Курс «Алгоритмы и алгоритмические языки»
1 семестр 2018/2019

Лекция 3

Диаграммы Тьюринга (ДТ)
Моделирование МТ

Определение. МТ $\boldsymbol{M}$ моделирует МТ $\boldsymbol{M'}$, если выполнены следующие условия:

(1) Данная начальная конфигурация вызывает машинный останов МТ $\boldsymbol{M}$ после конечного числа шагов тогда и только тогда, когда указанная начальная конфигурация вызывает машинный останов МТ $\boldsymbol{M}'$ после конечного числа шагов

(2) Данная начальная конфигурация вызывает переход за край ленты у МТ $\boldsymbol{M}$ после конечного числа шагов тогда и только тогда, когда указанная начальная конфигурация вызывает переход за край ленты у МТ $\boldsymbol{M}'$ после конечного числа шагов

(3) Последовательность текущих конфигураций МТ $\boldsymbol{M}'$ для данной начальной конфигурации является подпоследовательностью последовательности текущих конфигураций МТ $\boldsymbol{M}$ для той же начальной конфигурации

Что делать с алфавитами МТ *M* и *M'* ?

Диаграммы Тьюринга (ДТ)

Универсальная машина Тьюринга

Определение. Универсальной машиной Тьюринга (УМТ) для алфавита A_p называется такая МТ $\boldsymbol{U}$, на которой может быть промоделирована любая МТ над алфавитом A_p.

Замечание. На самом деле можно эффективно построить УМТ, моделирующую любую МТ над любым алфавитом. Для этого фиксируется некоторый алфавит (например $A_2 = \{0,1\}$) и добавляется кодирование и декодирование.

Идея УМТ. На ленту УМТ записывается программа моделируемой МТ (таблица) и исходные данные моделируемой МТ. УМТ по состоянию и текущему символу МТ находит на своей ленте команду моделируемой МТ, выясняет, какое действие нужно выполнить, и выполняет его.

Диаграммы Тьюринга (ДТ)

Универсальная машина Тьюринга

Этапы построения УМТ.

(1) **Как представить программу моделируемой МТ на ленте УМТ?**

Рабочий алфавит A' УМТ является расширением алфавита A_p.

$$A' = A_p \cup \{b_1, b_2, \ldots, b_p\} \cup \{b_0\} \cup \{r, l, h, +, -, \mathrm{O}, \mathrm{c}, \S, *\}$$

Программа: $cw_0cw_1\ldots cw_s\S$, где w_i – слово-программа для q_i.

Правило $q_ia_j \rightarrow v_{ij}q_k$, слово-правило: $\begin{cases} b_jv_{ij}+^{k-i}, & \text{если } k > i \\ b_jv_{ij}O, & \text{если } k = i \\ b_jv_{ij}-^{i-k}, & \text{если } k < i \end{cases}$

(2) **Как выглядит лента в исходном состоянии УМТ?**

$$\mathbf{[}\,{*}w_0cw_1\ldots cw_s\S\underset{\mathbf{q}}{\mathbf{w}}\Lambda\Lambda\,.\,.\,.$$

w – исходные данные моделируемой МТ. Звездочка отмечает q_0.

Диаграммы Тьюринга (ДТ)

Универсальная машина Тьюринга

Этапы построения УМТ.

(3) **Как происходит интерпретация моделируемой МТ?**

(а) УМТ “запоминает” (размножением состояний) обозреваемый в ячейке символ a_{j} из A_p и заменяет его на b_{j}.

(б) УМТ ищет слово программы w_i, описывающее текущее состояние моделируемой МТ (отмечено звездочкой).

(в) УМТ ищет символ b_j, соответствующий “запомненному” на шаге а) символу a_j, и сдвигается вправо через действие v_{ij} до описания сдвига на следующее состояние моделируемой МТ.

$$[\,cw_0cw_1\ldots\overbrace{*b_0v_{i0}++\ldots \underset{\mathbf{q}}{b_j v_{ij}}-\ldots b_pv_{ip}O}^{w_{\mathrm{i}}}\ldots cw_s\S\,\mathbf{a_{t1}a_{t2}}\overbrace{\ldots\mathbf{b_j}}^{a_{\mathrm{j}}}\ldots\mathbf{a_{tw}}\Lambda\Lambda\,.\;.\;.$$

Диаграммы Тьюринга (ДТ)

Универсальная машина Тьюринга

Этапы построения УМТ.

(3) **Как происходит интерпретация моделируемой МТ?**

(г) УМТ передвигает символ обозначения текущего состояния моделируемой МТ (звездочку) по описанию сдвига на ее символ c и возвращается на описание действия v_{ij}.

(д) УМТ ищет записанный на шаге а) символ b_{j} из данных моделируемой МТ (после символа §) и выполняет считанное действие (запись или сдвиг).

Замечание. Если при сдвиге УГ попала на символ §, отделяющий программу моделируемой МТ от данных, это означает, что моделируемая МТ зашла за левый край ленты.

(е) Снова можно выполнять шаг а).

Диаграммы Тьюринга (ДТ)

Универсальная машина Тьюринга

Этапы построения УМТ.

(4) **Как происходит останов УМТ?**

Если на шаге 3 б) при поиске слова w_i был найден символ § (справа после звездочки), моделируемая МТ находится в состоянии останова.

(а) УМТ ищет символ b_j из данных моделируемой МТ (после символа §) и записывает запомненный символ a_j.

(б) После этого УГ УМТ указывает на ячейку, на которой должна остановиться УГ моделируемой МТ.

$$[cw_0cw_1\ldots{*}w_s\S\mathbf{MT(w)}\Lambda\Lambda\ldots$$

$$\mathbf{q}$$

Машина Тьюринга (МТ)

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

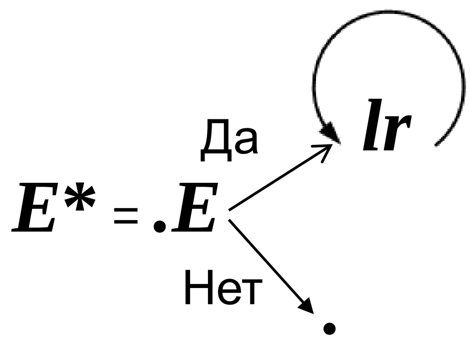
(другая формулировка) Остановится ли УМТ, моделирующая МТ $\boldsymbol{T}$ на входных данных w?

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 $\boldsymbol{S} \cup \boldsymbol{Q}$.

  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 $\{ p_1, p_2, \dots p_n \}$.

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):

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

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

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

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

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

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:

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

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

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