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

Алгоритмы и алгоритмические языки
Лекция 3

11 сентября 2019 г.

Моделирование МТ

МТ M *моделирует* МТ M', если:

1. Данная начальная конфигурация вызывает машинный останов/переход за край ленты МТ M после конечного числа шагов тогда и только тогда, когда указанная начальная конфигурация вызывает машинный останов/переход за край ленты МТ M' после конечного числа шагов.
2. Для последовательности (c'_n) текущих конфигураций МТ M' для данной начальной конфигурации можно указать *моделирующую* подпоследовательность (c_n) последовательности текущих конфигураций МТ M для той же начальной конфигурации:
   для каждой конфигурации c'_i машины M' её лента будет «частью» ленты конфигурации c_i машины M, УГ машины M будет находиться на ячейке, соответствующей положению рабочей ячейки машины M', и по конфигурации c_i можно указать состояние машины M' в конфигурации c'_i.

Универсальная машина Тьюринга

Универсальной машиной Тьюринга (УМТ) для алфавита A[1] называется такая машина $\boldsymbol{U}$, на которой может быть промоделирована любая МТ над алфавитом A.

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.

[1] 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.

УМТ. Представление программы

Моделируемая машина T:

- рабочий алфавит A_p,
- состояния q_0, q_1, ..., q_s,
- правила $q_i a_j \to v_{ij} q_k$, где $i, k = 0, \ldots, n$; $j = 1, \ldots, p$; $v_{ij} \in \{a_1, a_2, \ldots, a_p, l, r, h\}$.

Универсальная машина:

- алфавит $B_p = \{b_1, b_2, \ldots, b_p\}$,
- дополнительные символы $\{l, r, h, +, -, O\}$.

Для правила $q_i a_j \to v_{ij} q_k$ запись выглядит как

$$\begin{cases} b_j v_{ij} +^{k-i}, \text{если } k > i; \\ b_j v_{ij} O, \text{если } k = i; \\ b_j v_{ij} -^{i-k}, \text{если } k < i. \end{cases}$$

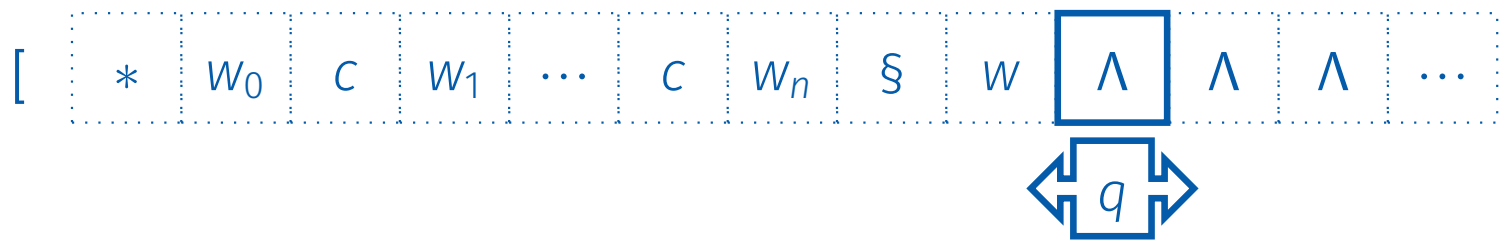
$+^{k-i}$ означает символ $+$, повторённый $k - i$ раз.

УМТ. Представление программы

Слово-программа: $cw_0cw_1 \dots cw_n\S$,
где w_i — слово с записью подряд всех правил состояния q_i.

- Слова правил разных состояний отделяются друг от друга вспомогательным маркером c.
- Вся программа заканчивается маркером §.

Лента в начальном состоянии:



w — исходные данные моделируемой МТ;
$*$ — маркер начального состояния.

УМТ. Интерпретация моделируемой МТ

1. Поиск правила для выполнения
2. Изменение текущего состояния моделируемой МТ
3. Выполнение действия моделируемой МТ
4. Переход на выполнение нового такта

УМТ. Интерпретация моделируемой МТ

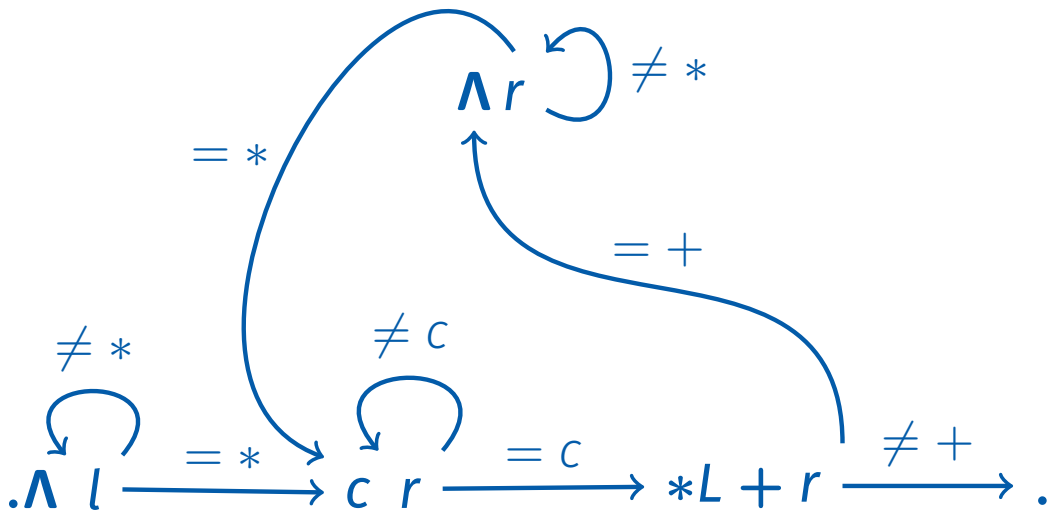
1. Поиск правила для выполнения
   - 1.1 “Запоминаем” обозреваемый символ a_j размножением состояний
   - 1.2 Заменяем символ a_j на его зеркальную пару b_j
   - 1.3 Ищем слово w_i, содержащее запись правила
   - 1.4 Ищем запись правила для символа a_j

$$cw_0cw_1\ldots\overbrace{*b_0v_{i0}++\ldots\underbrace{b_j}_{\uparrow}v_{ij}---\ldots}^{w_i}cw_s\S a_{t1}a_{t2}\ldots\overbrace{b_j}^{a_j}\ldots a_{tw}\Lambda\Lambda\ldots$$

2. Изменение текущего состояния моделируемой МТ
3. Выполнение действия моделируемой МТ
4. Переход на выполнение нового такта

УМТ. Интерпретация моделируемой МТ

1. Поиск правила для выполнения
2. Изменение текущего состояния моделируемой МТ
   - 2.1 Сдвигаемся на один символ вправо, пропуская v_{ij}
   - 2.2 По описанию сдвига пропускаем соответствующее количество символов-маркеров c и ставим символ текущего состояния $*$
   - 2.3 Возвращаемся на символ описания v_{ij} действия



3. Выполнение действия моделируемой МТ
4. Переход на выполнение нового такта

УМТ. Интерпретация моделируемой МТ

1. Поиск правила для выполнения
2. Изменение текущего состояния моделируемой МТ
3. Выполнение действия моделируемой МТ
   - 3.1 Ищем ячейку ленты, на которой находится УГ моделируемой МТ
   - 3.2 Выполняем считанное действие (запись или сдвиг[2])
4. Переход на выполнение нового такта

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

УМТ. Интерпретация моделируемой МТ

1. Поиск правила для выполнения
2. Изменение текущего состояния моделируемой МТ
3. Выполнение действия моделируемой МТ
   - 3.1 Ищем ячейку ленты, на которой находится УГ моделируемой МТ
   - 3.2 Выполняем считанное действие (запись или сдвиг[2])
4. Переход на выполнение нового такта
   - 4.1 *Ничего делать не нужно! Ура!*

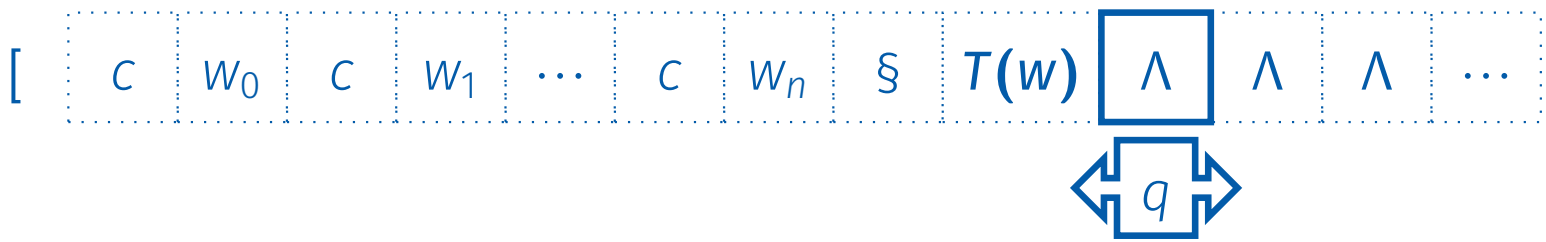
[2] Если при сдвиге УГ попала на символ §, отделяющий программу моделируемой МТ от данных, это означает, что. моделируемая МТ зашла за левый край ленты.

УМТ. Останов моделируемой МТ

Если при сдвиге маркера текущего состояния (шаг 2.2) происходит переход на символ §, то следующим состоянием будет являться состояние останова.

В таком случае УМТ нужно выполнить действие моделируемой машины, а потом остановиться.

Лента в состоянии останова:



Проблема останова

Существует ли алгоритм, определяющий, произойдет ли когда-либо останов машины T, запущенной на входных данных w? Или иначе, остановится ли универсальная машина Тьюринга, моделирующая МТ T на входных данных w?

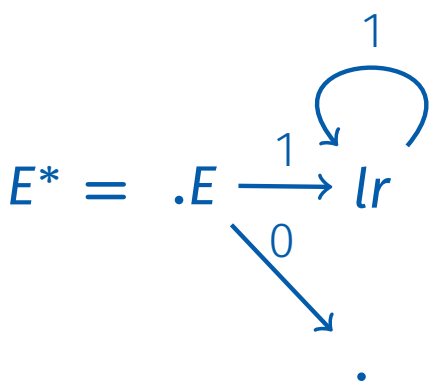
Утверждение. Проблема останова алгоритмически неразрешима.

Проблема останова. Доказательство

Пусть существует машина D, решающая проблему останова для всех МТ T и входных данных w.

Построим машину E, которая по данной МТ T запускает машину D для МТ T и записи (описания) T на ленте.

Машина E^*:



Останавливается ли машина E^*, будучи применённой к описанию самой себя (т.е. описанию машины E^*)?

Проблема самоприменимости

Машина Тьюринга T называется *самоприменимой*, если она останавливается, когда в качестве входного слова для неё используется описание[3] самой машины T.

Проблемой самоприменимости является вопрос о существовании алгоритма, определяющего самоприменимость любой заданной машины T.

Алгоритмическая неразрешимость проблемы самоприменимости может быть доказана тем же способом, что и неразрешимость проблемы останова: такой машиной является машина E с предыдущего слайда.

[3] Как и ранее, будем считать, что с помощью кодирования описание задано во входном алфавите нашей машины.

Нормальные алгоритмы Маркова. Обозначения

V — алфавит основных символов,
V' — алфавит символов-маркеров.

$\sigma, \sigma' \in V \cup V'$.

Подстановка $\sigma \to \sigma'$ переводит слово $\tau = \alpha\sigma\beta$ в слово $\tau' = \alpha\sigma'\beta$, где $\tau, \tau', \alpha, \beta \in V \cup V'$.

Как слова α и β, так и слова σ и σ' могут быть пустыми.

Метасимвол $\to$ отделяет левую часть подстановки от правой.

Нормальные алгоритмы Маркова. Определение

Нормальный алгоритм Маркова (НАМ) задаётся конечной последовательностью подстановок $p_1, p_2, \ldots, p_n$.

«Такт» работы алгоритма состоит в поиске подстановки, применимой к текущему обрабатываемому слову:

- поиск применимой подстановки ведётся, начиная с первой подстановки в последовательности;
- если ни одна подстановка не оказалась применимой, алгоритм завершается;
- первая найденная применимая подстановка применяется: заменяется самое левое вхождение слова из левой части подстановки;
- подстановка может быть помечена как *терминальная*, тогда после её применения алгоритм завершается.

Терминальная подстановка обозначается как $\to .$ или $\mapsto$

Нормальные алгоритмы Маркова. Процедура интерпретации

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

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 $j \leqslant n$, 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.