_2b_1a_1a_2a_2a_1a_1a_2a_2a_1c,$   
 $w_6 = c.$

**Теорема 11.3.3 (Теорема Хомского-Шютценберже).** Язык  $L \subseteq \Sigma^*$  является контекстно-свободным тогда и только тогда, когда существуют такие натуральное число  $n$ , автоматный язык  $L_1$  над алфавитом  $\Sigma_n^{(D)} = \{a_1, b_1, a_2, b_2, \dots, a_n, b_n\}$  и гомоморфизм  $h: (\Sigma_n^{(D)})^* \rightarrow \Sigma^*$ , что  $L = h(L_n^{(D)} \cap L_1)$ , где  $L_n^{(D)}$  - язык Дика над  $2n$  буквами.

Доказательство можно найти в [Лал, с. 331-333].

**Упражнение 11.3.4.** Рассмотрим язык  $L_1$ , порождаемый грамматикой

$$S \rightarrow aSbS,$$

$$S \rightarrow \varepsilon,$$

и язык  $L_2$ , порождаемый грамматикой

$$S \rightarrow bbSa,$$

$$S \rightarrow \varepsilon.$$

Найти такой гомоморфизм  $h: \{a, b\}^* \rightarrow \{a, b\}^*$ , что  $h(L_1 \cap \{a^m b^n \mid m \geq 0, n \geq 0\}) = L_2$ .

## Математическая теория формальных языков

### 12. Лекция: Детерминированные контекстно-свободные языки: версия для печати и PDA

В данной лекции рассматриваются детерминированные контекстно-свободные языки. Формулируется ряд свойств этого класса языков, рассматривается важность детерминированных контекстно-свободных языков для теоретической информатики, которая заключается в том, что для каждого такого языка можно указать быстрый алгоритм, распознающий принадлежность слова этому языку. Приведены практические примеры и упражнения для самостоятельного решения

К сожалению, теорема о детерминизации не переносится с конечных автоматов на автоматы с магазинной памятью. Возникает важный для практических приложений класс языков, распознаваемых детерминированными автоматами с магазинной памятью (то есть такими автоматами с магазинной памятью, которые ни в какой конфигурации не могут выбирать между несколькими очередными тактами). Точные определения этого класса автоматов и соответствующего класса языков даны в разделе 12.1. Чтобы получить полезный и естественный с точки зрения практики класс, нужно добавить в конец каждого слова специальный символ, называемый маркером конца слова. Языки из выделенного таким образом класса называются детерминированными контекстно-свободными языками. Во втором разделе лекции формулируется ряд свойств этого класса языков. Важность детерминированных контекстно-свободных языков для теоретической информатики обусловлена тем, что для каждого такого языка можно указать быстрый алгоритм, распознающий принадлежность слова этому языку.

#### 12.1. Детерминированные автоматы с магазинной памятью

**Определение 12.1.1.** Будем говорить, что два перехода МП-автомата

$\langle \langle p_1, x_1, \beta_1 \rangle, \langle q_1, \gamma_1 \rangle \rangle$  и  $\langle \langle p_2, x_2, \beta_2 \rangle, \langle q_2, \gamma_2 \rangle \rangle$  являются *совместными*, если

1.  $p_1 = p_2$ ;
2.  $x_1 \sqsubset x_2$  или  $x_2 \sqsubset x_1$ ;
3.  $\beta_1 \sqsubset \beta_2$  или  $\beta_2 \sqsubset \beta_1$ .

**Определение 12.1.2.** МП-автомат называется *детерминированным* (deterministic), если он имеет ровно одно начальное состояние и все переходы этого автомата попарно несовместны.

**Пример 12.1.3.** МП-автомат  $m$  из примера 10.2.8 не является детерминированным, так как переходы  $\langle \langle 2, a, \varepsilon \rangle, \langle 1, D \rangle \rangle$  и  $\langle \langle 2, \varepsilon, D \rangle, \langle 2, \varepsilon \rangle \rangle$  совместны.

**Определение 12.1.4.** Язык  $L$  над алфавитом  $\Sigma$  называется *детерминированным контекстно-свободным языком*, если существует детерминированный МП-автомат, распознающий язык  $L \cdot \{\$ \}$  над алфавитом  $\Sigma \cup \{\$ \}$ , где  $\$$  - дополнительный символ, не принадлежащий множеству  $\Sigma$ . Символ  $\$$  называется *маркером конца строки*.

**Пример 12.1.5.** Рассмотрим алфавит  $\Sigma = \{a, b\}$ . Язык

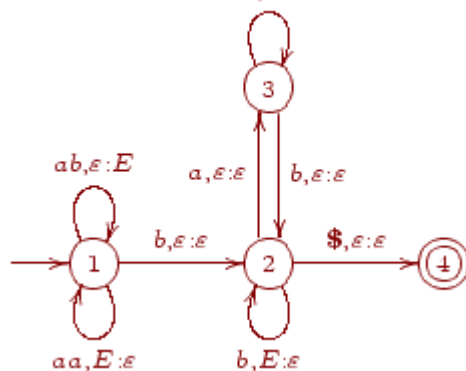
$L = \{a^m b^n \mid m = n \text{ или } n = 0\}$  - детерминированный контекстно-свободный язык, так как язык  $L \cdot \{\$ \}$  порождается детерминированным МП-автоматом (хотя язык  $L$  не порождается никаким детерминированным МП-автоматом).

**Пример 12.1.6.** Язык  $L$ , распознаваемый МП-автоматом  $M$  из примера 10.2.8, является детерминированным контекстно-свободным языком, так как язык  $L \cdot \{\$ \}$  порождается детерминированным МП-автоматом

$$M' = \langle \{1, 2, 3, 4\}, \{a, b, \$\}, \{E\}, \Delta', \{1\}, \{4\} \rangle,$$

где

$$\Delta' = \{ \langle \langle 1, ab, \varepsilon \rangle, \langle 1, E \rangle \rangle, \langle \langle 1, aa, E \rangle, \langle 1, \varepsilon \rangle \rangle, \langle \langle 1, b, \varepsilon \rangle, \langle 2, \varepsilon \rangle \rangle, \\ \langle \langle 2, b, E \rangle, \langle 2, \varepsilon \rangle \rangle, \langle \langle 2, a, \varepsilon \rangle, \langle 3, \varepsilon \rangle \rangle, \langle \langle 2, \$, \varepsilon \rangle, \langle 4, \varepsilon \rangle \rangle, \\ \langle \langle 3, ab, \varepsilon \rangle, \langle 3, E \rangle \rangle, \langle \langle 3, b, \varepsilon \rangle, \langle 2, \varepsilon \rangle \rangle \}.$$



**Упражнение 12.1.7.** Является ли детерминированным контекстно-свободный язык  $\{a^n b^{2m} ab^m a^n \mid m \geq 0, n \geq 0\}$ ?

## 12.2\*. Свойства класса детерминированных языков

**Теорема 12.2.1.** Каждый автоматный язык является детерминированным контекстно-свободным языком.

*Доказательство.* Утверждение следует из теоремы 2.7.1.

**Лемма 12.2.2.** Каждый детерминированный МП-автомат  $M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$  эквивалентен некоторому детерминированному МП-автомату

$M' = \langle Q', \Sigma, \Gamma', \Delta', I', F' \rangle$ , где для каждого перехода  $\langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta'$  выполняется неравенство  $|x| + |\beta| \leq 1$ .

*Доказательство.* Пусть дан детерминированный МП-автомат  $M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ . Назовем избытком перехода

$$\langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta$$

натуральное число  $\max(0, |x| + |\beta| - 1)$ . Докажем лемму индукцией по сумме избытков всех переходов.

Шаг индукции. Пусть существует переход

$$\langle \langle p_0, x_0, \beta_0 \rangle, \langle q_0, \gamma_0 \rangle \rangle \in \Delta$$

с положительным избытком. Рассмотрим четыре случая.

Случай 1.  $\Delta \cap ((\{p_0\} \times \{\varepsilon\} \times \Gamma^*) \times (Q \times \Gamma^*)) = \emptyset$ . Обозначим первый символ слова  $x_0$  через  $a_0$ . Преобразуем МП-автомат  $M$  в эквивалентный ему детерминированный МП-автомат с меньшей суммой избытков всех переходов. Для этого добавим новое состояние  $r$  и переход  $\langle \langle p_0, a_0, \varepsilon \rangle, \langle r, \varepsilon \rangle \rangle$ . Каждый переход вида

$\langle\langle p_0, a_0 x, \beta \rangle, \langle q, \gamma \rangle\rangle$  заменим на переход  $\langle\langle r, x, \beta \rangle, \langle q, \gamma \rangle\rangle$ . К полученному таким образом детерминированному МП-автомату применимо предположение индукции.

Случай 2.  $\Delta \cap ((\{p_0\} \times \Sigma^* \times \{\varepsilon\}) \times (Q \times \Gamma^*)) = \emptyset$ . Обозначим первый символ слова  $\beta_0$  через  $c_0$ . Преобразуем МП-автомат  $M$  в эквивалентный ему детерминированный МП-автомат с меньшей суммой избытков всех переходов. Для этого добавим новое состояние  $r$  и переход  $\langle\langle p_0, \varepsilon, c_0 \rangle, \langle r, \varepsilon \rangle\rangle$ . Каждый переход вида  $\langle\langle p_0, x, c_0 \beta \rangle, \langle q, \gamma \rangle\rangle$  заменим на переход  $\langle\langle r, x, \beta \rangle, \langle q, \gamma \rangle\rangle$ . К полученному таким образом детерминированному МП-автомату применимо предположение индукции.

Случай 3. Существует переход  $\langle\langle p_0, \varepsilon, \varepsilon \rangle, \langle q_1, \gamma_1 \rangle\rangle \in \Delta$ . Тогда переходы  $\langle\langle p_0, x_0, \beta_0 \rangle, \langle q_0, \gamma_0 \rangle\rangle$  и  $\langle\langle p_0, \varepsilon, \varepsilon \rangle, \langle q_1, \gamma_1 \rangle\rangle$  совместны. Противоречие.

Случай 4. Существуют переход  $\langle\langle p_0, \varepsilon, \beta_1 \rangle, \langle q_1, \gamma_1 \rangle\rangle \in \Delta$  и переход  $\langle\langle p_0, x_2, \varepsilon \rangle, \langle q_2, \gamma_2 \rangle\rangle \in \Delta$ , где  $\beta_1 \neq \varepsilon$  и  $x_2 \neq \varepsilon$ . Тогда переходы  $\langle\langle p_0, \varepsilon, \beta_1 \rangle, \langle q_1, \gamma_1 \rangle\rangle$  и  $\langle\langle p_0, x_2, \varepsilon \rangle, \langle q_2, \gamma_2 \rangle\rangle$  совместны. Противоречие.

**Лемма 12.2.3.** Каждый детерминированный МП-автомат  $M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$  эквивалентен некоторому детерминированному МП-автомату

$M' = \langle Q', \Sigma, \Gamma', \Delta', I', F' \rangle$ , где каждый переход  $\langle\langle p, x, \beta \rangle, \langle q, \gamma \rangle\rangle \in \Delta'$  удовлетворяет условиям  $|x| + |\beta| \leq 1$  и  $|\beta| + |\gamma| = 1$ .

*Доказательство.* Сначала применим лемму 12.2.2, затем преобразуем МП-автомат так, чтобы для каждого перехода  $\langle\langle p, x, \beta \rangle, \langle q, \gamma \rangle\rangle$  выполнялось неравенство  $|x| + |\beta| + |\gamma| \leq 1$  (см. пример 10.2.4), и в заключение заменим каждый переход вида  $\langle\langle p, x, \varepsilon \rangle, \langle q, \varepsilon \rangle\rangle$  на два последовательных перехода (см. пример 10.2.5).

**Лемма 12.2.4.** Пусть  $\$ \notin \Sigma$ ,  $L \subseteq \Sigma^*$  и язык  $L \cdot \{\$ \}$  порождается некоторым детерминированным МП-автоматом. Тогда этот язык порождается также некоторым детерминированным МП-автоматом  $\langle Q, \Sigma \cup \{\$ \}, \Gamma, \Delta, I, F \rangle$ , где  $Q = Q_1 \cup Q_2$ ,

$$Q_1 \cap Q_2 = \emptyset,$$

$$\Delta = \Delta_{11} \cup \Delta_{12} \cup \Delta_{22},$$

$$\Delta_{11} \subseteq (Q_1 \times \Sigma^* \times \Gamma^*) \times (Q_1 \times \Gamma^*),$$

$$\Delta_{12} \subseteq (Q_1 \times \{\$ \} \times \Gamma^*) \times (Q_2 \times \Gamma^*),$$

$$\Delta_{22} \subseteq (Q_2 \times \{\varepsilon\} \times \Gamma^*) \times (Q_2 \times \Gamma^*),$$

$$I \subseteq Q_1, F \subseteq Q_2 \text{ и каждый переход}$$

$$\langle\langle p, x, \beta \rangle, \langle q, \gamma \rangle\rangle \in \Delta$$

удовлетворяет условиям  $|x| + |\beta| \leq 1$  и  $|\beta| + |\gamma| = 1$ .

*Доказательство.* Применив лемму 12.2.3, получим детерминированный МП-автомат

$$\langle Q', \Sigma \cup \{\$ \}, \Gamma', \Delta', I', F' \rangle.$$

$$\langle Q, \Sigma \cup \{\$ \}, \Gamma, \Delta, I, F \rangle, \text{ положив } Q = Q' \times \{1, 2\}, Q_1 = Q' \times \{1\},$$

$$Q_2 = Q' \times \{2\}, \Gamma = \Gamma', I = I' \times \{1\}, F = F' \times \{2\},$$

$$\begin{aligned}\Delta_{11} &= \{ \langle \langle p, 1 \rangle, x, \beta \rangle, \langle \langle q, 1 \rangle, \gamma \rangle \mid \langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta', x \in \Sigma^* \}, \\ \Delta_{12} &= \{ \langle \langle p, 1 \rangle, \$, \beta \rangle, \langle \langle q, 2 \rangle, \gamma \rangle \mid \langle \langle p, \$, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta' \}, \\ \Delta_{22} &= \{ \langle \langle p, 2 \rangle, \varepsilon, \beta \rangle, \langle \langle q, 2 \rangle, \gamma \rangle \mid \langle \langle p, \varepsilon, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta' \}.\end{aligned}$$

**Теорема 12.2.5.** Язык  $L \subseteq \Sigma^*$  является детерминированным контекстно-свободным языком тогда и только тогда, когда найдется такой детерминированный МП-автомат

$$M' = \langle Q', \Sigma, \Gamma', \Delta', I', F' \rangle, \text{ что } L = \{ w \in \Sigma^* \mid \langle s, w, \varepsilon \rangle \stackrel{*}{\vdash} \langle q, \varepsilon, \alpha \rangle \text{ для некоторых } s \in I', q \in F', \alpha \in \Gamma'^* \}.$$

*Доказательство.* Достаточность проверяется легко. Докажем необходимость. Рассмотрим МП-автомат

$$M = \langle Q_1 \cup Q_2, \Sigma \cup \{ \$ \}, \Gamma, \Delta_{11} \cup \Delta_{12} \cup \Delta_{22}, \{ q_s \}, F \rangle,$$

о котором говорится в лемме 12.2.4.

Для любых  $\alpha \in \Gamma^*$  и  $H \subseteq Q_2$  введем обозначение

$$\Delta_{22}^{-1}(\alpha, H) = \{ p \in Q_2 \mid \langle p, \varepsilon, \alpha \rangle \stackrel{*}{\vdash} \langle q, \varepsilon, \varepsilon \rangle \text{ для некоторого } q \in H \}.$$

Очевидно, что  $\Delta_{22}^{-1}(\gamma, \Delta_{22}^{-1}(\alpha, H)) = \Delta_{22}^{-1}(\gamma\alpha, H)$  для любых  $\alpha \in \Gamma^*, \gamma \in \Gamma^*$  и  $H \subseteq Q_2$ .

Построим искомый МП-автомат  $\langle Q', \Sigma, \Gamma', \Delta', I', F' \rangle$ , положив

$$\begin{aligned}Q' &= Q_1 \times \mathcal{P}(Q_2), \\ \Gamma' &= \Gamma \times \mathcal{P}(Q_2), \\ I' &= \{ \langle q_s, \Delta_{22}^{-1}(\varepsilon, F) \rangle \}, \\ F' &= \{ \langle p, H \rangle \mid \langle \langle p, \$, \varepsilon \rangle, \langle q, C \rangle \rangle \in \Delta_{12}, H \subseteq Q_2, q \in \Delta_{22}^{-1}(C, H) \}, \\ \Delta' &= \{ \langle \langle \langle p, H \rangle, x, \langle C, K \rangle \rangle, \langle \langle q, K \rangle, \varepsilon \rangle \rangle \mid \\ &\quad \langle \langle p, x, C \rangle, \langle q, \varepsilon \rangle \rangle \in \Delta_{11}, H \subseteq Q_2, K \subseteq Q_2 \} \cup \\ &\quad \cup \{ \langle \langle \langle p, H \rangle, x, \varepsilon \rangle, \langle \langle q, \Delta_{22}^{-1}(C, H) \rangle, \langle C, H \rangle \rangle \rangle \mid \\ &\quad \langle \langle p, x, \varepsilon \rangle, \langle q, C \rangle \rangle \in \Delta_{11}, H \subseteq Q_2 \}.\end{aligned}$$

(Напоминаем, что через  $\mathcal{P}(Q_2)$  обозначается множество всех подмножеств множества  $Q_2$ .)

Индукцией по количеству тактов работы можно доказать, что

$$\langle \langle q_s, \Delta_{22}^{-1}(\varepsilon, F) \rangle, w, \varepsilon \rangle \stackrel{*}{\vdash}_{M'} \langle \langle q, H_0 \rangle, \varepsilon, \langle C_1, H_1 \rangle \langle C_2, H_2 \rangle \dots \langle C_n, H_n \rangle \rangle$$

тогда и только тогда, когда

$$\langle q_s, w, \varepsilon \rangle \stackrel{*}{\vdash}_M \langle q, \varepsilon, C_1 C_2 \dots C_n \rangle$$

и

$$H_k = \Delta_{22}^{-1}(C_{k+1} C_{k+2} \dots C_n, F)$$

для каждого  $k \leq n$ .

**Теорема 12.2.6.** Пусть  $L$  - детерминированный контекстно-свободный язык. Тогда язык  $L$  не является существенно неоднозначным.

*Доказательство.* Пусть дан детерминированный контекстно-свободный язык  $L$ . Рассмотрим МП-автомат

$$M = \langle Q_1 \cup Q_2, \Sigma \cup \{\$, \Gamma, \Delta_{11} \cup \Delta_{12} \cup \Delta_{22}, \{q_s\}, F \rangle,$$

о котором говорится в лемме 12.2.4. Применив к этому МП-автомату конструкцию из доказательства теоремы 10.2.7, получим однозначную контекстно-свободную

грамматику  $G$ , порождающую язык  $L \cdot \{\$\}$ . Стирая в этой грамматике все вхождения символа  $\$$ , получим контекстно-свободную грамматику  $G'$ , порождающую язык  $L$ .

Так как МП-автомат  $M$  не содержит переходов, ведущих из  $Q_2$  в  $Q_1$ , а символ  $\$$  встречается только на переходах, ведущих из  $Q_1$  в  $Q_2$ , то в каждом  $G'$ -выводе однозначно определяется правило, которому в  $G$  соответствует правило, содержащее  $\$$ . Поэтому существует естественное однозначное соответствие между деревьями вывода в грамматике  $G'$  и деревьями вывода в грамматике  $G$  (при этом кроны соответствующих друг другу деревьев различаются только символом  $\$$ ). Следовательно, грамматика  $G'$  тоже является однозначной.

**Теорема 12.2.7.** Дополнение каждого детерминированного контекстно-свободного языка является детерминированным контекстно-свободным языком.

Доказательство можно найти в [7, с. 110-116] или [2, с. 212-217].

**Пример 12.2.8.** Язык  $L = \{a^k b^m c^n \mid k \neq m \text{ или } m \neq n\}$  над алфавитом  $\{a, b, c\}$  не является детерминированным контекстно-свободным языком, так как его дополнение не является контекстно-свободным.

**Теорема 12.2.9.** Неверно, что для любых детерминированных контекстно-свободных языков  $L_1$  и  $L_2$  язык  $L_1 \cap L_2$  тоже является детерминированным контекстно-свободным языком.

*Доказательство.* Достаточно рассмотреть детерминированные контекстно-свободные языки  $L_1$  и  $L_2$  из доказательства теоремы 9.5.1.

**Теорема 12.2.10.** Неверно, что для любых детерминированных контекстно-свободных языков  $L_1$  и  $L_2$  язык  $L_1 \cup L_2$  тоже является детерминированным контекстно-свободным языком.

*Доказательство.* Утверждение следует из теорем 12.2.7 и 12.2.9 и закона де Моргана.

**Упражнение 12.2.11.** Является ли детерминированным контекстно-свободным язык  $\{a^k b^m c^n \mid k < \max(m, n)\}$  ?

## Математическая теория формальных языков

### 13. Лекция: Синтаксический разбор: версия для печати и PDA

В данной лекции рассматриваются формальные определения, связанные с прямыми синтаксическими анализаторами, определяется понятие синтаксического разбора и связанные с ним специфические определения. Приведены практические примеры и упражнения для самостоятельной проработки

В этой лекции даются формальные определения, связанные с прямыми (то есть читающими входную строку слева направо) синтаксическими анализаторами. В первом разделе доказывается, что для языков из класса  $LL(1)$  можно построить основанный на детерминированном автомате с магазинной памятью анализатор, который создает дерево разбора, двигаясь снизу вверх, то есть от листьев к корню (в теории контекстно-свободных грамматик принято изображать деревья с корнем наверху). Во втором разделе формулируется аналогичный результат об анализе сверху вниз для грамматик из класса  $LR(1)$ . При этом понятие анализа снизу вверх формализовано в терминах последовательности правил, примененных в левостороннем выводе, а понятие анализа сверху вниз - в терминах обращенной последовательности правил, примененных в правостороннем выводе.

#### 13.1. Нисходящий разбор

**Определение 13.1.1.** Процесс нахождения дерева вывода слова  $w$  в заданной контекстно-свободной грамматике называется *синтаксическим разбором* или *синтаксическим анализом* (parsing).

**Определение 13.1.2.** Протоколом левостороннего вывода в контекстно-свободной грамматике  $G = \langle N, \Sigma, P, S \rangle$  будем называть последовательность правил, примененных в этом выводе. Формально говоря, протоколом левостороннего вывода  $\omega_0, \omega_1, \dots, \omega_n$

является последовательность

$$(A_1 \rightarrow \alpha_1), \dots, (A_n \rightarrow \alpha_n),$$

где для каждого  $i < n$ , если  $\omega_i = uB\theta$  и  $\omega_{i+1} = u\beta\theta$  для некоторых  $u \in \Sigma^*$ ,

$B \in N$ ,  $\theta \in (N \cup \Sigma)^*$ ,  $\beta \in (N \cup \Sigma)^*$ , то  $A_{i+1} = B$  и  $\alpha_{i+1} = \beta$ .

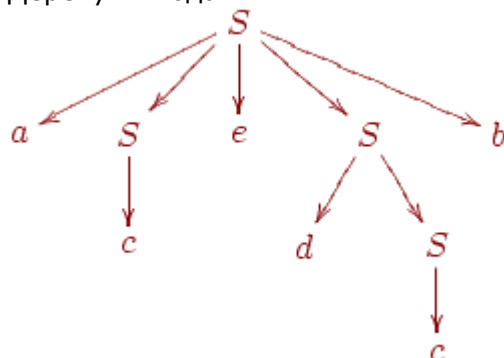
**Пример 13.1.3.** Рассмотрим контекстно-свободную грамматику

$$S \rightarrow c,$$

$$S \rightarrow dS,$$

$$S \rightarrow aSeSb.$$

Дереву вывода



соответствует левосторонний вывод

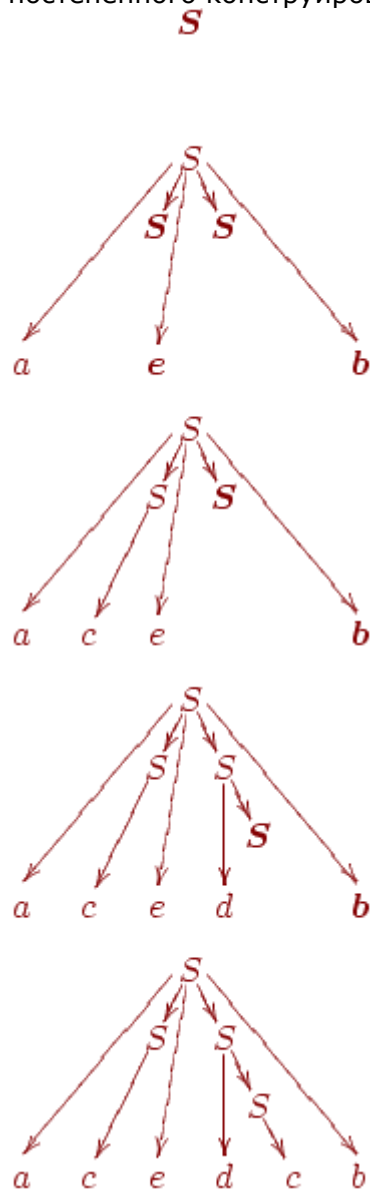
$$S \Rightarrow aSeSb \Rightarrow aceSb \Rightarrow acedSb \Rightarrow acedcb.$$

Протоколом этого вывода является последовательность  
 $(S \rightarrow aSeSb), (S \rightarrow c), (S \rightarrow dS), (S \rightarrow c)$ .

**Лемма 13.1.4.** Разным левосторонним выводам в одной и той же контекстно-свободной грамматике соответствуют разные протоколы.

**Замечание 13.1.5.** Протокол левостороннего вывода в контекстно-свободной грамматике является естественным описанием соответствующего дерева вывода в порядке префиксного обхода (preorder traversal). (При префиксном обходе упорядоченного дерева первым посещается корень этого дерева, затем выполняется префиксный обход первого непосредственного потомка корня, затем второго и т. д.)

Например, протокол левостороннего вывода из примера 13.1.3 задает процесс постепенного конструирования дерева вывода, изображенный ниже.



**Определение 13.1.6.** Левым разбором (left parse) слова  $w$  в контекстно-свободной грамматике  $G$  называется протокол любого левостороннего вывода слова  $w$  в грамматике  $G$ .

**Пример 13.1.7.** Левым разбором слова

acecbcb



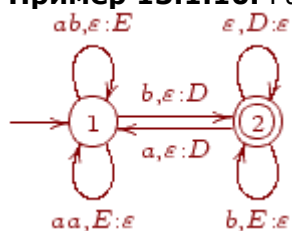
в грамматике из примера 13.1.3 является последовательность

$(S \rightarrow aSeSb), (S \rightarrow c), (S \rightarrow aSeSb),$   
 $(S \rightarrow aSeSb), (S \rightarrow c), (S \rightarrow c), (S \rightarrow c).$

**Определение 13.1.8.** Процесс нахождения левого разбора слова  $w$  в заданной контекстно-свободной грамматике  $G$  называется *нисходящим разбором* (top-down parsing).

**Определение 13.1.9.** Вычислительным процессом МП-автомата  $M$  будем называть конечную последовательность его конфигураций, каждая из которых (кроме первой) получается из предыдущей одним тактом работы автомата  $M$ .

**Пример 13.1.10.** Рассмотрим МП-автомат



из примера 10.2.8. Последовательность

$\langle 1, bab, \varepsilon \rangle, \langle 2, ab, D \rangle, \langle 1, b, DD \rangle, \langle 2, \varepsilon, DDD \rangle,$   
 $\langle 2, \varepsilon, DD \rangle, \langle 2, \varepsilon, D \rangle, \langle 2, \varepsilon, \varepsilon \rangle$

является вычислительным процессом этого МП-автомата.

**Определение 13.1.11.** Если в некотором вычислительном процессе МП-автомата

$M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$  первая конфигурация имеет вид  $\langle s, w, \varepsilon \rangle$ , где  $s \in I$  и  $w \in \Sigma^*$ , а последняя конфигурация имеет вид  $\langle q, \varepsilon, \varepsilon \rangle$ , где  $q \in F$ , то будем говорить, что этот вычислительный процесс *допускает* слово  $w$ .

**Пример 13.1.12.** Вычислительный процесс из примера 13.1.10 допускает слово  $bab$ .

**Замечание 13.1.13.** МП-автомат  $M$  допускает слово  $w \in \Sigma^*$  тогда и только тогда, когда некоторый вычислительный процесс МП-автомата  $M$  допускает слово  $w$ .

**Определение 13.1.14.** *Протоколом* вычислительного процесса МП-Автомата

$M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$  будем называть последовательность переходов, примененных в этом вычислительном процессе. Формально говоря, протоколом вычислительного процесса

$\langle q_0, w_0, \alpha_0 \rangle, \langle q_1, w_1, \alpha_1 \rangle, \dots, \langle q_n, w_n, \alpha_n \rangle$

является последовательность переходов

$\langle \langle q_0, x_1, \beta_1 \rangle, \langle q_1, \gamma_1 \rangle \rangle, \dots, \langle \langle q_{n-1}, x_n, \beta_n \rangle, \langle q_n, \gamma_n \rangle \rangle,$

где для каждого  $i$  между 1 и  $n$  слово  $x_i$  определяется равенством  $w_{i-1} = x_i w_i$ , а

через  $\beta_i$  и  $\gamma_i$  обозначены кратчайшие слова из  $\Gamma^*$ , удовлетворяющие условиям

$\langle \langle q_{i-1}, x_i, \beta_i \rangle, \langle q_i, \gamma_i \rangle \rangle \in \Delta,$

$\alpha_{i-1} = \beta_i \zeta,$

$\alpha_i = \gamma_i \zeta$

для некоторого слова  $\zeta \in \Gamma^*$ .

**Пример 13.1.15.** Протоколом вычислительного процесса из примера 13.1.10 является последовательность

$\langle\langle 1, b, \varepsilon \rangle, \langle 2, D \rangle\rangle, \langle\langle 2, a, \varepsilon \rangle, \langle 1, D \rangle\rangle, \langle\langle 1, b, \varepsilon \rangle, \langle 2, D \rangle\rangle,$   
 $\langle\langle 2, \varepsilon, D \rangle, \langle 2, \varepsilon \rangle\rangle, \langle\langle 2, \varepsilon, D \rangle, \langle 2, \varepsilon \rangle\rangle, \langle\langle 2, \varepsilon, D \rangle, \langle 2, \varepsilon \rangle\rangle.$

**Определение 13.1.16.** Для каждой контекстно-свободной грамматики

$G = \langle N, \Sigma, P, S \rangle$  обозначим через  $G_\$$  контекстно-свободную грамматику  $\langle N \cup \{S_\$\}, \Sigma \cup \{\$, \}, P_\$, S_\$ \rangle$ , где  $\$$  и  $S_\$$  - два различных новых символа, не принадлежащие множеству  $N \cup \Sigma$ , и  $P_\$ = P \cup \{S_\$ \rightarrow S\ \$\}$ . Грамматику  $G_\$$  будем называть *грамматикой с маркером конца строки*. Терминальный алфавит грамматики  $G_\$$  (то есть множество  $\Sigma \cup \{\ \$ \}$ ) будем обозначать через  $\Sigma_\$$ . Нетерминальный алфавит  $N \cup \{S_\$\}$  будем обозначать через  $N_\$$ .

**Пример 13.1.17.** Рассмотрим контекстно-свободную грамматику  $G$  с терминальным алфавитом  $\Sigma = \{m, n, +, -, *, /, (, ), \$\}$ , вспомогательным алфавитом  $N = \{S, A, B, C\}$  и правилами

$A \rightarrow A+B,$	$B \rightarrow B*C,$	$C \rightarrow (A),$
$A \rightarrow A-B,$	$B \rightarrow B/C,$	$C \rightarrow m,$
$A \rightarrow B,$	$B \rightarrow C,$	$C \rightarrow n.$

Ей соответствует следующая грамматика  $G_\$$  с маркером конца строки:

$T \rightarrow A\ \$,$	$B \rightarrow B*C,$	$C \rightarrow (A),$
$A \rightarrow A+B,$	$B \rightarrow B/C,$	$C \rightarrow m,$
$A \rightarrow A-B,$	$B \rightarrow C,$	$C \rightarrow n.$
$A \rightarrow B,$		

**Замечание 13.1.18.** Очевидно, что  $L(G_\$) = L(G) \cdot \{\ \$ \}$ .

**Лемма 13.1.19.** Если  $S_\$ \xrightarrow{*}_{\Sigma_\$} \eta \$ \theta$ , где  $\eta \in (N \cup \Sigma)^*$  и  $\theta \in (N \cup \Sigma)^*$ , то  $\theta = \varepsilon$ .

*Доказательство.* Очевидно, что если  $S_\$ \xrightarrow{*}_{\Sigma_\$} \omega$ , то слово  $\omega$  не содержит символа  $S_\$$ .

Теперь докажем индукцией по длине вывода, что если  $S_\$ \xrightarrow{*}_{\Sigma_\$} \eta \$ \theta$ , то  $\theta = \varepsilon$ . Пусть на последнем шаге в этом выводе применялось правило  $(A \rightarrow \alpha) \in P$ .

Если  $|\alpha|_\$ > 0$ , то  $A = S_\$$ , что невозможно (очевидно, что если  $S_\$ \xrightarrow{*}_{\Sigma_\$} \omega$ , то слово  $\omega$  не содержит символа  $S_\$$ ).

Если  $|\alpha|_\$ = 0$ , то рассмотрим два случая. При

$S_\$ \xrightarrow{*}_{\Sigma_\$} \eta_1 A \eta_2 \$ \theta \Rightarrow \eta_1 \alpha \eta_2 \$ \theta$

равенство  $\theta = \varepsilon$  обеспечивается предположением индукцией. Случай

$S_\$ \xrightarrow{*}_{\Sigma_\$} \eta \$ \theta_1 A \theta_2 \Rightarrow \eta \$ \theta_1 \alpha \theta_2$

невозможен, так как предположение индукции дает  $\theta_1 A \theta_2 = \varepsilon$ .

**Определение 13.1.20.** Пусть даны контекстно-свободная грамматика

$G = \langle N, \Sigma, P, S \rangle$  и МП-автомат  $M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ . Будем говорить, что МП-автомат  $M$  - нисходящий магазинный анализатор (или предсказывающий анализатор, top-down, left-to-right parser, predictive parser) для грамматики  $G$ , если  $L(M) = L(G_\$)$  и существует такой гомоморфизм  $r: \Delta^* \rightarrow P_\$^*$ , что для каждого вычислительного процесса (МП-автомата  $M$ ), допускающего слово  $w \in \Sigma^*$ , образ протокола этого вычислительного процесса при гомоморфизме  $r$  является протоколом некоторого левостороннего вывода слова  $w$  в грамматике  $G_\$$ . (При задании гомоморфизма  $r$  конечные множества  $\Delta$  и  $P_\$$  рассматриваются как два алфавита.)

**Пример 13.1.21.** Рассмотрим контекстно-свободную грамматику  $G_\$$  из примера 13.1.17. Язык  $L(G_\$)$  распознается недетерминированным МП-автоматом

$M = \langle \{1, 2\}, \Sigma, \Gamma, \Delta, \{1\}, \{2\} \rangle$ , где

$\Gamma = \{A, B, C, +, -, *, /, ), \$\}$

и

$\Delta = \{ \langle \langle 1, \epsilon, \epsilon \rangle, \langle 2, A\$ \rangle \rangle,$

$\langle \langle 2, \epsilon, A \rangle, \langle 2, A+B \rangle \rangle, \langle \langle 2, \epsilon, A \rangle, \langle 2, A-B \rangle \rangle, \langle \langle 2, \epsilon, A \rangle, \langle 2, B \rangle \rangle,$

$\langle \langle 2, \epsilon, B \rangle, \langle 2, B*C \rangle \rangle, \langle \langle 2, \epsilon, B \rangle, \langle 2, B/C \rangle \rangle, \langle \langle 2, \epsilon, B \rangle, \langle 2, C \rangle \rangle,$

$\langle \langle 2, (, C \rangle, \langle 2, A \rangle \rangle, \langle \langle 2, m, C \rangle, \langle 2, \epsilon \rangle \rangle, \langle \langle 2, n, C \rangle, \langle 2, \epsilon \rangle \rangle,$

$\langle \langle 2, +, + \rangle, \langle 2, \epsilon \rangle \rangle,$

$\langle \langle 2, -, - \rangle, \langle 2, \epsilon \rangle \rangle,$

$\langle \langle 2, *, * \rangle, \langle 2, \epsilon \rangle \rangle,$

$\langle \langle 2, /, / \rangle, \langle 2, \epsilon \rangle \rangle,$

$\langle \langle 2, ), ) \rangle, \langle 2, \epsilon \rangle \rangle,$

$\langle \langle 2, \$, \$ \rangle, \langle 2, \epsilon \rangle \rangle \}$ .

Легко убедиться, что МП-автомат  $M$  является нисходящим магазинным анализатором для грамматики  $G$  из примера 13.1.17.

**Определение 13.1.22.** Сентенциальной формой (sentential form) грамматики

$G = \langle N, \Sigma, P, S \rangle$  называется любое слово в алфавите  $N \cup \Sigma$ , выводимое из начального символа  $S$ .

**Пример 13.1.23.** Слова  $S$ ,  $aSeaceSbb$ ,  $aceaacecbeccbb$  являются сентенциальными формами грамматики из примера 13.1.3.

**Определение 13.1.24.** Пусть дана контекстно-свободная грамматика

$G = \langle N, \Sigma, P, S \rangle$ . Определим три функции  $FIRST_G$ ,  $FOLLOW_G$  и

$DIRECTOR_G$ , связанные с грамматикой  $G$ . Для краткости будем писать просто  $FIRST$ ,  $FOLLOW$  и  $DIRECTOR$ .

Функция  $FIRST$  ставит в соответствие каждому слову  $\omega \in (N \cup \Sigma)^*$  множество тех терминальных символов, с которых начинаются слова, выводимые из  $\omega$ , то есть

$FIRST(\omega) = \{b \in \Sigma \mid (\exists \theta \in (N \cup \Sigma)^*) \omega \xRightarrow{*} b\theta\}$ .

Функция FOLLOW ставит в соответствие каждому нетерминальному символу  $A$  множество тех терминальных символов, которые могут встречаться в сентенциальных формах непосредственно справа от  $A$ , то есть

$$\text{FOLLOW}(A) = \{b \in \Sigma \mid (\exists \eta \in (N \cup \Sigma)^*) (\exists \theta \in (N \cup \Sigma)^*) S \xRightarrow{*} \eta A b \theta\}.$$

Функция DIRECTOR ставит в соответствие каждому правилу  $(A \rightarrow \alpha) \in P$  множество терминальных символов, определяемое следующим образом: если  $\alpha \xRightarrow{*} \varepsilon$ , то

$$\text{DIRECTOR}(A \rightarrow \alpha) = \text{FIRST}(\alpha) \cup \text{FOLLOW}(A),$$

иначе

$$\text{DIRECTOR}(A \rightarrow \alpha) = \text{FIRST}(\alpha).$$

**Пример 13.1.25.** Рассмотрим контекстно-свободную грамматику из примера 13.1.3. Очевидно, что

$$\text{FIRST}(c) = \{c\},$$

$$\text{FIRST}(dS) = \{d\},$$

$$\text{FIRST}(aSeSb) = \{a\},$$

$$\text{FOLLOW}(S) = \{b, e\},$$

$$\text{DIRECTOR}(S \rightarrow c) = \{c\},$$

$$\text{DIRECTOR}(S \rightarrow dS) = \{d\},$$

$$\text{DIRECTOR}(S \rightarrow aSeSb) = \{a\}.$$

**Пример 13.1.26.** Рассмотрим контекстно-свободную грамматику  $G_{\$}$  из примера 13.1.17. Очевидно, что

$$\text{DIRECTOR}(B \rightarrow C) = \{(\,, m, n\},$$

$$\text{DIRECTOR}(B \rightarrow B/C) = \{(\,, m, n\},$$

$$\text{DIRECTOR}(B \rightarrow B*C) = \{(\,, m, n\}.$$

**Пример 13.1.27.** Пусть контекстно-свободная грамматика  $G = \langle N, \Sigma, P, S \rangle$  не содержит бесполезных символов. Пусть даны правило  $(A \rightarrow \alpha) \in P$  и символ  $b \in \Sigma$ . Тогда утверждение

$$b \in \text{DIRECTOR}(A \rightarrow \alpha)$$

равносильно тому, что найдутся такие  $x \in \Sigma^*$ ,  $y \in \Sigma^*$  и  $\theta \in (N \cup \Sigma)^*$ , что  $S \xRightarrow{*} xA\theta$  и  $x\alpha\theta \xRightarrow{*} xby$ .

**Теорема 13.1.28.** Пусть дана контекстно-свободная грамматика  $G = \langle N, \Sigma, P, S \rangle$ .

Пусть  $G_{\$} = \langle N_{\$}, \Sigma_{\$}, P_{\$}, S_{\$} \rangle$  - соответствующая контекстно-свободная грамматика с маркером конца строки, приведенная в определении 13.1.16. Обозначим через  $M$  МП-

автомат  $\langle Q, \Sigma_{\$}, \Gamma, \Delta, \{q_s\}, \{\varepsilon\} \rangle$ , где

$$Q = \{v \in \Sigma_{\$}^* \mid |v| \leq 1\} \cup \{q_s\},$$

$$q_s \notin \Sigma_{\$}, \Gamma = N_{\$} \cup \Sigma_{\$} \text{ и}$$

$$\begin{aligned}\Delta = & \{ \langle \langle q_s, \varepsilon, \varepsilon \rangle, \langle \varepsilon, S_\$ \rangle \rangle \} \cup \\ & \cup \{ \langle \langle \varepsilon, a, \varepsilon \rangle, \langle a, \varepsilon \rangle \rangle \mid a \in \Sigma_\$ \} \cup \\ & \cup \{ \langle \langle a, \varepsilon, a \rangle, \langle \varepsilon, \varepsilon \rangle \rangle \mid a \in \Sigma_\$ \} \cup \\ & \cup \{ \langle \langle b, \varepsilon, A \rangle, \langle b, \alpha \rangle \rangle \mid (A \rightarrow \alpha) \in P_\$, b \in \text{DIRECTOR}_{G_\$}(A \rightarrow \alpha) \}.\end{aligned}$$

Тогда МП-автомат  $M$  является нисходящим магазинным анализатором для грамматики  $G$ .

*Доказательство.* Индукцией по количеству тактов можно доказать, что если

$$\langle v, w, \beta \rangle \vdash^* \langle \varepsilon, \varepsilon, \varepsilon \rangle, \text{ где } v \in \Sigma_\$^*, \text{ то } \beta \xRightarrow[\text{Im}]{*} vw. \text{ Следовательно, } L(M) \subseteq L(G_\$).$$

С другой стороны, докажем, что если в грамматике  $G_\$$  выводится  $S_\$ \xRightarrow[\text{Im}]{*} u\beta$  и  $\beta \xRightarrow[\text{Im}]{*} bw$ , где  $u \in \Sigma_\$^*$ ,  $\beta \in (N \cup \Sigma)^*$ ,  $b \in \Sigma_\$, w \in \Sigma_\$^*$ , то  $\langle b, w, \beta \rangle \vdash^* \langle \varepsilon, \varepsilon, \varepsilon \rangle$ .

Проведем доказательство индукцией по сумме длины слова  $bw$  и длины вывода

$$\begin{aligned}\beta \xRightarrow[\text{Im}]{*} bw. \text{ Случай } |bw| = 1, \beta = bw \text{ образует базис индукции (очевидно, что} \\ \langle b, \varepsilon, b \rangle \vdash^* \langle \varepsilon, \varepsilon, \varepsilon \rangle). \text{ Проверим теперь шаг индукции. Так как } \beta \xRightarrow[\text{Im}]{*} bw \text{ и } bw \neq \varepsilon, \text{ то} \\ \beta \neq \varepsilon. \text{ Если } \beta = A\theta, \text{ где } A \in N, \text{ то вывод } \beta \xRightarrow[\text{Im}]{*} bw \text{ имеет вид} \\ A\theta \xRightarrow[\text{Im}]{*} \alpha\theta \xRightarrow[\text{Im}]{*} bw,\end{aligned}$$

где  $(A \rightarrow \alpha) \in P_\$$  и в силу леммы 13.1.27  $b \in \text{DIRECTOR}_{G_\$}(A \rightarrow \alpha)$ . Отсюда получаем, что  $\langle b, w, A\theta \rangle \vdash \langle b, w, \alpha\theta \rangle$ , и остается применить предположение индукции для левосторонних выводов  $S_\$ \xRightarrow[\text{Im}]{*} u\alpha\theta$  и  $\alpha\theta \xRightarrow[\text{Im}]{*} bw$ . Если же  $\beta = a\beta'$ , где  $a \in \Sigma_\$, то, очевидно, a = b и \beta' \xRightarrow[\text{Im}]{*} w. Если w = \varepsilon, то b = \$ и в силу леммы 13.1.19 \beta' = \varepsilon, но этот случай уже рассмотрен в базисе индукции. Пусть w \neq \varepsilon. Тогда w = b'w' для некоторых b' \in \Sigma_\$ и w' \in \Sigma_\$^*. Обозначим u' = ub. Применяя предположение$

индукции для левосторонних выводов  $S_\$ \xRightarrow[\text{Im}]{*} u'\beta'$  и  $\beta' \xRightarrow[\text{Im}]{*} b'w'$ , получаем

$$\begin{aligned}\langle b', w', \beta' \rangle \vdash^* \langle \varepsilon, \varepsilon, \varepsilon \rangle. \text{ Согласно построению МП-автомата } M \text{ имеем} \\ \langle b, b'w', b\beta' \rangle \vdash \langle \varepsilon, b'w', \beta' \rangle \vdash \langle b', w', \beta' \rangle.\end{aligned}$$

Шаг индукции доказан. Теперь легко проверить, что выполняется  $L(G_\$) \subseteq L(M)$ .

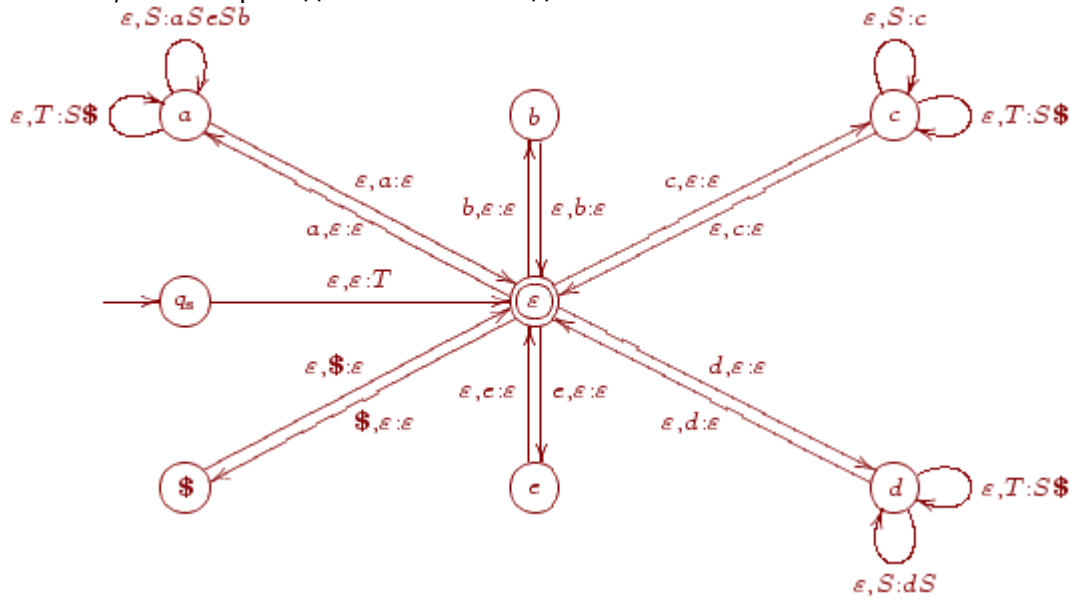
Гомоморфизм  $r: \Delta^* \rightarrow P_\$^*$  задается соотношениями

$$\begin{aligned}r(\langle \langle b, \varepsilon, A \rangle, \langle b, \alpha \rangle \rangle) &= (A \rightarrow \beta), \\ r(\delta) &= \varepsilon \text{ для остальных } \delta \in \Delta.\end{aligned}$$

**Пример 13.1.29.** Рассмотрим контекстно-свободную грамматику из примера 13.1.3. Грамматика с маркером конца строки имеет правила

$$\begin{aligned}T &\rightarrow S\$, \\ S &\rightarrow c, \\ S &\rightarrow dS, \\ S &\rightarrow aSeSb.\end{aligned}$$

Соответствующий нисходящий магазинный анализатор  $M$ , построенный в теореме 13.1.28, имеет приведенный ниже вид.



**Определение 13.1.30.** Контекстно-свободная грамматика  $G = \langle N, \Sigma, P, S \rangle$  принадлежит классу  $LL(1)$ , если для любых двух правил  $(A \rightarrow \alpha) \in P_\$$  и  $(A \rightarrow \beta) \in P_\$,$  где  $\alpha \neq \beta$ , множества  $DIRECTOR_{G_\$}(A \rightarrow \alpha)$  и  $DIRECTOR_{G_\$}(A \rightarrow \beta)$  не пересекаются.

**Замечание 13.1.31.** Первая буква  $L$  в названии класса  $LL(1)$  означает, что входное слово читается слева направо (left-to-right). Вторая буква  $L$  означает, что строится левосторонний вывод (leftmost derivation). Число  $1$  указывает на то, что на каждом шаге для принятия решения используется один символ неразобранной части входного слова.

**Пример 13.1.32.** Грамматика из примера 13.1.3 принадлежит классу  $LL(1)$ , а грамматика из примера 3.1.17 этому классу не принадлежит.

**Теорема 13.1.33.** МП-автомат  $M$ , построенный в теореме 13.1.28 по контекстно-свободной грамматике  $G$ , является детерминированным тогда и только тогда, когда грамматика  $G$  принадлежит классу  $LL(1)$ .

**Теорема 13.1.34.** Грамматики из класса  $LL(1)$  порождают только детерминированные контекстно-свободные языки.

**Теорема 13.1.35.** Пусть контекстно-свободная грамматика  $G = \langle N, \Sigma, P, S \rangle$  не содержит бесполезных символов. Система множеств  $FIRST(\omega)$  (где  $\omega \in (N \cup \Sigma)^*$ ) состоит из наименьших (по включению) множеств, удовлетворяющих следующим условиям:

1. если  $\phi \xRightarrow{*} \epsilon$ , то  $b \in FIRST(\phi b \psi)$  для всех  $b \in \Sigma$  и  $\psi \in (N \cup \Sigma)^*$ ;
2. если  $\phi \xRightarrow{*} \epsilon$ , то  $FIRST(A) \subseteq FIRST(\phi A \psi)$  для всех  $A \in N$  и  $\psi \in (N \cup \Sigma)^*$ ;
3. если  $(A \rightarrow \alpha) \in P$ , то  $FIRST(\alpha) \subseteq FIRST(A)$ .

**Теорема 13.1.36.** Пусть контекстно-свободная грамматика  $G = \langle N, \Sigma, P, S \rangle$  не содержит бесполезных символов. Система множеств  $\text{FOLLOW}(A)$  (где  $A \in N$ ) состоит из наименьших (по включению) множеств, удовлетворяющих следующим условиям:

1. если  $B \in N$  и  $(A \rightarrow \phi B \psi) \in P$ , то  $\text{FIRST}(\psi) \subseteq \text{FOLLOW}(B)$ ;
2. если  $B \in N$ ,  $(A \rightarrow \phi B \psi) \in P$  и  $\psi \xrightarrow{*} \varepsilon$ , то  $\text{FOLLOW}(A) \subseteq \text{FOLLOW}(B)$ .

**Замечание 13.1.37.** Теоремы 13.1.35 и 13.1.36 дают алгоритм проверки принадлежности контекстно-свободной грамматики  $G$  классу  $\text{LL}(1)$ . Можно предположить, что грамматика  $G$  не содержит бесполезных символов (их можно устранить).

Сначала необходимо вычислить значения  $\text{FIRST}_{G_{\#}}(\omega')$  параллельно для всех слов  $\omega'$ , являющихся суффиксами правых частей правил грамматики  $G_{\#}$ . Для этого постепенно пополняем множества  $\text{FIRST}_{G_{\#}}(\omega')$  (начав с пустых множеств), используя условия, приведенные в теореме 13.1.35.

Затем необходимо вычислить значения  $\text{FOLLOW}_{G_{\#}}(A)$  параллельно для всех вспомогательных символов  $A$ . Для этого постепенно пополняем множества

$\text{FOLLOW}_{G_{\#}}(A)$  (начав с пустых множеств), используя условия, приведенные в теореме 13.1.35.

В заключение вычислим значения  $\text{DIRECTOR}_{G_{\#}}(A \rightarrow \alpha)$  для всех правил  $A \rightarrow \alpha$ , следуя определению функции **DIRECTOR**, и проверим условия из определения 13.1.30.

**Теорема 13.1.38.** Пусть даны контекстно-свободная грамматика  $G = \langle N, \Sigma, P, S \rangle$  и символ  $A \in N$ . Пусть

$$P = P_0 \cup \{A \rightarrow A\alpha_1, \dots, A \rightarrow A\alpha_n, A \rightarrow \beta_1, \dots, A \rightarrow \beta_m\},$$

где множество  $P_0$  не содержит правил с левой частью  $A$  и ни одно из слов  $\beta_1, \dots, \beta_m$  не начинается с символа  $A$ . Тогда грамматика  $G$  эквивалентна контекстно-свободной грамматике  $G' = \langle N \cup \{A'\}, \Sigma, P', S \rangle$ , где  $A'$  - новый символ, не принадлежащий множеству  $N \cup \Sigma$ , и

$$P' = P_0 \cup \{A \rightarrow \beta_1 A', \dots, A \rightarrow \beta_m A', A' \rightarrow A\alpha_1, \dots, A' \rightarrow A\alpha_n, A' \rightarrow \varepsilon\}.$$

**Пример 13.1.39.** Если к контекстно-свободной грамматике  $G_{\#}$  из примера 13.1.17 два раза применить преобразование из теоремы 13.1.38 (для вспомогательного символа  $A$  и для вспомогательного символа  $B$ ), то получим эквивалентную грамматику  $G'$  с правилами



$$\begin{array}{lll}
S \rightarrow A\$, & B \rightarrow CD, & C \rightarrow (A), \\
A \rightarrow BE, & D \rightarrow *CD, & C \rightarrow m, \\
E \rightarrow +BE, & D \rightarrow /CD, & C \rightarrow n. \\
E \rightarrow -BE, & D \rightarrow \varepsilon, & \\
E \rightarrow \varepsilon, & & 
\end{array}$$

После упрощения получаем грамматику  $G''$  с правилами

$$\begin{array}{lll}
S \rightarrow A\$, & D \rightarrow *CD, & C \rightarrow (CDE), \\
A \rightarrow CDE, & D \rightarrow /CD, & C \rightarrow m, \\
E \rightarrow +CDE, & D \rightarrow \varepsilon, & C \rightarrow n. \\
E \rightarrow -CDE, & & \\
E \rightarrow \varepsilon, & & 
\end{array}$$

Язык  $L(G'')$  распознается детерминированным МП-автоматом

$$\begin{aligned}
M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle, \text{ где } Q = \{q_s, \varepsilon\}, \Gamma = \{C, E, D, ), \$\}, I = \{q_s\}, \\
F = \{\varepsilon\} \text{ и} \\
\Delta = \{ \langle \langle q_s, \varepsilon, \varepsilon \rangle, \langle \varepsilon, CDE\$ \rangle \rangle, \\
\langle \langle \varepsilon, *, D \rangle, \langle \varepsilon, CD \rangle \rangle, \\
\langle \langle \varepsilon, /, D \rangle, \langle \varepsilon, CD \rangle \rangle, \\
\langle \langle \varepsilon, +, DE \rangle, \langle \varepsilon, CDE \rangle \rangle, \\
\langle \langle \varepsilon, -, DE \rangle, \langle \varepsilon, CDE \rangle \rangle, \\
\langle \langle \varepsilon, ), DE \rangle, \langle \varepsilon, \varepsilon \rangle \rangle, \\
\langle \langle \varepsilon, \$, DE\$ \rangle, \langle \varepsilon, \varepsilon \rangle \rangle, \\
\langle \langle \varepsilon, m, C \rangle, \langle \varepsilon, \varepsilon \rangle \rangle, \\
\langle \langle \varepsilon, n, C \rangle, \langle \varepsilon, \varepsilon \rangle \rangle, \\
\langle \langle \varepsilon, (, C \rangle, \langle \varepsilon, CDE \rangle \rangle \rangle \}.
\end{aligned}$$

Можно доказать, что МП-автомат  $M$  является нисходящим магазинным анализатором для грамматики

$$\begin{array}{lll}
A \rightarrow CDE, & D \rightarrow *CD, & C \rightarrow (CDE), \\
E \rightarrow +CDE, & D \rightarrow /CD, & C \rightarrow m, \\
E \rightarrow -CDE, & D \rightarrow \varepsilon, & C \rightarrow n. \\
E \rightarrow \varepsilon, & & 
\end{array}$$

**Пример 13.1.40.** Пусть  $\Sigma = \{m, n, -, *\}$  и  $N = \{A, B, C\}$ . Рассмотрим контекстно-свободную грамматику  $G$  с правилами

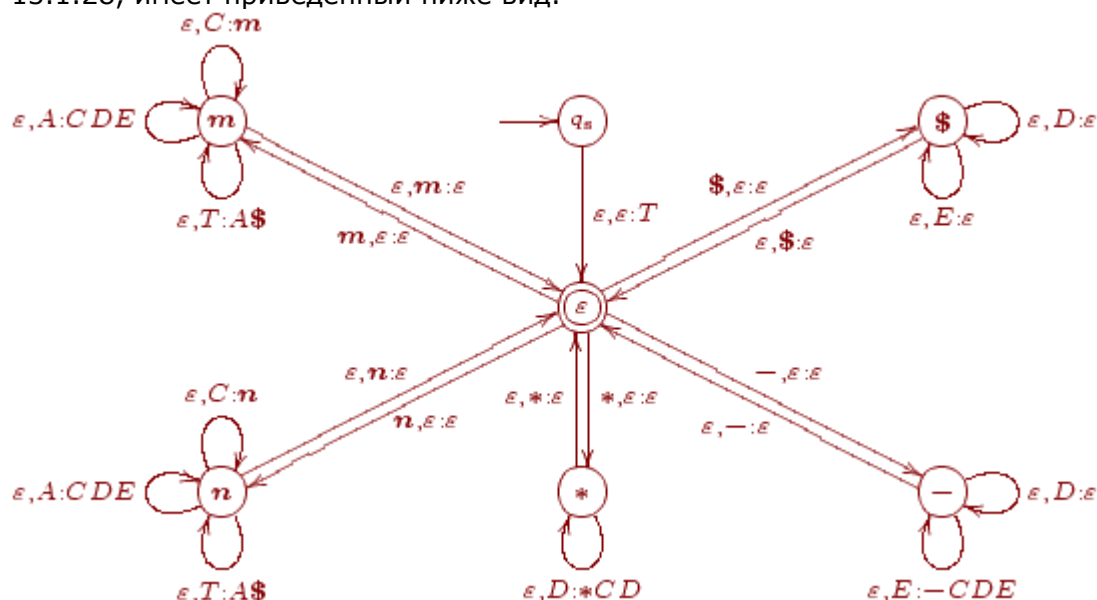
$$\begin{array}{ll}
A \rightarrow CDE, & D \rightarrow *CD, \\
E \rightarrow -CDE, & D \rightarrow \varepsilon, \\
E \rightarrow \varepsilon, & C \rightarrow m, \\
& C \rightarrow n.
\end{array}$$

Грамматика с маркером конца строки имеет правила



$$\begin{aligned}
 T &\rightarrow A\$, \\
 A &\rightarrow CDE, \\
 E &\rightarrow -CDE, \\
 E &\rightarrow \varepsilon, \\
 D &\rightarrow *CD, \\
 D &\rightarrow \varepsilon, \\
 C &\rightarrow m, \\
 C &\rightarrow n.
 \end{aligned}$$

Соответствующий нисходящий магазинный анализатор  $M$ , построенный в теореме 13.1.28, имеет приведенный ниже вид.



Легко убедиться, что грамматика  $G$  принадлежит классу  $LL(1)$  и МП-автомат  $M$  является детерминированным.

**Упражнение 13.1.41.** Существует ли детерминированный нисходящий магазинный анализатор для грамматики

$$\begin{aligned}
 S &\rightarrow bScS, \\
 S &\rightarrow bS, \\
 S &\rightarrow a?
 \end{aligned}$$

**Упражнение 13.1.42.** Принадлежит ли грамматика

$$\begin{aligned}
 S &\rightarrow a, \\
 S &\rightarrow bS, \\
 S &\rightarrow cSS
 \end{aligned}$$

классу  $LL(1)$ ?

**Упражнение 13.1.43.** Принадлежит ли грамматика

$$\begin{aligned}
 S &\rightarrow a, \\
 S &\rightarrow Sb, \\
 S &\rightarrow SSa
 \end{aligned}$$

классу  $LL(1)$ ?

## 13.2. Восходящий разбор

**Определение 13.2.1.** Протоколом правостороннего вывода в контекстно-свободной грамматике  $G = \langle N, \Sigma, P, S \rangle$  будем называть обращенную последовательность правил, примененных в этом выводе. Формально говоря, протоколом правостороннего вывода

$\omega_0, \omega_1, \dots, \omega_n$

является последовательность

$(A_1 \rightarrow \alpha_1), \dots, (A_n \rightarrow \alpha_n),$

где для каждого  $i < n$ , если  $\omega_i = \eta B v$  и  $\omega_{i+1} = \eta \beta v$  для некоторых  $\eta \in (N \cup \Sigma)^*$ ,  $B \in N$ ,  $v \in \Sigma^*$ ,  $\beta \in (N \cup \Sigma)^*$ , то  $A_{n-i} = B$  и  $\alpha_{n-i} = \beta$ .

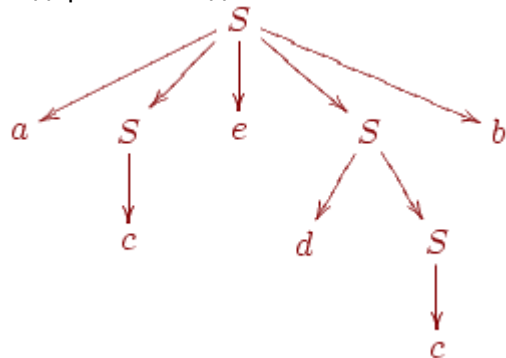
**Пример 13.2.2.** Рассмотрим контекстно-свободную грамматику

$S \rightarrow c$

$S \rightarrow dS$

$S \rightarrow aSeSb$

и дерево вывода



из примера 13.1.3. Этому дереву вывода соответствует правосторонний вывод

$S \Rightarrow aSeSb \Rightarrow aSedSb \Rightarrow aSedcb \Rightarrow acedcb.$

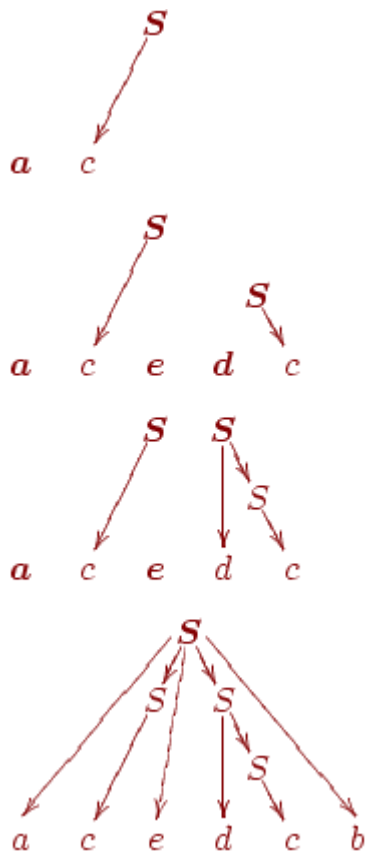
Протоколом этого вывода является последовательность

$(S \rightarrow c), (S \rightarrow c), (S \rightarrow dS), (S \rightarrow aSeSb).$

**Лемма 13.2.3.** Разным правосторонним выводам в одной и той же контекстно-свободной грамматике соответствуют разные протоколы.

Протокол правостороннего вывода в контекстно-свободной грамматике является естественным описанием соответствующего дерева вывода в порядке постфиксного обхода (postorder traversal). (Постфиксный обход упорядоченного дерева можно определить рекурсивно так: сначала выполняется постфиксный обход первого непосредственного потомка корня, затем второго и т. д., а в самом конце посещается корень дерева.)

Например, протокол правостороннего вывода из примера 13.2.2 задает процесс постепенного конструирования дерева вывода, изображенный ниже.



**Определение 13.2.5.** Правым разбором (right parse) слова  $w$  в контекстно-свободной грамматике  $G$  называется протокол любого правостороннего вывода слова  $w$  в грамматике  $G$ .

**Пример 13.2.6.** Правым разбором слова  $a c e a a c e c b e c b b$  в грамматике из примера 13.2.2 является последовательность

$(S \rightarrow c), (S \rightarrow c), (S \rightarrow c), (S \rightarrow a S e S b),$   
 $(S \rightarrow c), (S \rightarrow a S e S b), (S \rightarrow a S e S b).$

**Определение 13.2.7.** Процесс нахождения правого разбора слова  $w$  в заданной контекстно-свободной грамматике  $G$  называется *восходящим разбором* (bottom-up parsing).

**Определение 13.2.8.** Пусть даны контекстно-свободная грамматика

$G = \langle N, \Sigma, P, S \rangle$  и МП-автомат  $M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ . Будем говорить, что МП-автомат  $M$  - *восходящий магазинный анализатор* (bottom-up, left-to-right parser) для грамматики  $G$ , если  $L(M) = L(G_{\$})$  и существует такой гомоморфизм  $r: \Delta^* \rightarrow P_{\$}^*$ , что для каждого вычислительного процесса (МП-автомата  $M$ ), допускающего слово  $w \in \Sigma_{\$}^*$ , образ протокола этого вычислительного процесса при гомоморфизме  $r$  является протоколом некоторого правостороннего вывода слова  $w$  в грамматике  $G_{\$}$ .

**Пример 13.2.9.** Рассмотрим контекстно-свободную грамматику  $G_{\$}$  из примера 13.1.17.

Язык  $L(G_{\$})$  распознается недетерминированным МП-автоматом

$M = \langle \{1, 2\}, \Sigma, \Gamma, \Delta, \{1\}, \{2\} \rangle$ , где  
 $\Gamma = \{A, B, C, +, -, *, /\}$   
и

$$\Delta = \{ \langle \langle 1, +, \epsilon \rangle, \langle 1, + \rangle \rangle, \langle \langle 1, -, \epsilon \rangle, \langle 1, - \rangle \rangle, \\ \langle \langle 1, *, \epsilon \rangle, \langle 1, * \rangle \rangle, \langle \langle 1, /, \epsilon \rangle, \langle 1, / \rangle \rangle, \\ \langle \langle 1, (, \epsilon \rangle, \langle 1, ( \rangle \rangle, \\ \langle \langle 1, \epsilon, B+A \rangle, \langle 1, A \rangle \rangle, \langle \langle 1, \epsilon, B-A \rangle, \langle 1, A \rangle \rangle, \langle \langle 1, \epsilon, B \rangle, \langle 1, A \rangle \rangle, \\ \langle \langle 1, \epsilon, C*B \rangle, \langle 1, B \rangle \rangle, \langle \langle 1, \epsilon, C/B \rangle, \langle 1, B \rangle \rangle, \langle \langle 1, \epsilon, C \rangle, \langle 1, B \rangle \rangle, \\ \langle \langle 1, ), A \rangle, \langle 1, C \rangle \rangle, \langle \langle 1, m, \epsilon \rangle, \langle 1, C \rangle \rangle, \langle \langle 1, n, \epsilon \rangle, \langle 1, C \rangle \rangle, \\ \langle \langle 1, \$, A \rangle, \langle 2, \epsilon \rangle \rangle \}.$$

Можно проверить, что МП-автомат  $M$  является восходящим магазинным анализатором для грамматики  $G$  из примера 13.1.17.

**Определение 13.2.10.** Контекстно-свободная грамматика  $G = \langle N, \Sigma, P, S \rangle$  принадлежит классу  $LR(1)$ , если для любых  $\eta_1 \in (N\$ \cup \Sigma\$)^*$ ,  $\eta_2 \in (N\$ \cup \Sigma\$)^*$ ,  $A_1 \in N\$ \cup \Sigma\$$ ,  $A_2 \in N\$ \cup \Sigma\$$ ,  $b \in \Sigma$ ,  $v_1 \in \Sigma^*$ ,  $v_2 \in \Sigma^*$ ,  $\alpha_1 \in (N\$ \cup \Sigma\$)^*$ ,  $\alpha_2 \in (N\$ \cup \Sigma\$)^*$  из условий

$$S\$ \xRightarrow{rm} \eta_1 A_1 b v_1,$$

$$S\$ \xRightarrow{rm} \eta_2 A_2 b v_2,$$

$$(A_1 \rightarrow \alpha_1) \in P,$$

$$(A_2 \rightarrow \alpha_2) \in P,$$

$$\eta_1 \alpha_1 = \eta_2 \alpha_2$$

следует, что  $A_1 = A_2$  и  $\alpha_1 = \alpha_2$ .

**Замечание 13.2.11.** Классу  $LR(1)$  принадлежат те языки, для которых возможен восходящий разбор слова при чтении слева направо с заглядыванием на один символ вперед.

**Замечание 13.2.12.** Буква L в названии класса  $LR(1)$  означает, что входное слово читается слева направо (left-to-right). Буква R означает, что строится правосторонний вывод (rightmost derivation). Число 1 указывает на то, что на каждом шаге для принятия решения используется один символ неразобранной части входного слова.

**Теорема 13.2.13.** Для каждой грамматики из класса  $LR(1)$  существует детерминированный восходящий магазинный анализатор.

Доказательство можно найти в [2, с. 420-447].

**Теорема 13.2.14.** Грамматики из класса  $LR(1)$  порождают только детерминированные контекстно-свободные языки.

**Теорема 13.2.15.** Каждая грамматика из класса  $LL(1)$  принадлежит также классу  $LR(1)$ .

**Упражнение 13.2.16.** Найти все правые разборы слова  $bbacba$  в грамматике

$$S \rightarrow bScS,$$

$$S \rightarrow bS,$$

$$S \rightarrow a.$$

**Упражнение 13.2.17.** Найти детерминированный восходящий магазинный анализатор для грамматики

$$S \rightarrow a,$$

$$S \rightarrow Sb,$$

$$S \rightarrow SSc.$$

**Упражнение 13.2.18.** Найти детерминированный восходящий магазинный анализатор для грамматики

$$S \rightarrow a,$$

$$S \rightarrow bS,$$

$$S \rightarrow cSS.$$

## Математическая теория формальных языков

### 14. Лекция: Алгоритмические проблемы: версия для печати и PDA

В данной лекции рассматриваются понятия вычислимости, разрешимости, перечислимости и универсальных моделей вычислений. Рассмотрены алгоритмические проблемы, связанные с контекстно-свободными грамматиками, вводятся термины "массовая задача", "индивидуальная задача", "схема кодирования", "задача распознавания", "алгоритмическая проблема". Приведены практические примеры и задачи для самостоятельной проработки

Основная цель этой лекции - дать определения понятий, необходимых для математически строгой формулировки результатов следующих двух лекций. Для более подробного ознакомления с вычислимостью, разрешимостью, перечислимостью и универсальными моделями вычислений следует обратиться к какому-либо вводному курсу по теории алгоритмов, например [30, с. 93-106, 109-121] или [5, с. 8-20, 112-123]. В разделе 14.1 фиксируется конкретная модель вычислений - машина Тьюринга и даются определения (недетерминированной) машины Тьюринга, детерминированной машины Тьюринга и вычислимой функции. В разделе 14.2 определяются разрешимые и перечислимые множества. В разделе 14.3 вводятся термины "массовая задача", "индивидуальная задача", "схема кодирования", "задача распознавания", "алгоритмическая проблема". В 14.4\* формулируется теорема о соответствии машин Тьюринга грамматикам типа 0, а также теорема о нормальной форме для грамматик типа 0. В разделе 14.5 формулируется известная неразрешимая проблема - проблема соответствий Поста. На основании именно к этой алгоритмической проблеме основываются все доказательства неразрешимости в лекции 16, где рассматриваются алгоритмические проблемы, связанные с контекстно-свободными грамматиками.

#### 14.1. Машины Тьюринга

**Определение 14.1.1.** Машиной Тьюринга называется семка

$$M = \langle Q, \Sigma, \Gamma, b_0, \Delta, I, F \rangle,$$

где  $Q$ ,  $\Gamma$  и  $\Delta$  - конечные множества,  $\Sigma \subseteq \Gamma$ ,  $b_0 \in \Gamma - \Sigma$ ,  $I \subseteq Q$ ,  $F \subseteq Q$  и  $\Delta \subseteq ((Q - F) \times \Gamma) \times (Q \times \Gamma \times \{-1, 0, 1\})$ .

Здесь  $Q$  - множество состояний,  $\Sigma$  - входной алфавит (внешний алфавит),  $\Gamma$  - ленточный алфавит (tape alphabet),  $b_0$  - бланк (пробел, пустой символ, blank symbol),  $\Delta$  - множество переходов,  $I$  - множество начальных состояний,  $F$  - множество заключительных состояний.

**Определение 14.1.2.** Конфигурацией машины Тьюринга называется любая четверка

$$\langle x, q, a, y \rangle, \text{ где } x \in \Gamma^*, q \in Q, a \in \Gamma, y \in \Gamma^*.$$

**Определение 14.1.3.** Определим на множестве всех конфигураций машины Тьюринга

м бинарное отношение  $\vdash_M$  (такт работы) следующим образом.

Если  $\langle \langle p, a \rangle, \langle q, c, 0 \rangle \rangle \in \Delta$ , то  
 $\langle x, p, a, y \rangle \vdash_M \langle x, q, c, y \rangle$

для всех  $x \in \Gamma^*$  и  $y \in \Gamma^*$ .

Если  $\langle \langle p, a \rangle, \langle q, c, 1 \rangle \rangle \in \Delta$ , то  
 $\langle x, p, a, dy \rangle \vdash_M \langle xc, q, d, y \rangle$  и  $\langle x, p, a, \varepsilon \rangle \vdash_M \langle xc, q, b_0, \varepsilon \rangle$

для всех  $x \in \Gamma^*$ ,  $y \in \Gamma^*$  и  $d \in \Gamma$ .

Если  $\langle \langle p, a \rangle, \langle q, c, -1 \rangle \rangle \in \Delta$ , то  
 $\langle x d, p, a, y \rangle \vdash_M \langle x, q, d, c y \rangle$  и  $\langle \varepsilon, p, a, y \rangle \vdash_M \langle \varepsilon, q, b_0, c y \rangle$

для всех  $x \in \Gamma^*$ ,  $y \in \Gamma^*$  и  $d \in \Gamma$ .

**Замечание 14.1.4.** Если из контекста ясно, о какой машине Тьюринга идет речь, вместо  $\vdash_M$  будем писать просто  $\vdash$ .

**Определение 14.1.5.** Как и для МП-автомата, для машины Тьюринга бинарное отношение  $\vdash^*$  определяется как рефлексивное, транзитивное замыкание отношения  $\vdash$ .

**Замечание 14.1.6.** Если  $\langle x_1, q_1, a_1, y_1 \rangle \vdash^* \langle x_2, q_2, a_2, y_2 \rangle$ , то для любых  $k \in \mathbb{N}$  и  $l \in \mathbb{N}$  найдутся такие  $m \in \mathbb{N}$  и  $n \in \mathbb{N}$ , что  
 $\langle b_0^k x_1, q_1, a_1, y_1 b_0^l \rangle \vdash^* \langle b_0^m x_2, q_2, a_2, y_2 b_0^n \rangle$ .

**Замечание 14.1.7.** Конфигурацию  $\langle b_0^m x, q, a, y b_0^n \rangle$  иногда изображают сокращенно  $\underset{q}{x a y}$ .

**Замечание 14.1.8.** Машины Тьюринга можно изображать в виде *диаграмм*. При этом каждое состояние обозначается кружком, а переход - стрелкой. Каждое начальное состояние распознается по ведущей в него короткой стрелке. Каждое заключительное состояние отмечается на диаграмме двойным кружком. Стрелка с пометкой  $a:ck$ , ведущая из  $p$  в  $q$ , показывает, что  $\langle \langle p, a \rangle, \langle q, c, k \rangle \rangle$  является переходом данной машины Тьюринга. Обычно на диаграммах вместо чисел  $-1, 0, 1$  (обозначающих движение влево, стояние на месте, движение вправо) используются буквы  $L, N, R$  соответственно.

**Пример 14.1.9.** Рассмотрим машину Тьюринга

$$M = \langle Q, \Sigma, \Gamma, b_0, \Delta, I, F \rangle,$$

где  $Q = \{0, 1, 2, 3\}$ ,  $\Sigma = \{a\}$ ,  $\Gamma = \{a, b\}$ ,  $b_0 = b$ ,  $I = \{0\}$ ,  $F = \{3\}$ ,  
 $\Delta = \{ \langle \langle 0, b \rangle, \langle 1, b, 1 \rangle \rangle, \langle \langle 1, a \rangle, \langle 2, b, 1 \rangle \rangle, \langle \langle 2, a \rangle, \langle 1, b, 1 \rangle \rangle, \langle \langle 1, b \rangle, \langle 3, b, 0 \rangle \rangle \}$ .



Можно проверить, что для любого  $k \in \mathbb{N}$  выполняется следующее:

$$b_0^{k+1} \vdash_1 a a^k,$$

$$a a^{k+2} \vdash_1^* a a^k,$$

$$a a^{2k+1} \vdash_3^* b.$$

**Определение 14.1.10.** Машина Тьюринга

$$\langle Q, \Sigma, \Gamma, b_0, \Delta, I, F \rangle$$

называется *детерминированной*, если множество  $I$  содержит ровно один элемент и для

каждой пары  $\langle p, a \rangle \in (Q - F) \times \Gamma$  существует не более одной тройки  $\langle q, c, k \rangle \in Q \times \Gamma \times \{-1, 0, 1\}$  со свойством  $\langle \langle p, a \rangle, \langle q, c, k \rangle \rangle \in \Delta$ .

**Пример 14.1.11** Машина Тьюринга из примера 14.1.9 является детерминированной.

**Определение 14.1.12.** Пусть  $f$  - частичная функция из  $\Sigma^*$  в  $\Sigma^*$ . Детерминированная машина Тьюринга

$$M = \langle Q, \Sigma, \Gamma, b_0, \Delta, I, F \rangle$$

вычисляет (computes) функцию  $f$  тогда и только тогда, когда для каждого слова

$w \in \Sigma^*$  1) если  $f(w)$  не определено, то не существует таких  $p \in I, q \in F, a \in \Gamma, x \in \Gamma^*, y \in \Gamma^*$ , что  $\langle \varepsilon, p, b_0, w \rangle \vdash^* \langle x, q, a, y \rangle$ ;

2) если  $f(w) = z$ , то для некоторых  $p \in I, q \in F, m \in \mathbb{N}$  и  $n \in \mathbb{N}$  выполнено условие

$$\langle \varepsilon, p, b_0, w \rangle \vdash^* \langle b_0^m, q, b_0, zb_0^n \rangle.$$

**Определение 14.1.13.** Частичная функция из  $\Sigma^*$  в  $\Sigma^*$  называется *вычислимой*, если существует детерминированная машина Тьюринга, вычисляющая эту функцию.

**Пример 14.1.14.** Машина Тьюринга из примера 14.1.9 вычисляет следующую частичную функцию:

$$f(a^n) = \begin{cases} \varepsilon, & \text{если число } n \text{ четно,} \\ \text{не определено,} & \text{если число } n \text{ нечетно.} \end{cases}$$

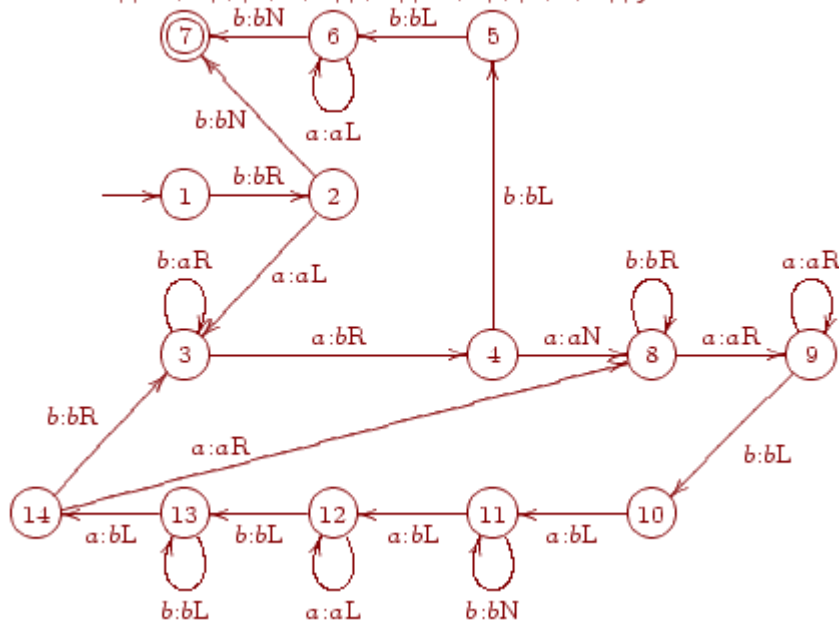
**Пример 14.1.15.** Рассмотрим детерминированную машину Тьюринга

$$\langle Q, \Sigma, \Gamma, b_0, \Delta, I, F \rangle,$$

где



$Q = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\},$   
 $\Sigma = \{a\},$   
 $\Gamma = \{a, b\},$   
 $b_0 = b,$   
 $I = \{1\},$   
 $F = \{7\},$   
 $\Delta = \{\langle\langle 1, b \rangle, \langle 2, b, 1 \rangle\rangle,$   
 $\langle\langle 2, b \rangle, \langle 7, b, 0 \rangle\rangle, \langle\langle 2, a \rangle, \langle 3, a, -1 \rangle\rangle,$   
 $\langle\langle 3, b \rangle, \langle 3, a, 1 \rangle\rangle, \langle\langle 3, a \rangle, \langle 4, b, 1 \rangle\rangle,$   
 $\langle\langle 4, b \rangle, \langle 5, b, -1 \rangle\rangle, \langle\langle 4, a \rangle, \langle 8, a, 0 \rangle\rangle,$   
 $\langle\langle 5, b \rangle, \langle 6, b, -1 \rangle\rangle,$   
 $\langle\langle 6, a \rangle, \langle 6, a, -1 \rangle\rangle, \langle\langle 6, b \rangle, \langle 7, b, 0 \rangle\rangle,$   
 $\langle\langle 8, b \rangle, \langle 8, b, 1 \rangle\rangle, \langle\langle 8, a \rangle, \langle 9, a, 1 \rangle\rangle,$   
 $\langle\langle 9, a \rangle, \langle 9, a, 1 \rangle\rangle, \langle\langle 9, b \rangle, \langle 10, b, -1 \rangle\rangle,$   
 $\langle\langle 10, a \rangle, \langle 11, b, -1 \rangle\rangle,$   
 $\langle\langle 11, b \rangle, \langle 11, b, 0 \rangle\rangle, \langle\langle 11, a \rangle, \langle 12, b, -1 \rangle\rangle,$   
 $\langle\langle 12, a \rangle, \langle 12, a, -1 \rangle\rangle, \langle\langle 12, b \rangle, \langle 13, b, -1 \rangle\rangle,$   
 $\langle\langle 13, b \rangle, \langle 13, b, -1 \rangle\rangle, \langle\langle 13, a \rangle, \langle 14, b, -1 \rangle\rangle,$   
 $\langle\langle 14, a \rangle, \langle 8, a, 1 \rangle\rangle, \langle\langle 14, b \rangle, \langle 3, b, 1 \rangle\rangle\}.$



Можно проверить, что для любых  $i \in \mathbb{N}$ ,  $j \in \mathbb{N}$  и  $k \in \mathbb{N}$  выполняется следующее:

$$ba^{k+2} \vdash_1^* abaa^k,$$

$$ab^{j+1}aa^2 \vdash_8^* ba^{j+2},$$

$$ab^{j+1}aa^{k+3} \vdash_8^* a^{j+2}baa^k,$$

$$a^{i+2}b^{j+1}aa^{k+2} \vdash_8^* a^{i+1}b^{j+2}aa^k,$$

$$ab^{j+1}aa^{k+2j+5} \vdash_8^* ab^{j+2}aa^k.$$

Следовательно,

$$ba^{i+(j+2)^2} \vdash_1^* ab^{j+1}aa^{i+2},$$

$$ba^{(k+2)^2} \vdash_7^* ba^{k+2}.$$

Рассматриваемая машина Тьюринга вычисляет следующую частичную функцию:

$$f(a^n) = \begin{cases} a^{\sqrt{n}}, & \text{если } \sqrt{n} \in \mathbb{N}, \\ \text{не определено,} & \text{если } \sqrt{n} \notin \mathbb{N}. \end{cases}$$

**Замечание 14.1.16.** Тезисом Черча-Тьюринга (а иногда просто тезисом Черча) называется следующее неформальное утверждение: для каждой вычислимой в интуитивном смысле частичной функции из  $\Sigma^*$  в  $\Sigma^*$  существует машина Тьюринга, которая эту функцию вычисляет.

**Упражнение 14.1.17.** Найти детерминированную машину Тьюринга с входным алфавитом  $\{a\}$ , вычисляющую функцию  $f$ , заданную соотношением

$$f(a^n) = a^{n+1}.$$

**Упражнение 14.1.18.** Найти детерминированную машину Тьюринга с входным алфавитом  $\{a\}$ , вычисляющую функцию  $f$ , заданную соотношением

$$f(a^n) = \varepsilon.$$

**Определение 14.1.19.** Зафиксируем два различных символа  $a$  и  $b$  и положим

$\Sigma = \{a, b\}$ . Пусть  $k$  - положительное целое число и  $f$  - частичная функция из  $\mathbb{N}^k$  в  $\mathbb{N}$ . Детерминированная машина Тьюринга

$$M = \langle Q, \Sigma, \Gamma, b, \Delta, I, F \rangle$$

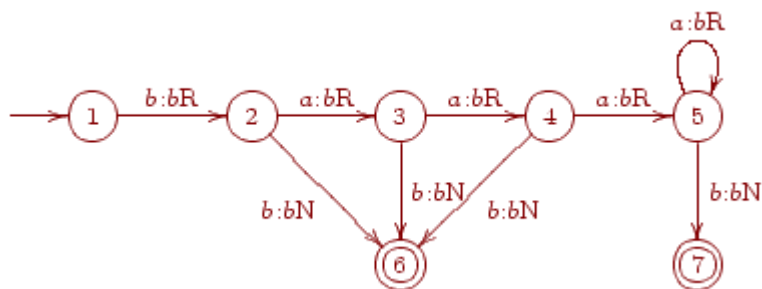
вычисляет функцию  $f$  тогда и только тогда, когда для любых натуральных чисел  $n_1, \dots, n_k$  выполняются следующие условия:

1) если  $f(n_1, \dots, n_k)$  не определено, то не существует таких  $p \in I$ ,  $q \in F$ ,  $a \in \Gamma$ ,  $x \in \Gamma^*$ ,  $y \in \Gamma^*$ , что  $\langle \varepsilon, p, b, a^{n_1}ba^{n_2}b \dots ba^{n_k} \rangle \vdash^* \langle x, q, a, y \rangle$ ;

2) если  $f(n_1, \dots, n_k) = l$ , то для некоторых  $p \in I$ ,  $q \in F$ ,  $m \in \mathbb{N}$  и  $n \in \mathbb{N}$  выполнено условие

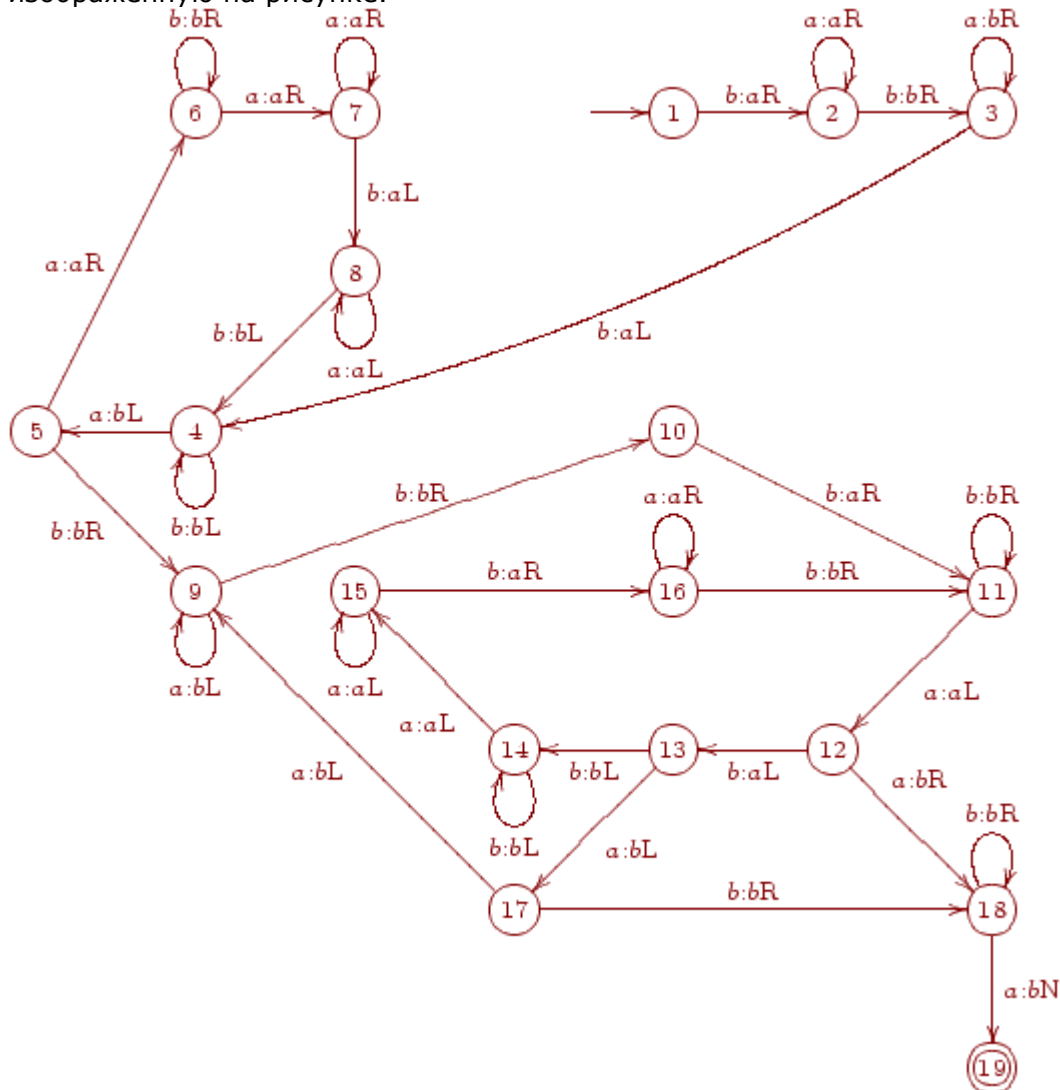
$$\langle \varepsilon, p, b, a^{n_1}ba^{n_2}b \dots ba^{n_k} \rangle \vdash^* \langle b^m, q, b_0, a^l b^n \rangle.$$

**Упражнение 14.1.20.** Рассмотрим детерминированную машину Тьюринга, изображенную на рисунке.



Какую функцию из  $\mathbb{N}$  в  $\mathbb{N}$  она вычисляет?

**Упражнение 14.1.21.** Рассмотрим детерминированную машину Тьюринга, изображенную на рисунке.



Какую функцию из  $\mathbb{N} \times \mathbb{N}$  в  $\mathbb{N}$  она вычисляет?

## 14.2. Разрешимые и перечислимые множества

**Определение 14.2.1.** Говорят, что детерминированная машина Тьюринга  $\langle Q, \Sigma, \Gamma, b_0, \Delta, \{q_s\}, \{q_a, q_r\} \rangle$

с выделенным состоянием  $q_a$  разрешает (decides) язык  $L \subseteq \Sigma^*$ , если

- 1) для каждого слова  $w \in L$  найдутся такие  $m \in \mathbb{N}$  и  $n \in \mathbb{N}$ , что  $\langle \varepsilon, q_s, b_0, w \rangle \vdash^* \langle b_0^m, q_a, b_0, b_0^n \rangle$ ;

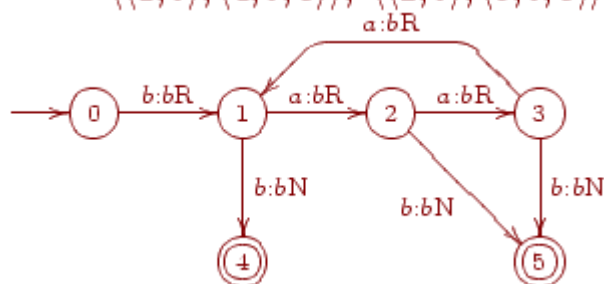
2) для каждого слова  $w \in \Sigma^* - L$  найдутся такие  $m \in \mathbb{N}$  и  $n \in \mathbb{N}$ , что

$\langle \varepsilon, q_s, b_0, w \rangle \vdash^* \langle b_0^m, q_r, b_0, b_0^n \rangle$ . Состояние  $q_a$  называется *допускающим* (*принимаящим*, accept state), состояние  $q_r$  называется *отвергающим* (*непринимаящим*, reject state).

**Пример 14.2.2.** Рассмотрим детерминированную машину Тьюринга

$M = \langle Q, \Sigma, \Gamma, b_0, \Delta, \{q_s\}, \{q_a, q_r\} \rangle$ ,

где  $Q = \{0, 1, 2, 3, 4, 5\}$ ,  $\Sigma = \{a\}$ ,  $\Gamma = \{a, b\}$ ,  $b_0 = b$ ,  $q_s = 0$ ,  $q_a = 4$ ,  $q_r = 5$  и  $\Delta = \{ \langle 0, b \rangle, \langle 1, b, 1 \rangle, \langle 1, a \rangle, \langle 2, b, 1 \rangle, \langle 2, a \rangle, \langle 3, b, 1 \rangle, \langle 3, a \rangle, \langle 1, b, 1 \rangle, \langle 1, b \rangle, \langle 4, b, 0 \rangle, \langle 2, b \rangle, \langle 5, b, 0 \rangle, \langle 3, b \rangle, \langle 5, b, 0 \rangle \}$ .



Эта машина Тьюринга разрешает язык  $\{a^{3n} \mid n \in \mathbb{N}\}$ .

**Определение 14.2.3.** Язык  $L$  над алфавитом  $\Sigma$  называется *разрешимым* или *рекурсивным* (decidable, recursive), если существует детерминированная машина Тьюринга

$\langle Q, \Sigma, \Gamma, b_0, \Delta, \{q_s\}, \{q_a, q_r\} \rangle$

(с выделенным состоянием  $q_a$ ), которая разрешает язык  $L$ .

**Определение 14.2.4.** Говорят, что машина Тьюринга

$M = \langle Q, \Sigma, \Gamma, b_0, \Delta, \{q_s\}, \{q_a\} \rangle$

допускает (принимает, accepts) слово  $w \in \Sigma^*$ , если для некоторых  $m \in \mathbb{N}$  и  $n \in \mathbb{N}$

$\langle \varepsilon, q_s, b_0, w \rangle \vdash^* \langle b_0^m, q_a, b_0, b_0^n \rangle$ .

**Определение 14.2.5.** Язык, *допускаемый* машиной Тьюринга  $M$ , - это язык, состоящий из всех допускаемых данной машиной Тьюринга слов.

**Определение 14.2.6.** Язык называется *перечислимым* (*рекурсивно перечислимым*, *полуразрешимым*, recursively enumerable), если существует детерминированная машина Тьюринга, допускающая этот язык.

**Замечание 14.2.7.** В определении 14.2.6 можно отбросить требование детерминированности машины Тьюринга.

**Теорема 14.2.8.** Каждый разрешимый язык является перечислимым.

*Доказательство.* Пусть дана машина Тьюринга

$M = \langle Q, \Sigma, \Gamma, b_0, \Delta, \{q_s\}, \{q_a, q_r\} \rangle$

с выделенным состоянием  $q_a$ , которая разрешает язык  $L \subseteq \Sigma^*$ . Тогда машина Тьюринга

$M' = \langle Q, \Sigma, \Gamma, b_0, \Delta, \{q_s\}, \{q_a\} \rangle$

допускает язык  $L$ .

**Пример 14.2.9.** Если в машине Тьюринга из примера 14.1.15 заменить переход

$\langle\langle 6, a \rangle, \langle 6, a, -1 \rangle\rangle$  на  $\langle\langle 6, a \rangle, \langle 6, b, -1 \rangle\rangle$ , то получится машина Тьюринга,

допускающая язык  $\{a^{n^2} \mid n \in \mathbb{N}\}$ . Следовательно, этот язык является перечислимым. Можно доказать, что он даже является разрешимым.

**Теорема 14.2.10.** Существует такая машина Тьюринга над однобуквенным алфавитом  $\Sigma = \{a\}$ , что язык, допускаемый этой машиной Тьюринга, неразрешим.

*Доказательство.* Доказательство существования перечислимого неразрешимого множества можно найти, например, в [31, 9.2].

**Упражнение 14.2.11.** Найти детерминированную машину Тьюринга с входным алфавитом  $\{a\}$ , разрешающую язык  $\{a^n \mid n \leq 2\}$ .

**Упражнение 14.2.12.** Найти детерминированную машину Тьюринга с входным алфавитом  $\{a\}$ , допускающую язык  $\{a^{3n+2} \mid n \geq 0\}$ .

### 14.3. Массовые задачи

**Определение 14.3.1.** Массовой задачей (problem) называется бесконечная серия "однотипных" индивидуальных задач (instance), каждая из которых имеет определенный ответ. С каждой массовой задачей связана некоторая фиксированная схема кодирования (encoding scheme), которая отображает индивидуальные задачи в их коды - слова в некотором фиксированном алфавите. При этом требуется, чтобы множество кодов всех индивидуальных задач было разрешимым.

**Определение 14.3.2.** Задачей распознавания (decision problem) называется массовая задача, в которой ответами индивидуальных задач могут быть только "да" и "нет" (то есть существует только два возможных ответа).

**Пример 14.3.3.** Зафиксируем некоторый алфавит  $\Sigma$ . Рассмотрим следующие однотипные индивидуальные задачи: даны два слова  $x \in \Sigma^*$  и  $y \in \Sigma^*$ , необходимо выяснить, является ли слово  $x$  подсловом слова  $y$ .

Пусть  $\#$  - новый символ, не принадлежащий алфавиту  $\Sigma$ . Кодом индивидуальной задачи про конкретные слова  $x$  и  $y$  будем считать слово  $x\#y$  в алфавите  $\Sigma \cup \{\#\}$ .

Эта массовая задача является задачей распознавания.

**Определение 14.3.4.** Алгоритмическая проблема - проблема, в которой требуется найти алгоритм для решения массовой задачи. Если такой алгоритм существует, то данная проблема называется алгоритмически разрешимой или просто разрешимой (decidable, solvable), в противном случае ее называют алгоритмически неразрешимой (undecidable, unsolvable).

**Замечание 14.3.5.** Алгоритмическая проблема, относящаяся к некоторой задаче распознавания, алгоритмически разрешима тогда и только тогда, когда разрешимо множество кодов индивидуальных задач с ответом "да".

**Пример 14.3.6.** Рассмотрим массовую задачу из примера 14.3.3. Соответствующая

алгоритмическая проблема разрешима, так как разрешим язык

$$\{x\#uv \mid x \in \Sigma^*, u \in \Sigma^*, v \in \Sigma^*\}.$$

**Пример 14.3.7.** Зафиксируем некоторый алфавит  $\Sigma$ . Рассмотрим следующие однотипные индивидуальные задачи: дана порождающая грамматика

$$G = \langle \{A_1, \dots, A_n\}, \Sigma, P, A_1 \rangle,$$

необходимо выяснить, является ли эта грамматика праволинейной.

Для полной строгости необходимо договориться, как кодировать грамматику  $G$  в виде слова. Например, можно использовать алфавит  $\Sigma \cup \{\heartsuit, \spadesuit, 0, 1\}$ , где  $\heartsuit, \spadesuit, 0, 1$  - дополнительные символы, не принадлежащие множеству  $\Sigma$ . Вспомогательный символ  $A_i$  заменим на слово, состоящее из символа  $\heartsuit$  и кода числа  $i$  в двоичной системе

счисления. В каждом правиле  $\alpha \rightarrow \beta$  добавим символ  $\spadesuit$  на месте  $\rightarrow$  и после слова  $\beta$ . Кодом грамматики  $G$  будем считать результат конкатенации кодов всех правил из множества  $P$ . Например, грамматика

$$A_1 \rightarrow a,$$

$$A_1 \rightarrow A_2 A_1,$$

$$A_2 \rightarrow a A_3,$$

$$A_3 \rightarrow \varepsilon,$$

$$A_3 \rightarrow b A_1$$

(над алфавитом  $\Sigma = \{a, b\}$ ) кодируется словом

$$\heartsuit 1 \spadesuit a \heartsuit \heartsuit 1 \spadesuit \heartsuit 10 \heartsuit 1 \spadesuit \heartsuit 10 \spadesuit a \heartsuit 11 \spadesuit \heartsuit 11 \spadesuit \spadesuit \heartsuit 11 \spadesuit b \heartsuit 1 \spadesuit.$$

Легко понять, что соответствующая алгоритмическая проблема (проблема проверки праволинейности) разрешима.

**Упражнение 14.3.8.** Разрешима ли алгоритмическая проблема распознавания четности длины слова над алфавитом  $\{a, b\}$ ?

## 14.4\*. Грамматики типа 0

**Теорема 14.4.1.** Класс языков, порождаемых грамматиками типа 0, совпадает с классом перечислимых языков.

Доказательство можно найти, например, в [СерГал с. 24-26].

**Упражнение 14.4.2.** Выводимо ли слово  $aab$  в грамматике

$$S \rightarrow Sb,$$

$$Ta \rightarrow aaT,$$

$$S \rightarrow ST,$$

$$Tb \rightarrow ab,$$

$$S \rightarrow SU,$$

$$Uaaa \rightarrow aaU,$$

$$S \rightarrow \varepsilon,$$

$$Uab \rightarrow b?$$

**Упражнение 14.4.3.** Выводимо ли слово  $aaaaaaaaab$  в грамматике

$S \rightarrow Sb,$	$Ta \rightarrow aaT,$
$S \rightarrow ST,$	$Tb \rightarrow ab,$
$S \rightarrow SU,$	$Uaaa \rightarrow aaU,$
$S \rightarrow \varepsilon,$	$Uab \rightarrow b?$

**Замечание 14.4.4.** Неизвестно, порождает ли грамматика

$R \rightarrow Ra,$	$Ta \rightarrow aaT,$
$R \rightarrow S,$	$Tb \rightarrow ab,$
$S \rightarrow Sb,$	$Uaaa \rightarrow aaU,$
$S \rightarrow ST,$	$Uab \rightarrow b$
$S \rightarrow SU,$	
$S \rightarrow \varepsilon,$	

все слова в алфавите  $\{a, b\}$ .

**Упражнение 14.4.5.** Пусть  $\Sigma = \{a, b, c, d, e\}$ . Рассмотрим грамматику

$R \rightarrow ccdcdcdccddcdccca,$	$ce \rightarrow eca,$
$ac \rightarrow ca,$	$eca \rightarrow ce,$
$ca \rightarrow ac,$	$de \rightarrow edb,$
$bc \rightarrow cb,$	$edb \rightarrow de,$
$cb \rightarrow bc,$	$cca \rightarrow ccae,$
$ad \rightarrow da,$	$ccae \rightarrow cca.$
$da \rightarrow ad,$	
$bd \rightarrow db,$	
$db \rightarrow bd,$	

Выводимо ли в ней слово  $cccdcdcdccddcdcccabba$ ?

**Упражнение 14.4.6.** Является ли разрешимым язык, порождаемый грамматикой

$S \rightarrow cS,$	$ac \rightarrow ca,$	$cdc \rightarrow cdcR,$
$S \rightarrow dS,$	$ca \rightarrow ac,$	$Rcca \rightarrow ccR,$
$S \rightarrow aa,$	$bc \rightarrow cb,$	$Rddb \rightarrow ddR,$
	$cb \rightarrow bc,$	$Rcdd \rightarrow cddT,$
	$ad \rightarrow da,$	$Tcc \rightarrow ccaT,$
	$da \rightarrow ad,$	$Tdd \rightarrow ddbT,$
	$bd \rightarrow db,$	$Tcdc \rightarrow cdc?$
	$db \rightarrow bd,$	

**Определение 14.4.7.** Порождающая грамматика в нормальной форме - это порождающая грамматика, в которой каждое правило имеет вид  $A \rightarrow a$ ,  $A \rightarrow BC$  или  $AB \rightarrow \varepsilon$ , где  $A \in N$ ,  $B \in N$ ,  $C \in N$ ,  $a \in \Sigma$ .

**Теорема 14.4.8.** Каждая порождающая грамматика эквивалентна некоторой порождающей грамматике в нормальной форме.

**Упражнение 14.4.9.** Существует ли такая порождающая грамматика  $G$ , что язык  $L(G) - \{\varepsilon\}$  не порождается ни одной грамматикой типа 0?

## 14.5. Проблема соответствий Поста

**Определение 14.5.1.** Постовской системой соответствия над алфавитом  $\Sigma$  называется упорядоченная пара конечных последовательностей  $((x_1, \dots, x_n), (y_1, \dots, y_n))$ , где  $x_i \in \Sigma^*$  и  $y_i \in \Sigma^*$  для всех  $i$ .

**Замечание 14.5.2.** Систему  $((x_1, \dots, x_n), (y_1, \dots, y_n))$  иногда изображают в виде  $\left[ \frac{x_1}{y_1} \right], \dots, \left[ \frac{x_n}{y_n} \right]$ .

**Определение 14.5.3.** Решением (match) постовской системы соответствия  $((x_1, \dots, x_n), (y_1, \dots, y_n))$  называется непустая последовательность индексов

$(i_1, \dots, i_k)$ , удовлетворяющая условию  $x_{i_1} \dots x_{i_k} = y_{i_1} \dots y_{i_k}$ ,

где  $1 \leq i_j \leq n$  для каждого  $j$ .

**Пример 14.5.4.** Пусть  $\Sigma = \{a, b, c\}$ . Рассмотрим постовскую систему соответствия  $\left[ \frac{aab}{a} \right], \left[ \frac{a}{aa} \right], \left[ \frac{caa}{bc} \right]$ .

Последовательность  $(2, 1, 3, 2, 2)$  является решением этой системы, так как  $a \cdot aab \cdot caa \cdot a \cdot a = aa \cdot a \cdot bc \cdot aa \cdot aa$ .

**Упражнение 14.5.5.** Пусть  $\Sigma = \{a, b, c, 0, 1, \diamond, \heartsuit, \spadesuit\}$ . Существует ли решение у постовской системы соответствия

$$\left[ \frac{a\spadesuit}{a\spadesuit aba\heartsuit 1 aca\spadesuit a} \right], \left[ \frac{ab}{ba} \right], \left[ \frac{ac}{ca} \right], \left[ \frac{a\spadesuit}{\spadesuit a} \right], \left[ \frac{a\spadesuit}{ba\spadesuit a} \right], \left[ \frac{a\spadesuit}{\spadesuit aba} \right],$$

$$\left[ \frac{a\heartsuit 1 ab}{\diamond a} \right], \left[ \frac{a\heartsuit 1 ac}{ca\heartsuit 10a} \right], \left[ \frac{aba\heartsuit 10ab}{\heartsuit 10abaca} \right], \left[ \frac{aca\heartsuit 10ab}{\heartsuit 10acaca} \right], \left[ \frac{a\heartsuit 10ac}{ca\heartsuit 11a} \right], \left[ \frac{a\heartsuit 11ac}{ca\heartsuit 1a} \right],$$

$$\left[ \frac{a\diamond ab}{\diamond a} \right], \left[ \frac{a\diamond ac}{\diamond a} \right], \left[ \frac{aba\diamond}{\diamond a} \right], \left[ \frac{aca\diamond}{\diamond a} \right], \left[ \frac{a\diamond a\spadesuit a\spadesuit}{\spadesuit} \right]?$$

**Определение 14.5.6.** Проблемой соответствий Поста (Post correspondence problem) называется проблема нахождения алгоритма, выясняющего для каждой постовской системы соответствия, существует ли решение этой системы.

**Теорема 14.5.7.** Пусть  $|\Sigma| \geq 2$ . Тогда не существует алгоритма, позволяющего по произвольной постовской системе соответствия над алфавитом  $\Sigma$  узнать, имеет ли она решение. (Другими словами, проблема соответствий Поста неразрешима.)

Доказательство можно найти в [ХопМот, 9.4].

**Упражнение 14.5.8.** Существует ли решение у постовской системы соответствия

$$\left[ \frac{ab}{abab} \right], \left[ \frac{b}{a} \right], \left[ \frac{aba}{b} \right], \left[ \frac{aa}{a} \right]?$$



**Упражнение 14.5.9.** Существует ли решение у постовской системы соответствия

$$\left[ \frac{b}{ca} \right], \left[ \frac{a}{ab} \right], \left[ \frac{ca}{a} \right], \left[ \frac{abc}{c} \right] ?$$

**Упражнение 14.5.10.** Существует ли решение у постовской системы соответствия

$$\left[ \frac{a}{ab} \right], \left[ \frac{ca}{cc} \right], \left[ \frac{bcc}{ccca} \right], \left[ \frac{b}{ab} \right] ?$$

**Упражнение 14.5.11.** Существует ли решение у постовской системы соответствия

$$\left[ \frac{ab}{aba} \right], \left[ \frac{b}{a} \right], \left[ \frac{aba}{b} \right], \left[ \frac{bbab}{abb} \right] ?$$

**Упражнение 14.5.12.** Существует ли решение у постовской системы соответствия

$$\left[ \frac{ab}{b} \right], \left[ \frac{b}{bba} \right], \left[ \frac{ba}{b} \right], \left[ \frac{b}{bab} \right], \left[ \frac{b}{abb} \right] ?$$

**Упражнение 14.5.13.** Существует ли решение у постовской системы соответствия

$$\left[ \frac{ab}{b} \right], \left[ \frac{b}{bba} \right], \left[ \frac{ba}{b} \right], \left[ \frac{aba}{aba} \right] ?$$

**Упражнение 14.5.14.** Существует ли решение у постовской системы соответствия

$$\left[ \frac{a}{aaa} \right], \left[ \frac{\varepsilon}{aaaaa} \right], \left[ \frac{aaaaaa}{aa} \right] ?$$

**Упражнение 14.5.15.** Существует ли постовская система соответствия, имеющая ровно одно решение?

## Математическая теория формальных языков

### 15. Лекция: Алгоритмически разрешимые проблемы: версия для печати и PDA

В данной лекции рассматриваются наиболее известные разрешимые проблемы, связанные с грамматиками, автоматами и регулярными выражениями. Сформулированы также теоремы о разрешимости для некоторых алгоритмических проблем. Приведены практические примеры и упражнения для самостоятельного решения

В этой лекции рассматриваются наиболее известные разрешимые проблемы, связанные с грамматиками, автоматами и регулярными выражениями. Поскольку все приведенные выше доказательства эквивалентности разных способов конечного задания формального языка были конструктивными, не важно, какой из эквивалентных способов задания языка (например, посредством контекстно-свободной грамматики или автомата с магазинной памятью) используется в точной формулировке массовой задачи.

Первые два раздела этой лекции посвящены контекстным языкам. В разделах 15.3-15.6 сформулированы теоремы о разрешимости для некоторых алгоритмических проблем. Доказательства этих теорем опираются на конструктивность рассуждений, проведенных в лекциях 3 и 8. В разделе 15.7\* приводится новый неожиданный результат, доказанный в 1997 году.

Проблема эквивалентности конечных автоматов (или, что то же самое, регулярных выражений) разрешима, но, к сожалению, она сложна, то есть нахождение ответа индивидуальной задачи требует (в некотором смысле) много времени. В разделе 15.9\* указывается, что даже для регулярных выражений без итерации (которые, очевидно, соответствуют конечным автоматам без циклов) проблема неэквивалентности  $NP$ -полна (необходимые определения из теории сложности вычислений приведены в разделе 15.8\*).

#### 15.1. Неукорачивающие грамматики

**Определение 15.1.1.** Порождающая грамматика  $\langle N, \Sigma, P, S \rangle$  называется *неукорачивающей* (noncontracting), если для каждого правила  $(\alpha \rightarrow \beta) \in P$  выполняется неравенство  $|\alpha| \leq |\beta|$ .

**Теорема 15.1.2.** Существует алгоритм, позволяющий по произвольной неукорачивающей грамматике  $G$  и по слову  $w$  узнать, верно ли, что  $w \in L(G)$ .

**Теорема 15.1.3.** Каждая контекстная грамматика является неукорачивающей. Каждая неукорачивающая грамматика эквивалентна некоторой контекстной грамматике.

**Пример 15.1.4.** Грамматика

$$S \rightarrow ASTA,$$

$$S \rightarrow AbA,$$

$$A \rightarrow a,$$

$$bT \rightarrow bb,$$

$$AT \rightarrow TA$$

эквивалентна контекстной грамматике из примера 1.5.4.

**Упражнение 15.1.5.** Найти неукорачивающую грамматику, порождающую язык

$$\{w \in \{a, b, c\}^+ \mid |w|_a = |w|_b = |w|_c\}.$$

**Упражнение 15.1.6.** Найти неукорачивающую грамматику, порождающую язык  $\{w \in \{a, b, c\}^* \mid |w|_a < |w|_b < |w|_c\}$ .

**Упражнение 15.1.7.** Найти неукорачивающую грамматику, порождающую язык  $\{c^n d c^n d c^n \mid n \geq 0\}$ .

**Упражнение 15.1.7.** Найти неукорачивающую грамматику, порождающую язык  $\{w c w \mid w \in \{a, b\}^*\}$ .

## 15.2\*. Линейно ограниченные автоматы

**Определение 15.2.1.** Машина Тьюринга

$\langle Q, \Sigma, \Gamma, b_0, \Delta, I, F \rangle$   
называется *линейно ограниченным автоматом* (linear bounded automaton), если не существует таких  $w \in \Sigma^*$ ,  $x \in \Gamma^*$ ,  $q \in Q$ ,  $a \in \Gamma$  и  $y \in \Gamma^*$ , что  $\langle \varepsilon, p, b_0, w b_0 \rangle \vdash^* \langle x, q, a, y \rangle$  и  $|x a y| > |b_0 w b_0|$ .

**Теорема 15.2.2.** Язык  $L$ , не содержащий пустого слова, порождается некоторой неукорачивающей грамматикой тогда и только тогда, когда существует линейно ограниченный автомат (в общем случае недетерминированный), допускающий язык  $L$ .

**Замечание 15.2.3.** Неизвестно, верен ли аналог теоремы 15.2.2 для детерминированных линейно ограниченных автоматов.

**Теорема 15.2.4.** Класс языков, порождаемых неукорачивающими грамматиками, то есть класс контекстных языков, замкнут относительно операций объединения, пересечения и дополнения.

Замкнутость относительно операции дополнения доказали в 1987 году (независимо друг от друга) Нил Иммерман (Neil Immerman) и Роберт Селепчени (Robert Szelepcs) (см. [Imm, Sze]). Замкнутость относительно операции объединения очевидна, а пересечение выражается через объединение и дополнение.

**Упражнение 15.2.5.** Является ли контекстным язык  $\{w \in \{a, b, c, d\}^* \mid |w|_a \neq |w|_b, |w|_c \neq |w|_d\}$ ?

**Упражнение 15.2.6.** Является ли контекстным язык  $\{a\}^* - \{a^{n^2} \mid n \geq 0\}$ ?

## 15.3. Проблема выводимости слова

**Теорема 15.3.1.** Существует алгоритм, позволяющий по произвольной контекстно-свободной грамматике  $G$  узнать, верно ли, что  $\varepsilon \in L(G)$ .

**Теорема 15.3.2.** Существует алгоритм, позволяющий по произвольной контекстно-свободной грамматике  $G$  и по слову  $w$  узнать, верно ли, что  $w \in L(G)$ .

**Упражнение 15.3.3.** Принадлежит ли слово aaaaaabbbbabb языку, порождаемому грамматикой

$$F \rightarrow aFbbb,$$

$$F \rightarrow aFba,$$

$$F \rightarrow aFa,$$

$$F \rightarrow aFabb,$$

$$F \rightarrow a?$$

**Упражнение 15.3.4.** Принадлежит ли слово *abababa* языку, порождаемому грамматикой

$$J \rightarrow aa,$$

$$J \rightarrow FbF,$$

$$F \rightarrow aJ,$$

$$F \rightarrow Jb,$$

$$F \rightarrow bFa,$$

$$F \rightarrow \varepsilon?$$

## 15.4. Проблема пустоты языка

**Теорема 15.4.1.** Существует алгоритм, позволяющий по произвольной контекстно-свободной грамматике  $G$  узнать, верно ли, что  $L(G) = \emptyset$ .

*Доказательство.* Удалим из  $G$  все бесполезные символы, кроме начального символа (как в доказательстве теоремы 8.1.3). Если полученная грамматика содержит хотя бы одно правило, то  $L(G) \neq \emptyset$ .

**Упражнение 15.4.2.** Является ли непустым язык, порождаемый грамматикой

$$S \rightarrow aaCM,$$

$$K \rightarrow bT,$$

$$S \rightarrow aaaKF,$$

$$K \rightarrow aT,$$

$$M \rightarrow aMb,$$

$$T \rightarrow bKa,$$

$$M \rightarrow bMa,$$

$$T \rightarrow ab?$$

$$M \rightarrow \varepsilon,$$

$$C \rightarrow aCa,$$

$$C \rightarrow bCb,$$

## 15.5. Проблема бесконечности языка

**Теорема 15.5.1.** Существует алгоритм, позволяющий по произвольной контекстно-свободной грамматике  $G$  узнать, является ли бесконечным язык  $L(G)$ .

**Пример 15.5.2.** Рассмотрим контекстно-свободную грамматику  $G$  из примера 8.1.4.

Чтобы выяснить, является ли язык  $L(G)$  бесконечным, удалим сначала все бесполезные символы. Затем устраним все  $\varepsilon$ -правила. Так как  $W \xRightarrow{*} Z$  и  $Z \xRightarrow{*} W$ , то можно всюду заменить  $w$  на  $z$ . Получится грамматика

$$\begin{aligned}
S &\rightarrow VZ, & Z &\rightarrow aab, \\
T &\rightarrow aa, & Z &\rightarrow b, \\
T &\rightarrow bb, \\
V &\rightarrow aTb, \\
V &\rightarrow bTa,
\end{aligned}$$

эквивалентная исходной грамматике  $G$ . Очевидно, что эта грамматика не содержит "рекурсивных" нетерминальных символов. Следовательно, язык  $L(G)$  является конечным.

**Упражнение 15.5.3.** Является ли бесконечным язык, порождаемый грамматикой

$$\begin{aligned}
S &\rightarrow aF, & S &\rightarrow aaCM, & C &\rightarrow aCa, \\
F &\rightarrow J, & S &\rightarrow aaaKF, & C &\rightarrow bCb, \\
F &\rightarrow a, & M &\rightarrow aMb, & K &\rightarrow bT, \\
J &\rightarrow F, & M &\rightarrow bMa, & K &\rightarrow aT, \\
J &\rightarrow b, & M &\rightarrow \varepsilon, & T &\rightarrow bKa, \\
& & & & T &\rightarrow ab?
\end{aligned}$$

## 15.6. Проблема эквивалентности конечных автоматов

**Теорема 15.6.1.** Существует алгоритм, позволяющий по произвольным двум праволинейным грамматикам  $G_1$  и  $G_2$  узнать, верно ли, что  $L(G_1) \subseteq L(G_2)$ .

**Теорема 15.6.2.** Существует алгоритм, позволяющий по произвольным двум праволинейным грамматикам  $G_1$  и  $G_2$  узнать, верно ли, что  $L(G_1) = L(G_2)$ .

**Упражнение 15.6.3.** Эквивалентны ли грамматика

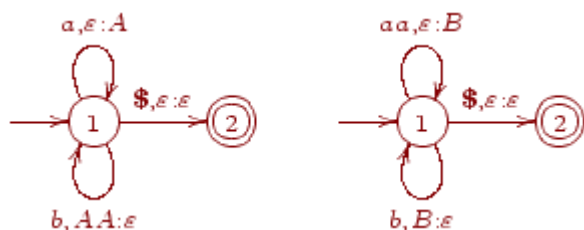
$$\begin{aligned}
S &\rightarrow aabS, \\
S &\rightarrow aaba \\
\text{и грамматика} \\
S &\rightarrow aT, \\
U &\rightarrow baT, \\
T &\rightarrow aU, \\
T &\rightarrow aba?
\end{aligned}$$

## 15.7\*. Проблема эквивалентности детерминированных МП-автоматов

**Теорема 15.7.1.** Существует алгоритм, позволяющий по произвольным двум детерминированным МП-автоматам  $M_1$  и  $M_2$  узнать, верно ли, что  $L(M_1) = L(M_2)$ .

Эту теорему доказал Жеро Сенизерг (Geraud Senizergues) в 1997 году. Упрощенное доказательство приводится в [Sti].

**Упражнение 15.7.2.** Эквивалентны ли два изображенных ниже МП-автомата над алфавитом  $\{a, b, \$\}$  ?



**Упражнение 15.7.3.** Эквивалентны ли два изображенных ниже МП-автомата над алфавитом  $\{a, b, \$\}$  ?



## 15.8\*. Классы P и NP

**Определение 15.8.1.** Пусть машина Тьюринга  $M$  допускает язык  $L$ . Говорят, что язык  $L$  *распознается за полиномиальное время* на машине  $M$ , если существует такой многочлен  $p$ , что любое слово  $w \in L$  принимается машиной  $M$  не более чем за  $p(|w|)$  тактов.

**Определение 15.8.2.** Через  $P$  обозначается класс тех языков, которые распознаются за полиномиальное время на некоторой детерминированной машине Тьюринга.

**Определение 15.8.3.** Через  $NP$  обозначается класс тех формальных языков, которые распознаются за полиномиальное время на некоторой (в общем случае недетерминированной) машине Тьюринга.

**Замечание 15.8.4.** Очевидно, что  $P \subseteq NP$ . Неизвестно, совпадают ли классы  $P$  и  $NP$ .

**Теорема 15.8.5.** Все языки из класса  $NP$  являются разрешимыми.

**Определение 15.8.6.** Всюду определенная функция  $f$  из  $\Sigma_1^*$  в  $\Sigma_2^*$  называется *вычислимой за полиномиальное время*, если ее вычисляет некоторая детерминированная машина Тьюринга  $M$  (если  $\Sigma_1 \neq \Sigma_2$ , то данная функция  $f$  рассматривается как частичная функция из  $(\Sigma_1 \cup \Sigma_2)^*$  в  $(\Sigma_1 \cup \Sigma_2)^*$ ) и существует такой многочлен  $p$ , что для любого слова  $w \in \Sigma_1^*$  машина  $M$  вычисляет значение  $f(w)$  не более чем за  $p(|w|)$  тактов.

**Определение 15.8.7.** Язык  $L_1 \subseteq \Sigma_1^*$  *полиномиально сводится* (is polynomial time reducible) к языку  $L_2 \subseteq \Sigma_2^*$ , если существует такая вычислимая за полиномиальное время всюду определенная функция  $f$  из  $\Sigma_1^*$  в  $\Sigma_2^*$ , что для любого слова  $w \in \Sigma_1^*$  условие  $f(w) \in L_2$  равносильно условию  $w \in L_1$ .

**Определение 15.8.8.** Формальный язык  $L$  называется *NP-сложным* (NP-hard}, если каждый язык из класса  $NP$  полиномиально сводится к языку  $L$ .

**Определение 15.8.9.** Формальный язык называется *NP-полным* (NP-complete), если он принадлежит классу  $NP$  и является NP-сложным.

**Определение 15.8.10.** Алгоритмическая проблема, относящаяся к некоторой задаче

распознавания, называется NP-полной, если множество кодов индивидуальных задач с ответом "да" является NP-полным языком.

**Теорема 15.8.11 (теорема Кука-Левина).** Проблема выполнимости булевых формул в конъюнктивной нормальной форме N-полна.

Доказательство можно найти, например, в [Сэв, с. 325-328] или [Гэрджо, с. 57-63].

**Упражнение 15.8.12.** Существует ли такой язык  $L \subseteq \Sigma^*$  из класса P, что язык  $\Sigma^* - L$  не принадлежит классу P?

**Упражнение 15.8.13.** Существуют ли такие языки  $L_1$  и  $L_2$  из класса P, что язык  $L_1 \cup L_2$  не принадлежит классу P?

**Упражнение 15.8.14.** Существуют ли такие языки  $L_1$  и  $L_2$  из класса NP, что язык  $L_1 \cup L_2$  не принадлежит классу NP?

## 15.9\*. Проблема неравенства регулярных выражений без итерации

**Теорема 15.9.1.** Проблема неравенства регулярных выражений без итерации (то есть регулярных выражений с нулевой звездной высотой) NP-полна.

*Доказательство.* По регулярному выражению без итерации легко построить конечный автомат с однобуквенными переходами, не содержащий циклов. Проблема неравенства таких конечных автоматов принадлежит классу NP: достаточно "угадать" слово, принадлежащее разности языков, распознаваемых двумя данными автоматами, и, продвигаясь по этому слову буква за буквой, подобно доказательству теоремы 2.7.1 вычислять, в каких состояниях автоматы могут оказаться. При этом длину слова можно ограничить максимумом числа состояний двух автоматов.

Осталось доказать, что проблема неравенства регулярных выражений без итерации NP-сложна. Для этого достаточно продемонстрировать, что к этой проблеме полиномиально сводится проблема выполнимости булевых формул в конъюнктивной нормальной форме (то есть множество кодов выполнимых булевых формул в конъюнктивной нормальной форме полиномиально сводится к множеству кодов пар неравных регулярных выражений без итерации). Искомое полиномиальное сведение может быть построено следующим образом.

Пусть дана булева формула  $C_1 \wedge C_2 \wedge \dots \wedge C_m$ , составленная из пропозициональных переменных  $x_1, x_2, \dots, x_n$  (при более формальном подходе можно считать  $x_i$

обозначением слова  $p\#^i$ , как это сделано в примере 7.2.8). Здесь для каждого  $j \leq m$  формула  $C_j$  является элементарной дизъюнкцией. Для краткости будем обозначать отрицание переменной  $x_i$  через  $\overline{x_i}$  (а не через  $\neg x_i$ , как в примере 7.2.8).

Построим два регулярных выражения над алфавитом  $\Sigma = \{a, b\}$ . В качестве первого регулярного выражения возьмем

$$\underbrace{(a+b) \cdot (a+b) \cdot \dots \cdot (a+b)}_{n \text{ раз}}.$$

Для удобства отождествим истинностные значения "ложь" и "истина" с символами  $a$  и  $b$  соответственно. Тогда оценку пропозициональных переменных  $x_1, x_2, \dots, x_n$  можно

естественным образом отождествить со словом длины  $n$  в алфавите  $\Sigma$ . Второе регулярное выражение  $e$  построим так, чтобы выполнялось равенство

$$L(e) = \{w \in \Sigma^* \mid |w| = n \text{ и } C_1 \wedge C_2 \wedge \dots \wedge C_m \text{ ложна при оценке } w\}.$$

Если  $m = 1$ , то это сделать легко. Возьмем в качестве  $e$  произведение  $n$  сомножителей, каждый из которых совпадает с одним из следующих четырех регулярных выражений:  $(a + b)$ ,  $a$ ,  $b$ ,  $0$ . При этом  $i$ -й сомножитель соответствует пропозициональной переменной  $x_i$  и выбирается следующим образом:

- если данная элементарная дизъюнкция не содержит ни литерала  $x_i$ , ни литерала  $\overline{x_i}$ , то используем выражение  $(a+b)$ ;
- если она содержит литерал  $x_i$ , но не содержит литерала  $\overline{x_i}$ , то используем выражение  $a$ ;
- если она содержит литерал  $\overline{x_i}$ , но не содержит литерала  $x_i$ , то используем выражение  $b$ ;
- если она содержит оба литерала  $x_i$  и  $\overline{x_i}$ , то используем выражение  $0$ .

Если  $m > 1$ , то построим по указанному методу для каждой из булевых формул  $C_1, C_2, \dots, C_m$  регулярное выражение и возьмем их сумму.

Легко видеть, что два построенных так регулярных выражения равны тогда и только тогда, когда булева формула  $C_1 \wedge C_2 \wedge \dots \wedge C_m$  невыполнима.

**Пример 15.9.2.** Пусть  $\Sigma = \{a, b\}$ . Регулярные выражения

$$(a+b)(a+b)(a+b)$$

и

$$ab(a+b) + (a+b)ab + b(a+b)(a+b) + (a+b)(a+b)a$$

равны, так как булева формула

$$(x_1 \vee \overline{x_2}) \wedge (x_2 \vee \overline{x_3}) \wedge \overline{x_1} \wedge x_3$$

невыполнима (или, другими словами, булева формула

$$(\overline{x_1} \wedge x_2) \vee (\overline{x_2} \wedge x_3) \vee x_1 \vee \overline{x_3}$$

является тавтологией).

**Упражнение 15.9.3.** Равны ли регулярные выражения

$$(a+b)(a+b)(a+b)$$

и

$$bb(a+b) + ba(a+b) + a(a+b)b + (a+b)aa \quad ?$$

**Упражнение 15.9.4.** Равны ли регулярные выражения

$$(a+b)(a+b)(a+b)(a+b)$$

и

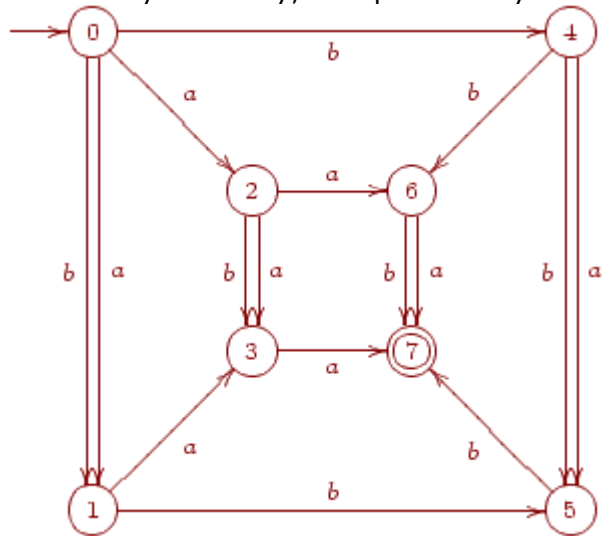
$$aab(a+b) + a(a+b)(a+b)b + (a+b)aa(a+b) + (a+b)bb(a+b) + \\ + (a+b)baa + bab(a+b) + b(a+b)(a+b)b \quad ?$$



**Упражнение 15.9.5.** Эквивалентен ли конечный автомат



конечному автомату, изображенному ниже?



© 2003-2008 INTUIT.ru. Все права защищены.

## Математическая теория формальных языков

### 16. Лекция: Алгоритмически неразрешимые проблемы: версия для печати и PDA

В данной лекции рассматриваются оказавшиеся неразрешимыми алгоритмические проблемы, связанные с контекстно-свободными языками. Доказывается неразрешимость проблем пустоты пересечения, бесконечности пересечения, однозначности, равенства, автоматности, контекстной свободности дополнения и пересечения контекстно-свободного языка. Доказательства приведены на практических примерах, а также приведены упражнения для самостоятельной проработки

В этой лекции рассматриваются оказавшиеся неразрешимыми алгоритмические проблемы, связанные с контекстно-свободными языками. Всюду предполагается, что в индивидуальных задачах каждый язык представлен контекстно-свободной грамматикой (но можно, конечно, использовать и автоматы с магазинной памятью).

В разделе 16.1 доказывается неразрешимость проблемы пустоты пересечения контекстно-свободных языков и проблемы бесконечности пересечения контекстно-свободных языков.

В разделе 16.2 доказывается неразрешимость проблемы однозначности контекстно-свободной грамматики.

В разделе 16.3 доказывается неразрешимость проблемы равенства контекстно-свободных языков.

В разделе 16.4 доказывается неразрешимость проблемы автоматности контекстно-свободного языка.

В разделе 16.5 доказывается неразрешимость проблем контекстной свободности дополнения контекстно-свободного языка и пересечения контекстно-свободных языков.

#### 16.1. Пересечение контекстно-свободных языков

**Определение 16.1.1.** Рассмотрим алфавит  $\Sigma_3 = \{a, b, c\}$ . Пусть  $\vec{x} = (x_1, \dots, x_n)$ , где  $x_i \in \{a, b\}^*$  для всех  $i$ . Обозначим через  $G(\vec{x})$  линейную грамматику

$\langle \{S\}, \Sigma_3, P, S \rangle$ , где  $P = \{S \rightarrow ba^i S x_i \mid 1 \leq i \leq n\} \cup \{S \rightarrow ba^i c x_i \mid 1 \leq i \leq n\}$ .

Обозначим через  $\mathcal{L}_{\vec{x}}$  язык, порождаемый грамматикой  $G(\vec{x})$ .

**Лемма 16.1.2.** Язык  $\mathcal{L}_{\vec{x}} \cap \mathcal{L}_{\vec{y}}$  является непустым тогда и только тогда, когда постовская система соответствия  $(\vec{x}, \vec{y})$  имеет решение.

**Пример 16.1.3.** Рассмотрим постовскую систему соответствия

$$\left[ \frac{b}{bba} \right], \left[ \frac{abbaa}{a} \right]$$

(то есть  $n = 2$ ,  $\vec{x} = (b, abbaa)$  и  $\vec{y} = (bba, a)$ ). Решениями этой системы являются последовательности  $(1, 1, 2)$ ,  $(1, 1, 2, 1, 1, 2)$  и т. д. Легко убедиться, что  $\mathcal{L}_{\vec{x}} \cap \mathcal{L}_{\vec{y}} = \{(baababa)^n c (bbabbaa)^n \mid n \geq 1\}$ .

**Теорема 16.1.4.** Пусть  $|\Sigma| \geq 2$ . Тогда не существует алгоритма, позволяющего по

произвольным контекстно-свободным грамматикам  $G_1$  и  $G_2$  над алфавитом  $\Sigma$  узнать, верно ли, что  $L(G_1) \cap L(G_2) = \emptyset$ .

**Доказательство.** Сначала докажем утверждение теоремы для случая  $|\Sigma| \geq 3$ . Из леммы 16.1.2 следует, что если бы проблема распознавания свойства  $L(G_1) \cap L(G_2) = \emptyset$  для контекстно-свободных грамматик над алфавитом  $\Sigma$  была разрешима, то проблема соответствий Поста тоже была бы разрешима. Поэтому из неразрешимости проблемы соответствий Поста следует неразрешимость проблемы распознавания свойства  $L(G_1) \cap L(G_2) = \emptyset$  для контекстно-свободных грамматик над алфавитом  $\Sigma$ .

Чтобы доказать утверждение теоремы для случая  $|\Sigma| = 2$  (например,  $\Sigma = \{d, e\}$ ), достаточно заменить в определении  $G(\vec{x})$  символ  $a$  на  $ede$ , символ  $b$  на  $edde$  и символ  $c$  на  $eddde$ .

**Лемма 16.1.5.** Язык  $\mathcal{L}_{\vec{x}} \cap \mathcal{L}_{\vec{y}}$  является бесконечным тогда и только тогда, когда постовская система соответствия  $(\vec{x}, \vec{y})$  имеет решение.

**Доказательство.** Если постовская система соответствия имеет хотя бы одно решение, то она имеет бесконечно много решений.

**Теорема 16.1.6.** Пусть  $|\Sigma| \geq 2$ . Тогда не существует алгоритма, позволяющего по произвольным контекстно-свободным грамматикам  $G_1$  и  $G_2$  над алфавитом  $\Sigma$  узнать, является ли бесконечным язык  $L(G_1) \cap L(G_2)$ .

**Упражнение 16.1.7.** Пусть  $\Sigma = \{a, b, c, d_1, d_2, d_3\}$ . Рассмотрим язык  $L_1$ , порождаемый грамматикой

$$F \rightarrow d_1 F a,$$

$$F \rightarrow d_2 F a a b,$$

$$F \rightarrow d_3 F b a a,$$

$$F \rightarrow c,$$

и язык  $L_2$ , порождаемый грамматикой

$$J \rightarrow d_1 J a a,$$

$$J \rightarrow d_2 J b b,$$

$$J \rightarrow d_3 J a,$$

$$J \rightarrow c.$$

Верно ли, что  $L_1 \cap L_2 = \{c\}$ ?

**Упражнение 16.1.8.** Пусть  $\Sigma = \{a, b, c, d_1, d_2, d_3\}$ . Рассмотрим язык  $L_1$ , порождаемый грамматикой

$$F \rightarrow d_1 F b,$$

$$F \rightarrow d_1 c b,$$

$$F \rightarrow d_2 F a,$$

$$F \rightarrow d_2 c a,$$

$$F \rightarrow d_3 F a b a,$$

$$F \rightarrow d_3 c a b a,$$

и язык  $L_2$ , порождаемый грамматикой

$$\begin{array}{ll}
J \rightarrow d_1 J, & J \rightarrow d_1 c, \\
J \rightarrow d_2 J a a, & J \rightarrow d_2 c a a, \\
J \rightarrow d_3 J b, & J \rightarrow d_3 c b.
\end{array}$$

Верно ли, что  $L_1 \cap L_2 = \emptyset$ ?

**Упражнение 16.1.9.** Пусть  $\Sigma = \{a, b, c\}$ . Рассмотрим язык  $L_1$ , порождаемый грамматикой

$$\begin{array}{ll}
F \rightarrow baFa, & F \rightarrow bac a, \\
F \rightarrow baaFaab, & F \rightarrow baac aab, \\
F \rightarrow baaaFbaa, & F \rightarrow baaacbaa,
\end{array}$$

и язык  $L_2$ , порождаемый грамматикой

$$\begin{array}{ll}
J \rightarrow baJaa, & J \rightarrow bacaa, \\
J \rightarrow baaJbb, & J \rightarrow baacbb, \\
J \rightarrow baaaJa, & J \rightarrow baaaca.
\end{array}$$

Верно ли, что  $L_1 \cap L_2 = \emptyset$ ?

## 16.2. Проблема однозначности

**Теорема 16.2.1.** Пусть  $|\Sigma| \geq 2$ . Тогда не существует алгоритма, позволяющего по произвольной контекстно-свободной грамматике  $G$  над алфавитом  $\Sigma$  узнать, является ли грамматика  $G$  однозначной.

*Доказательство.* Рассмотрим язык  $\mathcal{L}_{\vec{x}} \cup \mathcal{L}_{\vec{y}}$ . Следуя доказательству теоремы 9.4.3, построим грамматику  $G$  для этого языка, исходя из грамматик  $G(\vec{x})$  и  $G(\vec{y})$ .

Грамматика  $G$  является неоднозначной тогда и только тогда, когда постовская система соответствия  $(\vec{x}, \vec{y})$  имеет решение.

**Упражнение 16.2.2.** Однозначна ли контекстно-свободная грамматика

$$\begin{array}{ll}
S \rightarrow Q, & R \rightarrow baRbba, \\
S \rightarrow R, & R \rightarrow baaRa, \\
Q \rightarrow baQb, & R \rightarrow bacbba, \\
Q \rightarrow baaQabbaa, & R \rightarrow baaca? \\
Q \rightarrow bacb, & \\
Q \rightarrow baacabbaa, &
\end{array}$$

## 16.3. Дополнение контекстно-свободного языка

**Лемма 16.3.1.** Рассмотрим алфавит  $\Sigma_3 = \{a, b, c\}$ . Язык  $\Sigma_3^* - \mathcal{L}_{\vec{x}}$  является контекстно-свободным при любом  $\vec{x}$ .

**Пример 16.3.2.** Пусть  $\vec{x} = (b, abbaa)$ . Тогда язык  $\Sigma_3^* - \mathcal{L}_{\vec{x}}$  над алфавитом  $\Sigma_3 = \{a, b, c\}$  порождается контекстно-свободной грамматикой

$S \rightarrow \varepsilon,$	$R \rightarrow \varepsilon,$	$T_1 \rightarrow U_b,$
$S \rightarrow AW,$	$R \rightarrow BW,$	$T_2 \rightarrow U_a bbaa,$
$S \rightarrow bR,$	$R \rightarrow aaaW,$	$T_2 \rightarrow U_b baa,$
$A \rightarrow a,$	$R \rightarrow a,$	$T_2 \rightarrow U_b aa,$
$A \rightarrow c,$	$R \rightarrow aa,$	$T_2 \rightarrow U_a a,$
$B \rightarrow b,$	$R \rightarrow abRb,$	$T_2 \rightarrow U_a,$
$B \rightarrow c,$	$R \rightarrow aabRabbaa,$	$U_a \rightarrow WB,$
$Z \rightarrow a,$	$R \rightarrow acZWb,$	$U_a \rightarrow \varepsilon,$
$Z \rightarrow B,$	$R \rightarrow aacZWabbaa,$	$U_b \rightarrow WA,$
$W \rightarrow ZW,$	$R \rightarrow aBT_1,$	$U_b \rightarrow \varepsilon.$
$W \rightarrow \varepsilon,$	$R \rightarrow aaBT_2,$	

Заметим, что  $\{w \in \Sigma_3^* \mid T_i \xRightarrow{*} w\} = \Sigma_3^* - (\Sigma_3^* \cdot \{x_i\})$  для каждого  $i$ .

Язык  $\Sigma_3^* - \mathcal{L}_{\vec{x}}$  является даже линейным (чтобы получить линейную грамматику, достаточно "раскрыть" вспомогательные символы  $A$ ,  $B$  и  $Z$ ).

**Замечание 16.3.3.** Лемму 16.3.1 можно доказать, явно построив контекстно-свободную грамматику (как в примере 16.3.2), а можно и вывести из теоремы 12.2.7}, проверив, что  $\mathcal{L}_{\vec{x}}$  - детерминированный контекстно-свободный язык.

**Определение 16.3.4.** Пусть  $\Sigma_3 = \{a, b, c\}$ ,  $\vec{x} = (x_1, \dots, x_n)$ ,  $\vec{y} = (y_1, \dots, y_n)$ , где  $x_i \in \{a, b\}^*$  и  $y_i \in \{a, b\}^*$  для всех  $i$ . Обозначим через  $\mathcal{M}_{\vec{x}, \vec{y}}$  язык  $\Sigma_3^* - (\mathcal{L}_{\vec{x}} \cap \mathcal{L}_{\vec{y}})$ .

**Лемма 16.3.5.** Язык  $\mathcal{M}_{\vec{x}, \vec{y}}$  является контекстно-свободным при любых  $\vec{x}$  и  $\vec{y}$ .

*Доказательство.*  $\mathcal{M}_{\vec{x}, \vec{y}} = (\Sigma_3^* - \mathcal{L}_{\vec{x}}) \cup (\Sigma_3^* - \mathcal{L}_{\vec{y}})$ .

**Лемма 16.3.6.** Дополнение языка  $\mathcal{M}_{\vec{x}, \vec{y}}$  является непустым тогда и только тогда, когда постовская система соответствия  $(\vec{x}, \vec{y})$  имеет решение.

*Доказательство* Утверждение следует из леммы 16.1.2.

**Теорема 16.3.7.** Пусть  $|\Sigma| \geq 2$ . Тогда не существует алгоритма, позволяющего по произвольной контекстно-свободной грамматике  $G$  над алфавитом  $\Sigma$  узнать, верно ли, что  $L(G) = \Sigma^*$ .

*Доказательство* Очевидно, что  $L(G) = \Sigma^*$  тогда и только тогда, когда дополнение языка  $L(G)$  является пустым.

**Теорема 16.3.8.** Пусть  $|\Sigma| \geq 2$ . Тогда не существует алгоритма, позволяющего по произвольным контекстно-свободным грамматикам  $G_1$  и  $G_2$  над алфавитом  $\Sigma$  узнать,

верно ли, что  $L(G_1) = L(G_2)$ .

**Доказательство** Утверждение следует из предыдущей теоремы и примера 1.5.16.

**Теорема 16.3.9.** Пусть  $|\Sigma| \geq 2$ . Тогда не существует алгоритма, позволяющего по произвольным контекстно-свободным грамматикам  $G_1$  и  $G_2$  над алфавитом  $\Sigma$  узнать, верно ли, что  $L(G_1) \subseteq L(G_2)$ .

**Доказательство** Очевидно, что  $\Sigma^* \subseteq L(G)$  тогда и только тогда, когда  $L(G) = \Sigma^*$ .

**Лемма 16.3.10.** Дополнение языка  $\mathcal{M}_{\vec{x}, \vec{y}}$  является бесконечным тогда и только тогда, когда постовская система соответствия  $(\vec{x}, \vec{y})$  имеет решение.

**Теорема 16.3.11.** Пусть  $|\Sigma| \geq 2$ . Тогда не существует алгоритма, позволяющего по произвольной контекстно-свободной грамматике  $G$  над алфавитом  $\Sigma$  узнать, является ли бесконечным множество  $\Sigma^* - L(G)$ .

**Упражнение 16.3.12.** Рассмотрим язык, порождаемый грамматикой

$$\begin{array}{lll} S \rightarrow F, & F \rightarrow \varepsilon, & T \rightarrow a, \\ S \rightarrow T, & F \rightarrow aJ, & T \rightarrow bT, \\ & F \rightarrow bF, & T \rightarrow Tb, \\ & J \rightarrow aF, & T \rightarrow TaT, \\ & J \rightarrow bJ, & \end{array}$$

Содержит ли этот язык все слова из  $\{a, b\}^*$ ?

**Упражнение 16.3.13.** Рассмотрим язык, порождаемый грамматикой

$$\begin{array}{ll} S \rightarrow F, & T \rightarrow a, \\ S \rightarrow T, & T \rightarrow bT, \\ F \rightarrow aaF, & T \rightarrow Tb, \\ F \rightarrow bF, & T \rightarrow TaT, \\ F \rightarrow \varepsilon, & \end{array}$$

$\{a, b\}^*$ ?

## 16.4. Проблема автоматности

**Лемма 16.4.1.** Пусть  $\vec{x} = (x_1, \dots, x_n)$ ,  $\vec{y} = (y_1, \dots, y_n)$ , где  $x_i \in \{a, b\}^*$ ,  $y_i \in \{a, b\}^*$  и  $x_i y_i \neq \varepsilon$  для всех  $i$ . Тогда язык  $\mathcal{M}_{\vec{x}, \vec{y}}$  является автоматным в том и только том случае, когда постовская система соответствия  $(\vec{x}, \vec{y})$  не имеет решений.

**Доказательство** Пусть  $(i_1, \dots, i_k)$  - решение постовской системы соответствия  $(\vec{x}, \vec{y})$ , где  $x_i y_i \neq \varepsilon$  для всех  $i$ . Положим

$$u = ba^{i_k}ba^{i_k-1} \dots ba^{i_1},$$

$$v = x_{i_1} \dots x_{i_k},$$

$$L_0 = \{u\}^* \cdot \{c\} \cdot \{v\}^*.$$

Легко проверить, что  $u \neq \varepsilon$ ,  $v \neq \varepsilon$  и язык  $L_0$  является автоматным. Очевидно, что  $\mathcal{L}_{\vec{x}} \cap \mathcal{L}_{\vec{y}} \cap L_0 = \{u^m cv^m \mid m > 0\}$ .

Как было установлено в упражнении 3.4.2, язык  $\mathcal{L}_{\vec{x}} \cap \mathcal{L}_{\vec{y}} \cap L_0$  не является автоматным. Согласно теореме 3.2.1 язык  $\mathcal{L}_{\vec{x}} \cap \mathcal{L}_{\vec{y}}$  не является автоматным. Следовательно, и язык  $\mathcal{M}_{\vec{x}, \vec{y}}$  не является автоматным.

Обратно, если постовская система соответствия  $(\vec{x}, \vec{y})$  не имеет решений, то  $\mathcal{M}_{\vec{x}, \vec{y}} = \{a, b, c\}^*$ , а этот язык является автоматным.

**Теорема 16.4.2.** Пусть  $|\Sigma| \geq 2$ . Тогда не существует алгоритма, позволяющего по произвольной контекстно-свободной грамматике  $G$  над алфавитом  $\Sigma$  узнать, является ли автоматным язык  $L(G)$ .

**Доказательство** Докажем утверждение теоремы для случая  $|\Sigma| \geq 3$ . Из леммы 16.4.1 следует, что если бы проблема распознавания автоматности языка  $L(G)$  для контекстно-свободных грамматик над алфавитом  $\Sigma$  была разрешима, то также была бы разрешима проблема соответствий Поста для постовских систем соответствия

$((x_1, \dots, x_n), (y_1, \dots, y_n))$ , где  $x_i \in \{a, b\}^*$ ,  $y_i \in \{a, b\}^*$  и  $x_i y_i \neq \varepsilon$  для всех  $i$ . Но тогда была бы разрешима проблема соответствий Поста для всех постовских систем соответствия над алфавитом  $\{a, b\}$  (если  $x_i y_i = \varepsilon$  для некоторого  $i$ , то рассматриваемая постовская система соответствия имеет решение, а именно  $(i)$ ).

**Упражнение 16.4.3.** Является ли автоматным язык, порождаемый грамматикой

$$S \rightarrow RT,$$

$$R \rightarrow TR,$$

$$R \rightarrow a,$$

$$T \rightarrow TR,$$

$$T \rightarrow b?$$

**Упражнение 16.4.4.** Является ли автоматным язык, порождаемый грамматикой

$$R \rightarrow TR,$$

$$R \rightarrow a,$$

$$T \rightarrow R,$$

$$T \rightarrow bR?$$

**Упражнение 16.4.5.** Является ли автоматным язык, порождаемый грамматикой

$$R \rightarrow TR,$$

$$R \rightarrow a,$$

$$T \rightarrow RT,$$

$$T \rightarrow b?$$

**Упражнение 16.4.6.** Является ли автоматным язык, порождаемый грамматикой

$$S \rightarrow RT,$$

$$R \rightarrow TR,$$

$$R \rightarrow a,$$

$$T \rightarrow RT,$$

$$T \rightarrow b?$$

## 16.5. Проблемы контекстной свободности

**Определение 16.5.1.** Пусть  $\Sigma_3 = \{a, b, c\}$ ,  $\vec{x} = (x_1, \dots, x_n)$ ,  $\vec{y} = (y_1, \dots, y_n)$ , где  $x_i \in \{a, b\}^*$  и  $y_i \in \{a, b\}^*$  для всех  $i$ . Обозначим через  $\mathcal{K}_{\vec{x}, \vec{y}}$  язык  $\mathcal{L}_{\vec{x}} \cdot \{c\} \cdot \mathcal{L}_{\vec{y}}^R$ .

**Лемма 16.5.2.** Язык  $\mathcal{K}_{\vec{x}, \vec{y}}$  является контекстно-свободным при любых  $\vec{x}$  и  $\vec{y}$ .

*Доказательство* Утверждение следует из теорем 9.4.4 и 9.4.2.

**Лемма 16.5.3.** Рассмотрим алфавит  $\Sigma_3 = \{a, b, c\}$ . Пусть  $\vec{x} = (x_1, \dots, x_n)$  и  $\vec{y} = (y_1, \dots, y_n)$ , где  $x_i \in \{a, b\}^*$ ,  $y_i \in \{a, b\}^*$  и  $x_i y_i \neq \varepsilon$  для всех  $i$ . Тогда язык  $\mathcal{K}_{\vec{x}, \vec{y}} \cap \{zc z^R \mid z \in \Sigma_3^*\}$  является контекстно-свободным в том и только том случае, когда постовская система соответствия  $(\vec{x}, \vec{y})$  не имеет решений.

*Доказательство.* Пусть  $(i_1, \dots, i_k)$  - решение постовской системы соответствия  $(\vec{x}, \vec{y})$ , где  $x_i y_i \neq \varepsilon$  для всех  $i$ . Положим

$$u = ba^{i_k} ba^{i_{k-1}} \dots ba^{i_1},$$

$$v = x_{i_1} \dots x_{i_k},$$

$$L_0 = \{u\}^* \cdot \{c\} \cdot \{v\}^* \cdot \{c\} \cdot \{v^R\}^* \cdot \{c\} \cdot \{u^R\}^*.$$

Легко проверить, что  $u \neq \varepsilon$ ,  $v \neq \varepsilon$  и язык  $L_0$  является автоматным. Очевидно, что  $\mathcal{K}_{\vec{x}, \vec{y}} \cap \{zc z^R \mid z \in \Sigma_3^*\} \cap L_0 = \{u^m c v^m c (v^R)^m c (u^R)^m \mid m > 0\}$ .

Можно доказать (например, используя лемму 9.1.1), что язык

$\{u^m c v^m c (v^R)^m c (u^R)^m \mid m > 0\}$  не является контекстно-свободным. Согласно теореме 9.6.1 язык  $\mathcal{K}_{\vec{x}, \vec{y}} \cap \{zc z^R \mid z \in \Sigma_3^*\}$  также не является контекстно-свободным.

Обратно, если постовская система соответствия  $(\vec{x}, \vec{y})$  не имеет решений, то

$$\mathcal{K}_{\vec{x}, \vec{y}} \cap \{zc z^R \mid z \in \Sigma_3^*\} = \emptyset.$$

**Теорема 16.5.4.** Пусть  $|\Sigma| \geq 2$ . Тогда не существует алгоритма, позволяющего по произвольным контекстно-свободным грамматикам  $G_1$  и  $G_2$  над алфавитом  $\Sigma$  узнать, является ли контекстно-свободным язык  $L(G_1) \cap L(G_2)$ .

*Доказательство.* Достаточно построить по постовской системе соответствия  $(\vec{x}, \vec{y})$ , где



$\vec{x} = (x_1, \dots, x_n)$ ,  $\vec{y} = (y_1, \dots, y_n)$  и для всех  $i$  выполняется  $x_i \in \{a, b\}^*$ ,  $y_i \in \{a, b\}^*$  и  $x_i y_i \neq \varepsilon$ , контекстно-свободную грамматику  $G_1$ , порождающую язык  $\mathcal{K}_{\vec{x}, \vec{y}}$ , и контекстно-свободную грамматику  $G_2$ , порождающую язык  $\{zcz^R \mid z \in \Sigma_3^*\}$ .

С учетом леммы 16.5.3 неразрешимость рассматриваемой задачи сводится к неразрешимости проблемы соответствий Поста рассуждением, аналогичным приведенному в доказательстве теоремы 16.4.2.

**Лемма 16.5.5.** Рассмотрим алфавит  $\Sigma_3 = \{a, b, c\}$ . Язык

$\Sigma_3^* - (\mathcal{K}_{\vec{x}, \vec{y}} \cap \{zcz^R \mid z \in \Sigma_3^*\})$  является контекстно-свободным при любых  $\vec{x}$  и  $\vec{y}$ .

*Доказательство.* Положим  $L_0 = \{w \in \Sigma_3^* \mid |w|_c = 1\}$ . Язык

$\Sigma_3^* - (\mathcal{K}_{\vec{x}, \vec{y}} \cap \{zcz^R \mid z \in \Sigma_3^*\})$  можно представить в виде объединения пяти контекстно-свободных языков

$$L_1 = \{w \in \Sigma_3^* \mid |w|_c \neq 3\},$$

$$L_2 = \{v_1 c v_2 c v_3 c v_4 \mid v_1, v_2, v_3, v_4 \in \{a, b\}^*, v_1 \neq v_4^R\},$$

$$L_3 = \{v_1 c v_2 c v_3 c v_4 \mid v_1, v_2, v_3, v_4 \in \{a, b\}^*, v_2 \neq v_3^R\},$$

$$L_4 = ((\Sigma_3^* - \mathcal{L}_{\vec{x}}) \cap L_0) \cdot \{c\} \cdot L_0,$$

$$L_5 = L_0 \cdot \{c\} \cdot ((\Sigma_3^* - \mathcal{L}_{\vec{y}})^R \cap L_0).$$

**Теорема 16.5.6.** Пусть  $|\Sigma| \geq 2$ . Тогда не существует алгоритма, позволяющего по произвольной контекстно-свободной грамматике  $G$  над алфавитом  $\Sigma$  узнать, является ли контекстно-свободным язык  $\Sigma^* - L(G)$ .

*Доказательство.* Рассмотрим алфавит  $\Sigma_3 = \{a, b, c\}$ . Достаточно построить по постовской системе соответствия  $(\vec{x}, \vec{y})$ , где  $\vec{x} = (x_1, \dots, x_n)$ ,  $\vec{y} = (y_1, \dots, y_n)$  и для всех  $i$  выполняется  $x_i \in \{a, b\}^*$ ,  $y_i \in \{a, b\}^*$  и  $x_i y_i \neq \varepsilon$ , контекстно-свободную грамматику  $G$ , порождающую язык  $\Sigma_3^* - (\mathcal{K}_{\vec{x}, \vec{y}} \cap \{zcz^R \mid z \in \Sigma_3^*\})$ . С учетом леммы 16.5.5 неразрешимость рассматриваемой задачи сводится к неразрешимости проблемы соответствий Поста рассуждением, аналогичным приведенному в доказательстве теоремы 16.4.2.

**Лемма 16.5.7.** Рассмотрим алфавит  $\Sigma_3 = \{a, b, c\}$ . Язык  $\mathcal{K}_{\vec{x}, \vec{y}} \cap \{zcz^R \mid z \in \Sigma_3^*\}$  является контекстным при любых  $\vec{x}$  и  $\vec{y}$ .

**Теорема 16.5.8.** Пусть  $|\Sigma| \geq 2$ . Тогда не существует алгоритма, позволяющего по произвольной контекстной грамматике  $G$  над алфавитом  $\Sigma$  узнать, является ли контекстно-свободным язык  $L(G)$ .

*Доказательство.* Достаточно построить по постовской системе соответствия  $(\vec{x}, \vec{y})$ , где  $\vec{x} = (x_1, \dots, x_n)$ ,  $\vec{y} = (y_1, \dots, y_n)$  и для всех  $i$  выполняется  $x_i \in \{a, b\}^*$ ,  $y_i \in \{a, b\}^*$  и  $x_i y_i \neq \varepsilon$ , контекстную грамматику  $G$ , порождающую язык  $\mathcal{K}_{\vec{x}, \vec{y}} \cap \{zcz^R \mid z \in \Sigma_3^*\}$ .

**Упражнение 16.5.9.** Является ли контекстно-свободным язык  $\{a, b\}^* - L$ , где язык  $L$  порождается грамматикой

$$S \rightarrow \varepsilon,$$

$$S \rightarrow aSbS?$$

© 2003-2008 INTUIT.ru. Все права защищены.