Курс «Алгоритмы и алгоритмические языки»
1 семестр 2017/2018

Лекция 3

Машина Тьюринга (МТ)

Проблема останова. Существует ли алгоритм, определяющий, произойдет ли когда-либо останов МТ $\boldsymbol{T}$ на входных данных w?

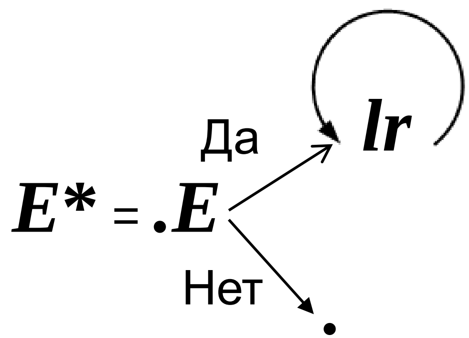
(другая формулировка) Остановится ли УМТ, моделирующая МТ $\boldsymbol{T}$ на входных данных w?

Утверждение. Проблема останова алгоритмически неразрешима.

Машина Тьюринга (МТ)
Проблема останова.

Пусть существует машина $\boldsymbol{D}$, решающая проблему останова для всех МТ $\boldsymbol{T}$ и входных данных w. Построим машину $\boldsymbol{E}$, которая по данной МТ $\boldsymbol{T}$ запускает машину $\boldsymbol{D}$ для МТ $\boldsymbol{T}$ и записи (описания) $\boldsymbol{T}$ на ленте.

Машина $\boldsymbol{E^*}$:



Останавливается ли машина $\boldsymbol{E^*}$, если ее применить к описанию самой себя?

Машина Тьюринга (МТ)
Проблема самоприменимости

- Рассмотрим МТ $\boldsymbol{T}$ для алфавита $\boldsymbol{A_p}$. МТ $\boldsymbol{T}$ называется самоприменимой, если она останавливается, когда в качестве начальных данных используется описание $\boldsymbol{T}$ – слово над алфавитом $\boldsymbol{S \cup Q}$.

  Существует ли МТ, которая по описанию МТ распознает, самоприменима ли она?
- Алгоритмическая неразрешимость проблемы самоприменимости следует из свойств МТ $\boldsymbol{E}$ и $\boldsymbol{E^*}$ с предыдущего слайда.

Нормальные алгоритмы Маркова

Определение нормального алгоритма Маркова (НАМ)

V – алфавит основных символов

V' – алфавит маркеров

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

Подстановка: $\sigma \to \sigma'$ 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 «$\to$» (***стрелка***) отделяет левую часть подстановки от правой и «**.**» (***точка***) отмечает терминальную подстановку

Нормальные алгоритмы Маркова

Определение. ***Нормальный алгоритм Маркова*** (***НАМ***) задается конечной последовательностью подстановок $\{\, p_1,\, p_2,\, \ldots\, p_n \}$.

При этом:

(1) если применимо несколько подстановок, применяется подстановка, которая встречается в описании алгоритма раньше других;

(2) если подстановка применима к нескольким подсловам обрабатываемого слова, выбирается самое левое подслово;

(3) после применения терминальной подстановки алгоритм завершается;

(4) если ни одна из подстановок неприменима, алгоритм завершается.

Нормальные алгоритмы Маркова

Пример НАМ. Шифр Юлия Цезаря

$V = \{a, b, c, \ldots, z\}$, $V' = \{*\}$.

j-ая буква латинского алфавита шифруется $j+h \pmod{26}$-ой буквой того же алфавита.

Например, для $h = 3$ подстановки имеют вид

(маркер $*$ помечает текущую шифруемую букву, цифра в скобках – номер правила подстановки):

$(1)\ {*}\boldsymbol{A} \to \boldsymbol{D}{*}$, $(2)\ {*}\boldsymbol{B} \to \boldsymbol{E}{*}$, $(3)\ {*}\boldsymbol{C} \to \boldsymbol{F}{*}$, $\ldots$,

$(23)\ {*}\boldsymbol{W} \to \boldsymbol{Z}{*}$, $(24)\ {*}\boldsymbol{X} \to \boldsymbol{A}{*}$, $(25)\ {*}\boldsymbol{Y} \to \boldsymbol{B}{*}$,

$(26)\ {*}\boldsymbol{Z} \to \boldsymbol{C}{*}$, $(27)\ {*} \to.$, $(28)\ \to *$

Применим построенный НАМ к слову $\boldsymbol{CAESAR}$:

$\boldsymbol{CAESAR}\ (28)\!\to {*}\boldsymbol{CAESAR}\ (3)\!\to \boldsymbol{F}{*}\boldsymbol{AESAR}\ (1)\!\to \boldsymbol{FD}{*}\boldsymbol{ESAR}\ (5)\!\to$
$\boldsymbol{FDH}{*}\boldsymbol{SAR}\ (13)\!\to \boldsymbol{FDHV}{*}\boldsymbol{AR}\ (1)\!\to \boldsymbol{FDHVD}{*}\boldsymbol{R}\ (12)\!\to$
$\boldsymbol{FDHVDU}{*}\ (27)\!\to \boldsymbol{FDHVDU}$

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 $j \le n$, 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 . (точка).

Иногда вместо $\rightarrow.$ используется метасимвол $\longmapsto$

Нормальные алгоритмы Маркова

Пример. Сложение чисел в единичной системе счисления:

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

Правила подстановок: $\{\ \mid + \rightarrow + \mid;\ \ + \mid \rightarrow \mid;\ \ \mid \rightarrow . \mid\ \}$

Пример применения алгоритма:

$$«|\,|\,|\,|+|\,|+|\,|\,|» = «|\,|\,|\,|+|\,|+|\,|\,|» \overset{(1)}{\Rightarrow} «|\,|\,|+|\,|\,|+|\,|\,|»$$

$$\overset{(1)}{\Rightarrow} «|\,|+|\,|\,|\,|+|\,|\,|» \overset{(1)}{\Rightarrow} «|+|\,|\,|\,|\,|+|\,|\,|»$$

$$\overset{(1)}{\Rightarrow} «+|\,|\,|\,|\,|\,|+|\,|\,|» \overset{(1)}{\Rightarrow} «+|\,|\,|\,|\,|+|\,|\,|\,|» \overset{(1)}{\Rightarrow} \ldots$$

$$\overset{(1)}{\Rightarrow} «++|\,|\,|\,|\,|\,|\,|\,|\,|» \overset{(2)}{\Rightarrow} «+|\,|\,|\,|\,|\,|\,|\,|\,|» \overset{(2)}{\Rightarrow} «|\,|\,|\,|\,|\,|\,|\,|\,|» \overset{(3)}{\Rightarrow}$$

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.

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.

Заключительные замечания

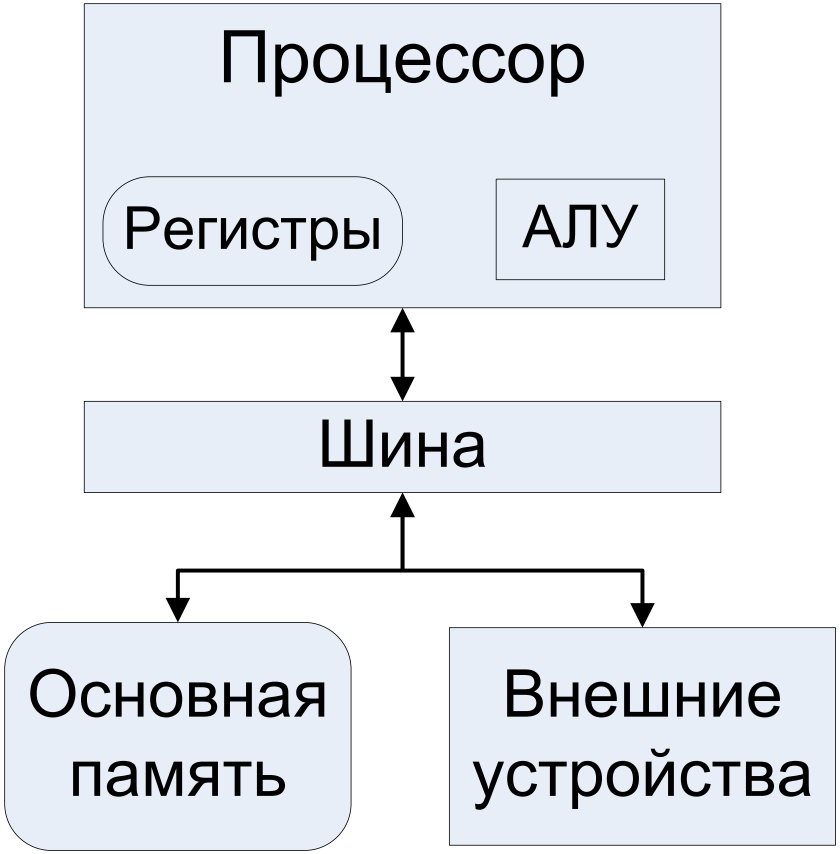
- Можно доказать эквивалентность двух формальных систем Тьюринга и Маркова конструктивным путем: построить универсальную МТ, которая могла бы интерпретировать любой НАМ и, наоборот, построить универсальный НАМ, который интерпретирует любую МТ.
- Существуют и другие формальные описания алгоритмов: машина 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

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)