1 Недетерминированная машина Тьюринга

На прошлой лекции говорили о детерминированной машине Тьюринга (ДМТ) $(\Sigma, Q, q_0, Q_t, \delta)$, где Σ — конечный алфавит, Q — множество состояний, q_0 — начальное состояние, $Q_t \subseteq Q$ — конечные, $\delta : \Sigma \times Q \mapsto \Sigma \times Q \times \{\leftarrow, \bullet, \rightarrow\}$. Это была ДМТ, сделаем недетерминированную машину Тьюринга (НМТ). У НМТ все аналогично, но $\delta \subseteq \Sigma \times Q \times \Sigma \times Q \times \{\leftarrow, \bullet, \rightarrow\}$. Теперь

$$\begin{tikzcd}
 & \boxed{(q_1, c_1, \rightarrow)} \\
\boxed{(q, c)} \arrow[ur, bend left] \arrow[dr, bend right] & \\
 & \boxed{(q_2, c_2, \leftarrow)}
\end{tikzcd}$$

- Есть оракул, который умеет управлять машиной и она ждет его
- При попадании в такую ситуацию мы раздваиваемся на 2 вселенные и вычисляем параллельно. Если хотя бы в одной вселенной удалось, то перемещаемся в нее ...

Вопрос: Зачем? Вычислительная мощность — класс задач ($\mathcal{P}$). $\mathcal{P}$(ДМТ)$=\mathcal{P}$(НМТ). Докажем это. $\subseteq$ очевидно. $\supseteq$ возьмем универсальную машину Тьюринга (УМТ):

Программа

Текущее слово

Текущее состояние

Буфер

В буфере находятся конфигурации.

1. Итерация УМТ
2. ПУМТ пишет все результирующие конфигурации на 4 ленту. Что-то вроде очереди

2 Другие вычислительные устройства. Нормальный Алгорифм

Состоит из инструкций вида:

$$\begin{aligned} s_1 &\to t_1 \\ s_2 &\to t_2 \\ &\vdots \\ s_n &\Rightarrow t_n \end{aligned}$$

Последняя инструкция является завершающей. Рассмотрим пример. Вход: строка "$abcdef$".$a \to b, bbc \to x, ef \Rightarrow y$. Шаги: "$bbcdef$" $\to$ "$xdef$" $\Rightarrow$ "xdy".
Задача инкремента.

$$\begin{aligned} &0\lambda \Rightarrow 1\lambda \\ &1\lambda \Rightarrow 2\lambda \\ &\quad\vdots \\ &9\lambda \to \star 0\lambda \\ &0\star \Rightarrow 1 \\ &1\star \Rightarrow 2 \\ &\quad\vdots \\ &9\star \to \star 0 \\ &\lambda\star \Rightarrow \lambda 1 \end{aligned}$$

Пример с инкрементом.

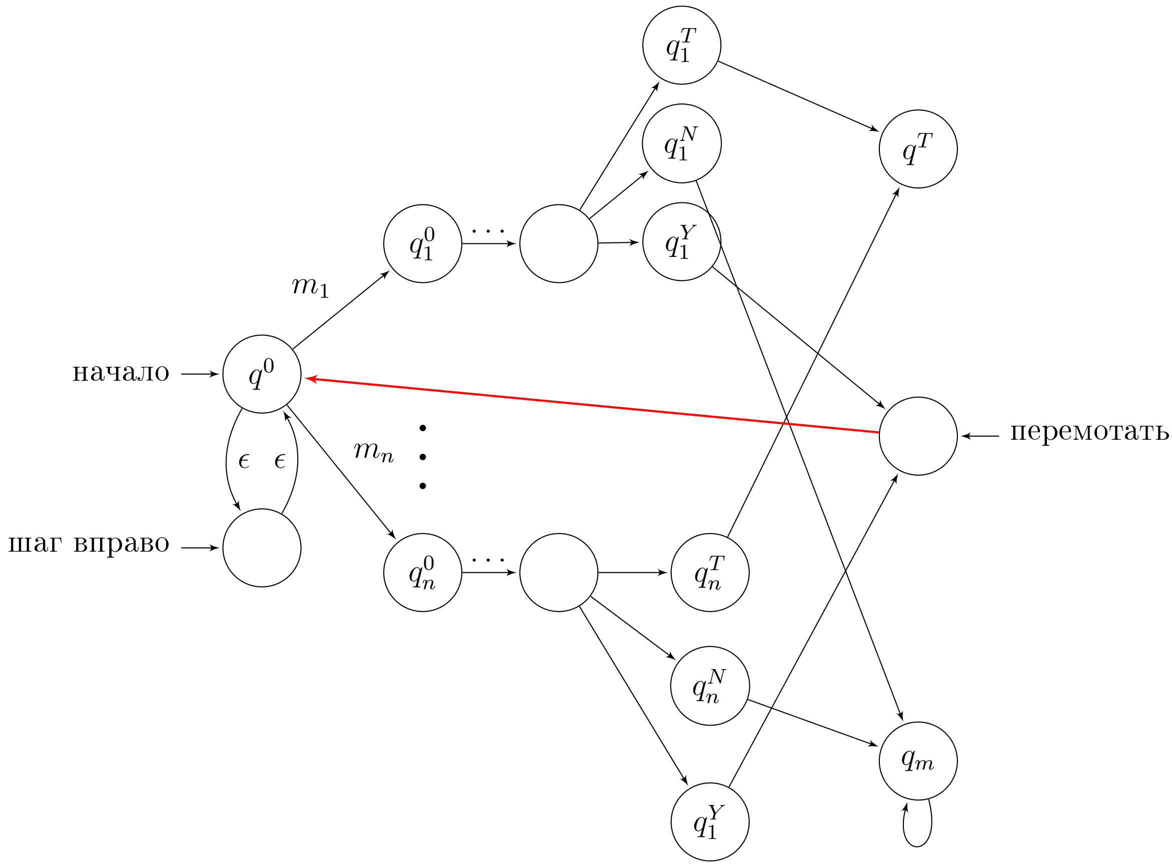
$$\begin{aligned} 12 &\Rightarrow 13 \\ 19 &\to 1 \star 0 \Rightarrow 20 \\ 199 &\to 19 \star 0 \to 1 \star 00 \to 200 \\ 99 &\to 9 \star 0 \to \star 00 \Rightarrow 100 \end{aligned}$$

Утверждение 1.

$$\mathcal{P}(NA) = \mathcal{P}(MT)$$

Доказательство. В сторону $\subseteq$. $p_1 : s_1 \to t_1, p_2 :\to t_2, \dots p_n$. Построим серию МТ m_i (конкретная подпрограмма), где каждая, во-первых, начиная с текущей позиции проверяет, что префикс $= s_i$, если нет, то переходит в в q_i^N. Во-вторых, если удачно, то производит замену s_i на t_i и переходит в q_i^T, если p_i —

завершающее, иначе q_i^y. Начало q_i^0. МТ $= (Q, \Sigma, q_0, Q_t, \delta), Q = Q^{m_1} \cup Q^{m_2} \cup \dots \cup Q^{'}$. $Q^{m_1} = \{q_1^0, q_1^N, q_1^T, q_1^Y, \dots\}$.



В сторону $\supseteq$. МТ: $\{(q_1, s_1, q_1^{'}, s_1^{'}, m_1) = \delta_1, \dots, \delta_n\}$. Построим алгорифм.

1. $\lambda x \to \dashv [q_0]x, x \in \Sigma$. ($\dashv$ — левый упор. $[q_0]$ — подразумевается один символ. В записи указали и положение, и состояние). Перед первым символом состояние q_0.

2. 2.1. $\forall \delta_i$, где $m_i = \bullet$ будет $[q_i]s_i \to [q_i]^{'} s_i^{'}$

   2.2. $\forall \delta_i, m_i = \to$ будет $[q_i]s_i \to s_i^{'}[q_i^{'}]$

   2.3. $\forall i : m_i = \leftarrow$ будет $x[q_i]s_i \to [q_i^{'}]xs_i^{'} \forall x \in \Sigma^{'}$

3. $\forall q \in Q_t, [q] \Rightarrow \lambda$

$\square$

3 Машина Минского

Машина умеет:

1. Увеличивать число на 1
2. Уменьшать число на 1 и сравнивать его с 0
3. завершаться

$(Q, R_1, \ldots, R_n)$, где Q — состояния, $R_1, \ldots, R_n$ — числовые регистры.

(i) $q_i : R_j \leftarrow R_j + 1$, goto q_k

(ii) $q_i : R_j \leftarrow R_j - 1 ? q_y : q_n$

(iii) $q_i :$ stop

Задача сложения

$$\begin{aligned} &R_1 := R_1 + R_2 \\ &q_0 : R_2 \leftarrow R_2 - 1 ? q_1 : q_2 \\ &q_1 : R_1 \leftarrow R_1 + 1, q_0 \\ &q_2 : R_1 \leftarrow R_1 + 1, q_3 \\ &q_3 : STOP \end{aligned}$$