

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

**Лекция 3**

# Диаграммы Тьюринга (ДТ)

## Моделирование МТ

**Определение.** МТ  $M$  моделирует МТ  $M'$ , если выполнены следующие условия:

- (1) Данная начальная конфигурация вызывает машинный останов МТ  $M$  после конечного числа шагов тогда и только тогда, когда указанная начальная конфигурация вызывает машинный останов МТ  $M'$  после конечного числа шагов
- (2) Данная начальная конфигурация вызывает переход за край ленты у МТ  $M$  после конечного числа шагов тогда и только тогда, когда указанная начальная конфигурация вызывает переход за край ленты у МТ  $M'$  после конечного числа шагов
- (3) Последовательность текущих конфигураций МТ  $M'$  для данной начальной конфигурации является подпоследовательностью последовательности текущих конфигураций МТ  $M$  для той же начальной конфигурации

## **Диаграммы Тьюринга (ДТ)**

### **Универсальная машина Тьюринга**

**Определение.** Универсальной машиной Тьюринга (УМТ) для алфавита  $A_p$  называется такая МТ  $U$ , 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  $A_p$ .

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

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

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

(2) Как выглядит лента в исходном состоянии УМТ?

$$[*w_0cw_1\dots cw_s\S w \Lambda \Lambda \dots]$$

$\uparrow$   
 $q$

$w$  – исходные данные моделируемой МТ. Звездочка отмечает  $q_0$ .

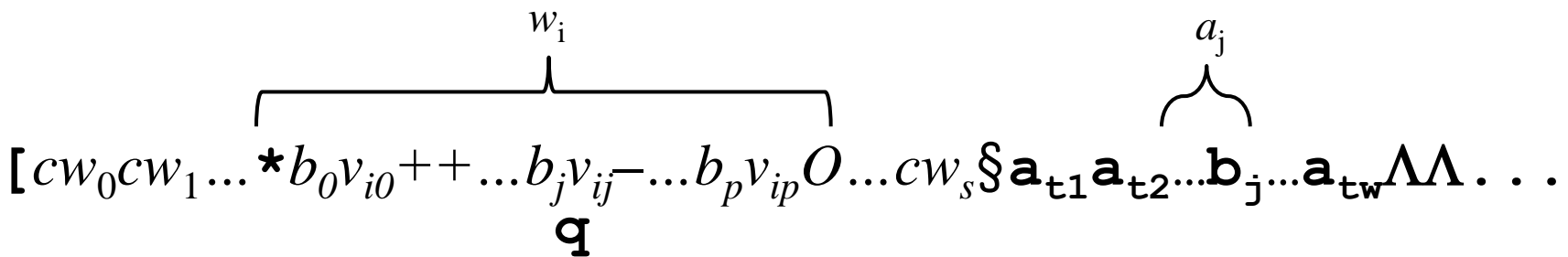
# Диаграммы Тьюринга (ДТ)

## 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.



## **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  $\S$ ) и выполняет считанное действие (запись или сдвиг).

**Замечание.** Если при сдвиге УГ попала на символ  $\S$ , 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  $\S$  (справа после звездочки), моделируемая МТ находится в состоянии останова.

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

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

$[cw_0cw_1\dots *w_s\S_{MT}(\mathbf{w})\Lambda\Lambda\dots$   
 $\qquad\qquad\qquad\quad\uparrow$

## ***Машина Тьюринга (МТ)***

***Проблема останова.*** Существует ли алгоритм, определяющий, произойдет ли когда-либо останов МТ ***T*** на входных данных  $w$ ?  
(для *любых* машин  $T$  и *любых* входных данных)

(другая формулировка) Остановится ли УМТ, моделирующая МТ ***T*** на входных данных  $w$ ?

***Утверждение.*** Проблема останова алгоритмически неразрешима.

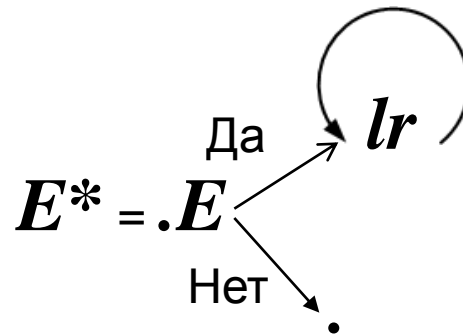


## Машина Тьюринга (МТ)

### Проблема останова.

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  $E^*$ , если ее применить к описанию самой себя?

## **Машина Тьюринга (МТ)**

### **Проблема самоприменимости**

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

Существует ли МТ, которая по описанию МТ распознает, самоприменима ли она?

- ◇ Алгоритмическая неразрешимость проблемы самоприменимости следует из свойств МТ  $E$  и  $E^*$  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:**  $\sigma \rightarrow \sigma'$  переводит

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

подслова  $\alpha$  и  $\beta$  могут быть пустыми ( $\varepsilon$ )

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**

**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

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

$j$ -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):

$$\begin{aligned} (1) \ *A \rightarrow D^*, \quad (2) \ *B \rightarrow E^*, \quad (3) \ *C \rightarrow F^*, \quad \dots, \\ (23) \ *W \rightarrow Z^*, \quad (24) \ *X \rightarrow A^*, \quad (25) \ *Y \rightarrow B^*, \\ (26) \ *Z \rightarrow C^*, \quad (27) \ * \rightarrow . \ , \quad (28) \ \rightarrow * \end{aligned}$$

Применим построенный НАМ к слову **CAESAR**:

$$\begin{aligned} CAESAR \xrightarrow{(28)} *CAESAR \xrightarrow{(3)} F*AESAR \xrightarrow{(1)} FD*ESAR \xrightarrow{(5)} \\ FDH*SAR \xrightarrow{(13)} FDHV*AR \xrightarrow{(1)} FDHVD*R \xrightarrow{(12)} \\ FDHVDU* \xrightarrow{(27)} FDHVDU \end{aligned}$$