# ТЕОРЕТИЧЕСКАЯ ИНФОРМАТИКА

## Алфавиты, слова, языки, алгоритмические проблемы

Кафедра информатики и вычислительной техники

Доцент, к.т.н. Дамов Михаил Витальевич

Красноярск - 2016

# Цели и задачи

1. Ввести подходящий формализм для работы с текстами - представлениями данных.

- Основные понятия - алфавит, слово и язык.

2. Показать, как использовать введённые понятия, применять их для получения формальных представлений алгоритмических проблем.

- Проблемы принадлежности (разрешимости)

- Оптимизационные проблемы

3. Рассмотреть некоторые вопросы, связанные со сжатием текстов.

- Сложность по Колмогорову

# Формальный язык

Алфавитом называется любое непустое конечное множество.
Каждый элемент алфавита называется символом.
Алфавит языка – множество символов (букв)
Язык – множество строк
Строка (слово) – последовательность символов
Примеры:
“студент”, “123”, “house”
∑={‘0’, ‘1’, ‘2’, ’3’, ’4’, ’5’, ’6’, ’7’, ’8’, ’9’}
∑={‘a’, ‘b’, ‘c’, …, ’z’}  ∑={‘а’, ‘б’, ‘в’, …, ’я’}

# Примеры стандартных алфавитов

$\sum_{bool}$={0, 1} - логический (Булевый) алфавит

$\sum_{lat}$={a, b, c, …, z} - латинский алфавит

$\sum_{keyboard}=\sum_{lat} U\{….\}$ – алфавит символов, которые можно набрать на клавиатуре

$\sum_{m}$= {0, 1, …, m-1} , m> 1 – алфавит для записи чисел в m-ичной системе счисления

$\sum_{logic}$= {0, 1, *x*, (, ), ∩, U, ¬} – алфавит формул алгебры логики

# Алфавит и строки

Будем использовать алфавит из двух букв

$\sum$={a, b}

Строки (слова)

a, ab, abba, baba, aaabbbaabab

u = ab

v = bbbaaa

w = abba

# Операции над строками

$w = a_1a_2\ldots a_n$ $v = b_1b_2\ldots b_m$

w = abba v = bbbaaa

**Конкатенация:** **Длина строки:**

$wv = a_1a_2\ldots a_nb_1b_2\ldots b_m$ $|w| = m$

wv = abbabbbaaa |abba| = 4

**Обращение:** **Длина конкатенации строк:**

$v^R = b_m\ldots b_2b_1$ $|uv| = |u| + |v|$

$v^R$= aaabbb

# Операции над строками

**Пустая строка:**

Строка, не содержащая букв: λ  $|\lambda| = 0$

$\lambda w = w\lambda = w$  $\lambda abba = abba\lambda = abba$

**Подстрока:**

Строка: abbab

Подстроки: ab, abba, b, bbab, λ

# Операции над строками

**Префиксы и суффиксы:**

Строка: abba

| Префиксы: | | Суффиксы: | w = uv |
|---|---|---|---|
| λ | abba | u - префикс | |
| a | bba | v - суффикс | |
| ab | ba | | |
| abb | a | | |
| abba | λ | | |

# Операции над строками

**Итерация:** $w^n = \underbrace{ww \cdots w}_{n} = \lambda$

$(abba)^2 = abbaabba = ab^2a^2b^2a$ $(abba)^0 = \lambda$

**Звезда Клини:**

$\sum^*$ - множество всех возможных слов в алфавите $\sum$

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

$\sum^* = \{\lambda, a, b, aa, ab, ba, bb, \ldots\}$

# Операции над строками

**Плюс Клини:**

$\sum^{+} = \sum^{*} - \lambda$ $\sum^{+} = \{a, b, aa, ab, ba, bb, \ldots\}$

Язык в алфавите $\sum$ - любое подмножество $\sum^{*}$

# Операции над языками

**Обычные теоретико-множественные операции:**

Объединение: {a, ab, aaaa} U {bb, ab} = {a, ab, bb, aaaa}

Пересечение: {a, ab, aaaa} ∩ {bb, ab} = {ab}

Разность: {a, ab, aaaa} - {bb, ab} = {a, aaaa}

**Дополнение:** $\bar{L}$ =∑*-L

$\overline{\{a, ba\}}$={λ, b, aa, ab, ba, bb, aaa, …}

# Операции над языками

**Обращение**

$L^R$ $\{w^R: w \in L\}$

$\{ab, aab, baba\}^R = \{ba, baa, abab\}$

**Конкатенация**

$L_1L_2 = \{xy: x \in L_1, y \in L_2\}$

$\{a, ab, ba\}\{b, aa\} = \{ab, aaa, abb, abaa, bab, baaa\}$

# Операции над языками

**Итерация**

$$L^n = \underbrace{LL\cdots L}_{n}$$

$\{a, b\}^3 = \{a, b\}\ \{a, b\}\ \{a, b\} = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$

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

$\{a, b\}^0 = \{\lambda\}$

# Операции над языками

**Звезда Клини (замыкание)**

$L^* = L^0 \cup L^1 \cup L^2 \cup \ldots$

$$\{a, bb\}^* = \left\{ \begin{array}{c} \lambda \\ a, bb \\ aa, abb, bba, bbbb \\ aaa, aabb, abba, bbbb, \ldots \end{array} \right\}$$

# Операции над языками

**Плюс Клини (положительное замыкание)**

$L^+ = L^1 \cup L^2 \cup \ldots = L^* - \{\lambda\}$

$$\{a, bb\}^+ = \left\{ \begin{array}{c} a, bb \\ aa, abb, bba, bbbb \\ aaa, aabb, abba, bbbb, \ldots \end{array} \right\}$$

# Грамматики

Грамматики определяют языки, является ли данное предложение правильным предложением данного языка.

Пример: грамматика русского языка

<предложение> → <подлежащее> <сказуемое> <дополнение>

<подлежащее> → <существительное>

<сказуемое> → <глагол>

<дополнение> → <наречие>

<существительное> → птица | студент

<сказуемое> → летает | учится

<наречие> → высоко | хорошо

# Вывод предложения

<предложение> => <подлежащее> <сказуемое> <дополнение> =>

=> <существительное> <глагол> <наречие> =>

=> Птица летает высоко

**Возможные предложения**

Птица летает хорошо

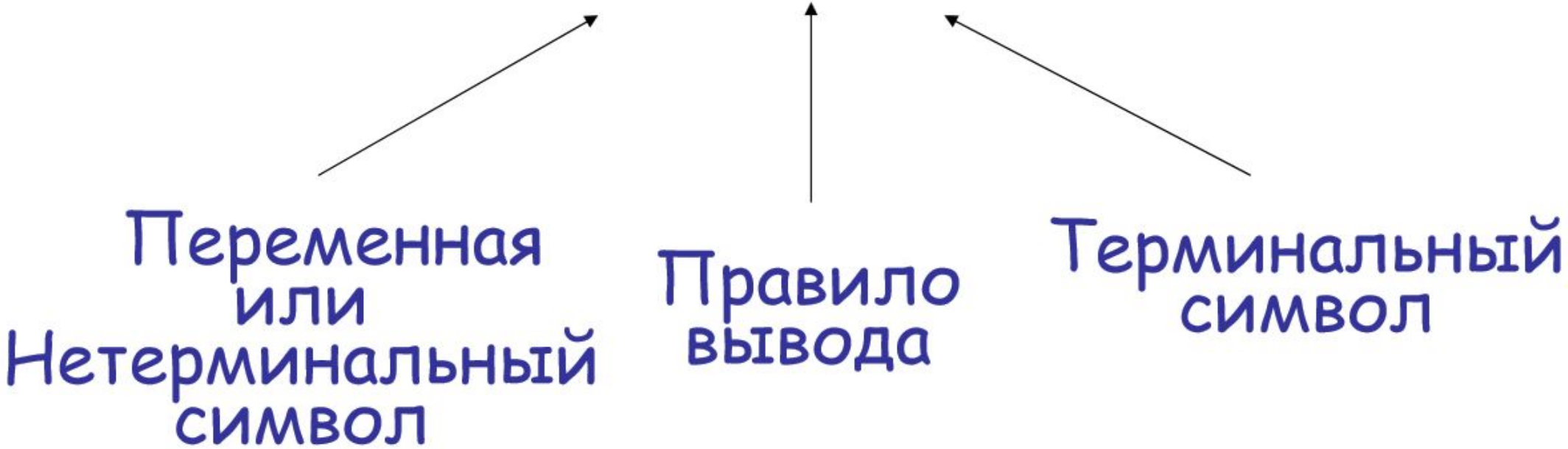
Птица учится высоко

Студент летает хорошо

# Обозначения

<глагол> → *летает*

<глагол> → *учится*



# Пример формальной грамматики

Грамматика: $S \to aSb$

$S \to \lambda$

Вывод предложения $ab$ :

$$S \Rightarrow aSb \Rightarrow ab$$

$S \to aSb$ $\qquad$ $S \to \lambda$

# Определение формальной грамматики

$$G = \{V, T, S, P\}$$

V = Множество нетерминальных символов

T = Множество терминальных символов

S = Начальный символ

P = Множество правил вывода (продукций)

**Пример:**

Грамматика $S \rightarrow aSb$, $S \rightarrow \lambda$

V = {S}

T = {a, b}

P = {$S \rightarrow aSb$, $S \rightarrow \lambda$}

# Язык, порождаемый грамматикой

**Определение:**

Для грамматики G с начальным символом S

$$L(G) = \{w: S \Rightarrow w\}$$

-язык, порождаемый этой грамматикой

**Пример:**

Грамматика G $\{S \rightarrow Ab, A \rightarrow aAb, A \rightarrow \lambda\}$

$$L(G) = \{a^n b^n b: n \geq 0\}$$

поскольку $S \Rightarrow a^n b^n b$ и никакие друге слова не выводимы

# Алгоритмические проблемы

Любая программа (алгоритм) A выполняет отображение:

$$A: \sum_1{}^* \rightarrow \sum_2{}^*$$

-входы представлены как слова над алфавитом $\sum_1$

-выходы представлены как слова над алфавитом $\sum_2$

-A однозначно определяет выход по каждому входу

Для некоторых алгоритма A и входа *x* обозначим записью A(x) выход алгоритма A для этого входа. Будем говорить, что два алгоритма (программы) *A и B* эквивалентны, если они работают над одним и тем же алфавитом $\sum$ и при этом *A(x) = B(x)* для всех x $\epsilon \sum^*$.

# Проблема принадлежности

**Обычные теоретико-множественные операции:**

Объединение: {a, ab, aaaa} U {bb, ab} = {a, ab, bb, aaaa}

Пересечение: {a, ab, aaaa} ∩ {bb, ab} = {ab}

Разность: {a, ab, aaaa} - {bb, ab} = {a, aaaa}

**Дополнение:** $\bar{L}$ =∑*-L

$\overline{\{a, ba\}}$={λ, b, aa, ab, ba, bb, aaa, ...}

$$A(x) = \begin{cases} 1, \text{если } x \in L \\ 0, \text{если } x \notin L \end{cases}$$

# Оптимизационная проблема

$U = \{\sum_I, \sum_O, L, M, cost, goal)$

$\sum_I$ – входной алфавит

$\sum_O$ – выходной алфавит

L – язык подходящих входов

M – множество допустимых решений

Cost – функция стоимости

Goal – цель (максимизация или минимизация)

Пример: задача коммивояжера

# Сложность по Колмогорову

Колмогоровская сложность объекта есть мера вычислительных ресурсов, необходимых для точного описания этого объекта

Сложность строки — это длина описания этой строки на некотором универсальном языке описания.

Пример:

Две строки:

abababababababababababababababababababababababab = $(ab)^{25}$

4c1j5b2p0cv4w1x8rx2y39umgw5q85s7uraqbjfdppa0q

# Задание на лабораторную работу

Написать программу на языке C++, в которой необходимо выполнить следующие операции:

1. Задать алфавит языка (3 – 5 латинских букв)
2. Задать максимально возможную длину слова (5 – 7 символов)
3. Построить словарь языка.
4. Используя грамматику слайда 16, отнести случайным образом слова к трем классам.
5. Сгенерировать текст из заданного количества случайных предложений.
6. Сжать текст, используя операцию итерации.
7. Вывести в отдельные файлы: словарь, текст, сжатый текст