Теория автоматов и формальных языков

Формальные языки и грамматики

Институт Информационных Технологий
ЧелГУ, 2013

Это скучный слайд с терминологией

Основные понятия

Алфавит – непустое конечное множество:

$$\Sigma = \{a_1, a_2, \dots, a_n\}$$

Пример:
Σ = {0,1} - бинарный
Σ = {a,b,c,…,z} – множество строчных букв английского алфавита
Σ – множество всех символов (печатных символов) таблицы ASCII

Цепочка (*слово*) над алфавитом **Σ** есть конечная последовательность его элементов.
Множество всех цепочек над алфавитом **Σ** обозначим Σ^*.
Множество всех непустых цепочек над алфавитом **Σ** обозначим Σ^+.

Длина цепочки ***x*** есть количество её элементов и обозначается $|x|$.
Цепочка нулевой длины называется пустой и обозначается ε.

Это скучный слайд с терминологией

Основные понятия

Пример:

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, aba, \ldots\}$$

$\Sigma^+ = ?$

Цепочки ***x*** и ***y*** *равны*, если они одинаковой длины и совпадают в точности до порядка символов, из которых они состоят.

Над цепочками ***x*** и ***y*** определена операция *сцепления (произведения, конкатенации)*, заключающаяся в дописывании всех символов цепочки ***y*** после символов цепочки ***x***. Получившаяся цепочка обозначается ***xy***.

Пример:
x=кото ***y***=пёс , тогда

xy=котопёс ***yx***=пёското

Это скучный слайд с терминологией

Основные понятия

Понятие языка

Язык **L** над алфавитом **Σ** есть некоторое множество цепочек над **Σ**.

$L = \emptyset$

$L = \{\varepsilon\}$ → Различные языки

Формальный язык **L** над алфавитом **Σ** - это язык, выделенный при помощи некоторого конечного множества *формальных правил*.

Над языками определена операция *сцепления (произведения, конкатенации)*:

L - язык

M - язык

⇒ $LM = \{xy \mid x \in L, y \in M\}$ - язык

$\{\varepsilon\}L = L\{\varepsilon\} = L$

Примеры языков

множество всех слов над $\{a, b\}$

множество $\{a^n\}$, где n — простое число, а a^n означает, что a повторяется n раз

множество синтаксически корректных программ в данном языке программирования

Это скучный слайд с терминологией

Основные понятия

Понятие языка – образование новых языков

- *Конкатенация (сцепление)* L_1L_2 содержит все слова, удовлетворяющие форме vw, где v — это слово из L_1, а w — слово из L_2.
- *Пересечение* $L_1 \cap L_2$ содержит все слова, содержащиеся и в L_1, и в L_2.
- *Объединение* $L_1 \cup L_2$ содержит все слова, содержащиеся или в L_1, или в L_2.
- *Дополнение* языка L_1 содержит все слова алфавита, которые не содержатся в L_1.
- *Правое отношение* L_1 / L_2 содержит все слова v, для которых существует слово w в L_2 такое, что vw находидось в L_1.

Пример конкатенации:

L1={к,п,т}, *L2*={ε,орт} *L1L2=?*

Это скучный слайд с терминологией

Основные понятия

Понятие языка – образование новых языков

- *Замыкание Клини* L_1^* содержит все слова, которые могут быть записаны в форме $w_1w_2...w_n$, где w_i содержится в L_1 и $n \geq 0$. Следует помнить, что это включает и пустое слово ε, так как $n = 0$ допустимо по условию.
- *Обращение* L_1^R содержит обращенные слова из L_1.

Примечание:

обращением (или *реверсом)* цепочки называется цепочка, символы которой записаны в обратном порядке.

Задача: определить замыкание Клини для *L1*={ε,00,1}?

Это скучный слайд с терминологией

Основные понятия

$$L^* = \bigcup_{i=0}^{\infty} L^i$$ - итерация

$$L^+ = \bigcup_{i=1}^{\infty} L^i$$ - усечённая итерация

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

$$L^1 = L$$

$$L^{n+1} = L^n L = L L^n$$

i - степень языка

Пример:

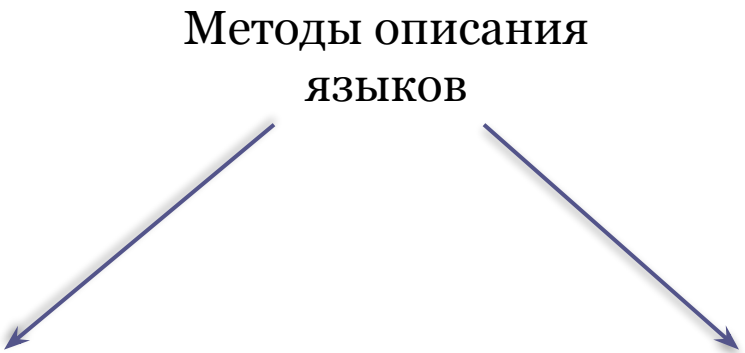
$$L = \{a\}$$

$$L^* = \{\varepsilon, a, aa, aaa, \dots\}$$

$$L^+ = \{a, aa, aaa, \dots\}$$

Это скучный слайд с терминологией

Описание языков



Порождающие
Предполагается наличие алгоритма, последовательно порождающего все правильные предложения языка.

Распознающие
Предполагается наличие алгоритма, определяющего принадлежность любой фразы данному языку.

Это скучный слайд с терминологией

Основные понятия

Понятие грамматики – способы задания

Простым перечислением слов, входящих в данный язык. Этот способ, в основном, применим для определения конечных языков и языков простой структуры.

Словами, порождёнными некоторой формальной грамматикой (иерархия Хомского)

Словами, порождёнными регулярным выражением

Словами, распознаваемыми некоторым конечным автоматом

Словами, порождёнными БНФ-конструкцией (Форма Бэкуса — Наура)

Основные понятия

Понятие грамматики – способы задания

Форма Бэкуса — Наура

<Терминал>:=Задание_терминала

<…> - ограничивают (выделяют) терминалы

| - разделяет несколько альтернативных терминалов

[…] – конструкция встречается 1 раз или отсутствует

{…} – конструкция повторяется некоторое (возможно нулевое) количество раз

Определение идентификатора:
<Буква> := A|B|C|...|Z
<Цифра> := 0|1| ... |9
<Идент> := <Буква> {<Буква>|<Цифра>}

A B A1 B2C3 DA123

http://kitaev2005.narod.ru/tr1.htm

Основные понятия

Понятие грамматики – способы задания

Форма Бэкуса — Наура

Грамматика целых чисел без знака:
<число> := <цифра>|<цифра><число>
<цифра> := 0|1|2|...|9

Формула с плюсами и минусами без скобок.
<формула> := <число>|<формула><знак><число>
<знак> := +|-

Задачи:

1. Описать терминал <формула> (с +,-,*,/ [и круглыми скобками])
2. Описать оператор while, а затем if (с некоторой степенью упрощения).

http://kitaev2005.narod.ru/tr1.htm

Это скучный слайд с терминологией

Основные понятия

Порождающая грамматика – это четвёрка:

$G = (V_T, V_N, P, S)$

V_T - конечный алфавит, определяющий множество *терминальных символов*

V_N - конечный алфавит, определяющий множество *нетерминальных символов*

$$V = V_T \cup V_N$$
$$V_T \cap V_N = \emptyset$$

P - конечное множество *правил вывода (продукций)*, т.е. правил вида:

$$u \to v$$
$$u, v \in (V_T \cup V_N)^*$$

S - *начальный символ (аксиома грамматики)*

$S \in V_N$

Это скучный слайд с терминологией

Основные понятия

Аксиомой называется символ в левой части первого правила вывода грамматики.
Цепочки языка, порождённого грамматикой, состоят из терминальных символов.

Будем обозначать терминальные символы - строчными, а нетерминальные символы – заглавными буквами.

В грамматике **G** цепочка ***x*** *порождает* цепочку ***y*** (обозначается $x \Rightarrow y$), если:

$$\begin{array}{l} x = \alpha u \beta \\ y = \alpha v \beta \\ u \to v \end{array}$$

В этом случае также говорят, что ***y*** *выводится* из ***x***.

Языком, порождаемым грамматикой G, называется множество терминальных цепочек, выводимых из некоторой аксиомы:

$$L(G) = \{x \mid x \in V_T^*; S \Longrightarrow x\}$$

Пример определения языка

Задача:	**Решение:**
$G = (V_T, V_N, P, S)$ $V_T = \{a, b\}$ $V_N = \{S\}$ $P = \{S \to aSb, S \to ab\}$ Определить язык, определяемый данной грамматикой.	$S \to ab$ $S \to aSb \Rightarrow aabb$ $S \to aSb \Rightarrow aaSbb \Rightarrow aaabbb$ … Таким образом: $L(G) = \{a^n b^n \mid n \in \mathbb{N}\}$

Это скучный слайд с терминологией

Классификация Хомского

Грамматики типа 0:

На правила вывода не накладывается никаких ограничений.

Грамматики типа 1 (контекстные грамматики, контекстно-зависимые):

Все правила таких грамматик имеют вид:

$$xAy \to x\varphi y, \quad A \in V_N, \quad x,y \in V^*, \quad \varphi \in V^+$$

(Прим. Если α → β, то |α| ≤ |β|)

Грамматики типа 2 (контекстно-свободные грамматики):

Все правила таких грамматик имеют вид:

$$A \to \varphi, \quad A \in V_N, \quad \varphi \in V^+$$

Грамматики типа 3 (автоматные грамматики, регулярные):

Леворекурсивные содержат только правила вида:

$$A \to Aa, A \to a, \qquad A \in V_N, \qquad a \in (V_T \cup V_N)^*$$

Праворекурсивные содержат только правила вида:

$$A \to aA, A \to a, \qquad A \in V_N, \qquad a \in (V_T \cup V_N)^*$$

Пример 1

Дана грамматика, выписать все порождаемые её цепочки и определить длину их вывода:

$$G = (V_T, V_N, P, S)$$

$$V_T = \{a, d, e\}$$

$$V_N = \{B, C, S\}$$

$$P = \{S \to aB, B \to Cd, B \to dC, C \to e\}$$

Пример 1

Дана грамматика, выписать все порождаемые её цепочки и определить длину их вывода:

$$G = (V_T, V_N, P, S)$$

$$V_T = \{a, d, e\}$$

$$V_N = \{B, C, S\}$$

$$P = \{S \to aB, B \to Cd, B \to dC, C \to e\}$$

$$S \to aB \to aCd \to aed$$

$$S \to aB \to adC \to ade$$

Длина вывода всех (обеих, их всего 2) цепочек равна 3.

Пример 2

Дана грамматика, определить, принадлежит ли некоторая цепочка порождаемому её языку:

$$G = (V_T, V_N, P, C)$$

$$V_T = \{a, b, c, d, e\}$$

$$V_N = \{A, B, C, D, E\}$$

$$P = \{A \to ed, B \to Ab, C \to Bc, C \to dD, D \to Ae, E \to bc\}$$

Цепочка для проверки: $eadabcb$

Пример 2

Дана грамматика, определить, принадлежит ли некоторая цепочка порождаемому её языку:

$$G = (V_T, V_N, P, C)$$

$$V_T = \{a, b, c, d, e\}$$

$$V_N = \{A, B, C, D, E\}$$

$$P = \{A \to ed, B \to Ab, C \to Bc, C \to dD, D \to Ae, E \to bc\}$$

Цепочка для проверки: $eadabcb$

Выведем всевозможные цепочки из указанных правил вывода:

$$C \to Bc \to Abc \to edbc$$

$$C \to dD \to dAe \to dede$$

Указанную цепочку вывести никак не получается, следовательно, она не принадлежит порождённому указанной грамматикой языку.

Пример 3

Построить контекстно-свободную грамматику, порождающую цепочки из 0 и 1 с одинаковым количеством и тех и других.

$G = (V_T, V_N, P, S)$

$V_T = \{0, 1\}$

$V_N = \{S\}$

Пример 3

Построить контекстно-свободную грамматику, порождающую цепочки из 0 и 1 с одинаковым количеством и тех и других.

$$G = (V_T, V_N, P, S)$$

$$V_T = \{0, 1\}$$

$$V_N = \{S\}$$

$$P = \{S \to \varepsilon, S \to 0S1, S \to 1S0, S \to 10S, S \to 01S, S \to S10, S \to S01\}$$

Пример 4 и 5

4. Описать грамматику языка, содержащего палиндромы. Определить тип грамматики.

5. Описать грамматику, которая представляет выражения с операциями + и *. (Пусть в образовании идентификаторов переменных участвуют только 0,1,a,b). Определить тип грамматики.

Пример 4 и 5

4. Описать грамматику языка, содержащего палиндромы. Определить тип грамматики.

5. Описать грамматику, которая представляет выражения с операциями + и *. (Пусть в образовании идентификаторов переменных участвуют только 0,1,a,b). Определить тип грамматики.

Продукции для грамматики 4

$$P \to \varepsilon$$
$$P \to 0$$
$$P \to 1$$
$$P \to 0P0$$
$$P \to 1P1$$

Продукции для грамматики 5

$$E \to I$$
$$E \to E + E$$
$$E \to E * E$$
$$E \to (E)$$
$$I \to a$$
$$I \to b$$
$$I \to Ia$$
$$I \to Ib$$
$$I \to I0$$
$$I \to I1$$

Это скучный слайд с терминологией

Регулярные выражения

Регулярные выражения задают языки.

Обозн. Пусть *E* - регулярное выражение, *L(E)* – язык, заданный этим выражением.

Пример: **01*+10***

Базис при построении регулярных выражений:

1. Константы ε и ∅. Определяют языки L(ε)={ε} и L(∅)={∅}.

2. Если *a* – произвольный символ алфавита, то **a** – регулярное выражение, определяющее язык L(**a**)={*a*}.

3. Переменная, обозначенная прописной буквой, например, L, представляет язык.

Это скучный слайд с терминологией

Регулярные выражения

Индукция (операторы и скобки):

1. Если E и F — регулярные выражения, то $E + F$ — регулярное выражение, определяющее объединение языков $L(E)$ и $L(F)$, т.е. $L(E + F) = L(E) \cup L(F)$.
2. Если E и F — регулярные выражения, то EF — регулярное выражение, определяющее конкатенацию языков $L(E)$ и $L(F)$. Таким образом, $L(EF) = L(E)L(F)$.
3. Если E — регулярное выражение, то E^* — регулярное выражение, определяющее итерацию языка $L(E)$. Таким образом, $L(E^*) = (L(E))^*$.
4. Если E — регулярное выражение, то (E) — регулярное выражение, определяющее тот же язык $L(E)$, что и выражение E. Формально, $L((E)) = L(E)$.

Пример 1

Написать регулярное выражение для множества цепочек, состоящих из чередующихся нулей и единиц.

Пример 1

Написать регулярное выражение для множества цепочек, состоящих из чередующихся нулей и единиц.

Первый вариант решения:

$$(01)^* + (10)^* + 0(10)^* + 1(01)^*$$

Пример 1

Написать регулярное выражение для множества цепочек, состоящих из чередующихся нулей и единиц.

Первый вариант решения:

$$(01)^* + (10)^* + 0(10)^* + 1(01)^*$$

А если немного подумать, то получится и второй вариант решения:

$$(\varepsilon + 1)(01)^*(\varepsilon + 0)$$

Задачи

Напишите регулярные выражения для следующих языков:

а) (*) множество цепочек с алфавитом $\{a, b, c\}$, содержащих хотя бы один символ a и хотя бы один символ b;

б) множество цепочек из нулей и единиц, в которых десятый от правого края символ равен 1;

в) множество цепочек из нулей и единиц, содержащих не более одной пары последовательных единиц.

(!) Напишите регулярные выражения для следующих языков:

а) (*) множество всех цепочек из нулей и единиц, в которых каждая пара смежных нулей находится перед парой смежных единиц;

б) множество цепочек, состоящих из нулей и единиц, в которых число нулей кратно пяти.