

Московский государственный университет им. М. В. Ломоносова
Факультет вычислительной математики и кибернетики

Алгоритмы и алгоритмические языки

Лекция 4

18 сентября 2019 г.

Пусть задано входное слово $\sigma_0 \in (V \cup V')^*$ и набор подстановок p_1, p_2, \dots, p_n .

1. Положить $i = 0$.
2. Положить $j = 1$.
3. Если подстановка p_j применима к слову σ_i , перейти к шагу 5.
4. Положить $j = j + 1$. Если $j \leq n$, то перейти к шагу 3, иначе остановиться.
5. Применить подстановку p_j к слову σ_i и построить слово σ_{i+1} . Если p_j — терминальная подстановка, то остановиться. Иначе положить $i = i + 1$ и перейти к шагу 2.

Говорят, что НАМ применим к слову σ_0 , если в результате выполнения описанной процедуры интерпретации произойдёт остановка.

Нормальные алгоритмы Маркова. Примеры

Шифр Юлия Цезаря: i -я буква латинского алфавита шифруется $(i + c) \bmod 26$ -й буквой, где i — номер буквы (начиная с нуля), c — некоторая константа.

НАМ: $*a \rightarrow d*$, $*b \rightarrow e*$, $*c \rightarrow f*$, \dots , $*y \rightarrow b*$, $*z \rightarrow c*$, $* \mapsto ,$ $\rightarrow *$.

- Маркер устанавливается в начало слова с помощью подстановки с пустой левой частью ($\rightarrow *$)
- Шифрование выполняется одной из 26 подстановок вида $*a_i \rightarrow a_{(i+3) \bmod 26}*$, где $0 \leq i < 26$, $a + i \in A_{26} = \{a, b, c, \dots, z\}$
- Последняя подстановка удаляет маркер из зашифрованного слова ($* \mapsto$)

Сложение чисел в единичной системе счисления: $V = \{+, |\}$,
 $V' = \{\}$.

Лобовая программа: $|+ \rightarrow +|$, $+| \rightarrow |$, $| \mapsto |$.

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.

Дома. Постройте НАМ, осуществляющий композицию двух НАМ F и G , вычисляющий по слову w слово $G(F(w))$. Считайте, что оба НАМ работают над одним алфавитом A .

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.

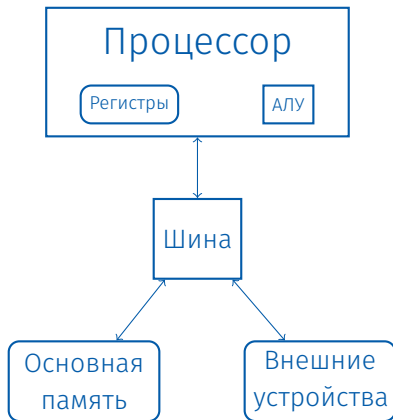
МТ невозможно реализовать на конечной машине: МТ с лентой конечных размеров не обеспечивает реализации всех алгоритмов.

Тезис Тьюринга-Чёрча (основная гипотеза теории алгоритмов). Для любой интуитивно вычислимой функции существует вычисляющая её значения МТ.

Поиск данных осуществляется за линейное время: нет «адресов», которые позволили бы попасть в нужное место ленты за константное время.

Нельзя «скопировать» символ программой константного размера: размножение состояний приводит к росту программы в зависимости от размера входного алфавита.

Схема простейшего компьютера



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)
`_Thread_local`, `_Generic`, `_Align*`, `_Noreturn...`

2018 стандарт C18 (только исправления ошибок в C11)

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

```
#include <stdio.h>

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

Программа:

- объявления переменных или функций
- определения функций

```
#include <stdio.h>

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

Директивы препроцессора

Системные библиотеки

Строковые константы

Управляющие последовательности