

**Курс «Алгоритмы и алгоритмические языки»
1 семестр 2015/2016**

Лекция 4

Нормальные алгоритмы Маркова

Определение нормального алгоритма Маркова (НАМ)

V – алфавит основных символов

V' – алфавит маркеров

$\sigma, \sigma' \in (V \cup V')^*$

Подстановка: $\sigma \rightarrow \sigma'$ переводит

слово $\tau = \alpha\sigma\beta \in (V \cup V')^*$ в слово $\tau' = \alpha\sigma'\beta \in (V \cup V')^*$

подслова α и β могут быть пустыми (ε)

Помимо символов алфавита $V \cup V'$ в подстановках используются метасимволы « \rightarrow » (**стрелка**) отделяет левую часть подстановки от правой и « \cdot » (**точка**) отмечает терминальную подстановку

Нормальные алгоритмы Маркова

Процедура интерпретации НАМ

- (1) Положить $i = 0$.
- (2) Положить $j = 1$.
- (3) Если правило p_j применимо к σ_i , перейти к шагу (5).
- (4) Положить $j = j + 1$. Если $j \leq n$, то перейти к шагу (3).
В противном случае – остановка.
- (5) Применить p_j к σ_i и найти σ_{i+1} . Положить $i = i + 1$.
Если p_j – нетерминальное правило, то перейти к 2.
В противном случае – остановка.

Говорят, что НАМ *применим* к слову σ_0 , если в результате произойдет остановка.

Терминальное правило содержит метасимвол \cdot (точка).

Иногда вместо $\rightarrow\cdot$ используется метасимвол \vdash

Нормальные алгоритмы Маркова

Пример. Сложение чисел в единичной системе счисления:

$$V = \{ +, / \}, V' = \{ \}.$$

Правила подстановок: $\{ | + \rightarrow + |; + | \rightarrow |; | \rightarrow \cdot | \}$

Пример применения алгоритма:

$$\begin{aligned} \langle\langle | | | | + | | + | | | \rangle\rangle &= \langle\langle | | | | \color{red}{| +} | | + | | | \rangle\rangle \xRightarrow{(1)} \langle\langle | | | \color{red}{| +} | | | + | | | \rangle\rangle \\ \xRightarrow{(1)} \langle\langle | \color{red}{| +} | | | | + | | | \rangle\rangle &\xRightarrow{(1)} \langle\langle | \color{red}{+} | | | | | + | | | \rangle\rangle \\ \xRightarrow{(1)} \langle\langle + | | | | | | \color{red}{| +} | | | \rangle\rangle &\xRightarrow{(1)} \langle\langle + | | | | | | \color{red}{| +} | | | \rangle\rangle \xRightarrow{(1)} \dots \\ \xRightarrow{(1)} \langle\langle + \color{red}{+} | | | | | | | | \rangle\rangle &\xRightarrow{(2)} \langle\langle \color{red}{+} | | | | | | | | \rangle\rangle \xRightarrow{(2)} \langle\langle | | | | | | | | \rangle\rangle \xRightarrow{(3)} \end{aligned}$$

Первое правило «перегоняет» плюсы налево до упора, второе правило их «стирает», третье правило «убеждается», что плюсов не осталось.

Нормальные алгоритмы Маркова

Заключительные замечания

- ◇ **Тезис Маркова.** Любой алгоритм в алфавите V может быть представлен нормальным алгоритмом Маркова над алфавитом V .
- ◇ Примерно так же, как и для МТ, можно доказать алгоритмическую неразрешимость проблемы останова и самоприменимости.
- ◇ Существуют различные НАМ решения одной и той же задачи. Проблема построения алгоритма, который может определить эквивалентность любых двух НАМ, алгоритмически неразрешима.
- ◇ Можно построить универсальный НАМ U , который мог бы *интерпретировать* любой нормальный алгоритм, включая самого себя.

Заключительные замечания

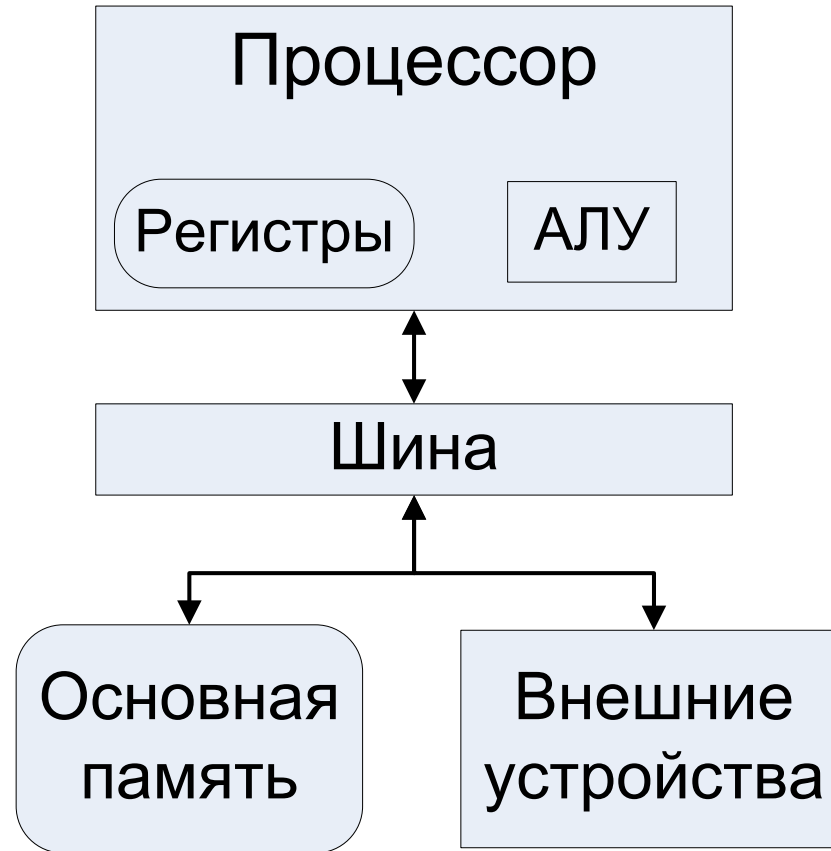
- ◇ Можно доказать эквивалентность двух формальных систем Тьюринга и Маркова конструктивным путем: построить универсальную МТ, которая могла бы интерпретировать любой НАМ и, наоборот, построить универсальный НАМ, который интерпретирует любую МТ.
- ◇ Существуют и другие формальные описания алгоритмов: машина Поста, λ -исчисление, рекурсивные функции и др. Для всех таких формальных систем доказана их эквивалентность МТ.
- ◇ МТ невозможно реализовать на *конечной* машине: МТ с лентой конечных размеров не обеспечивает реализации всех алгоритмов.
- ◇ **Тезис Тьюринга – Черча** (*основная гипотеза теории алгоритмов*). Для любой интуитивно вычислимой функции существует вычисляющая её значения МТ.

Критика модели вычислений Тьюринга

- ◆ Медленная (неускоряемая)
 - ◆ *частые копирования данных*: у нормальных МТ каждое неэлементарное действие выполняется над крайними правыми словами ленты
 - ◆ отказ от нормальных вычислений приведет к постоянному поиску данных и усложнит алгоритм
 - ◆ число состояний МТ часто зависит от числа символов в алфавите МТ

Введение в язык программирования Си

Схема простейшего компьютера



Язык программирования Си

- ◆ Си разрабатывался как язык для реализации первой в мире универсальной операционной системы UNIX
- ◆ 1973 – первая версия Си
- ◆ 1978 – выход книги Б. Кернигана и Д. Ритчи «Язык программирования Си» (K&R C). Русский перевод вышел в 1985 году.
- ◆ 1989 – первый стандарт ANSI C (C89)
- ◆ 1999 – стандарт C99
- ◆ 2011 – стандарт C11 (ранее назывался C1X)

Введение в язык программирования Си

Характеристики языка Си

- ◆ Императивный язык
- ◆ Удобный синтаксис
- ◆ Позволяет естественно оперировать «машинными» понятиями
- ◆ Переносимость на уровне исходного кода
 - ◆ Конфигурируемость
- ◆ Хорошие системные библиотеки
- ◆ Хорошие оптимизирующие компиляторы

Первая программа на Си

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    printf ("Hello, world\n");
```

```
    return 0;
```

```
}
```

Программа:

объявления переменных или функций

определения функций

Первая программа на Си

```
#include <stdio.h>

int main (void)
{
    printf ("Hello, world\n");
    return 0;
}
```

Директивы препроцессора

Системные библиотеки

Строковые константы

Управляющие последовательности