
Реализация ветвления на машине Тьюринга.

Есть три машины Тьюринга:

- **T_P** — вычисляет предикат (условие) $P(x)P(x)$.
Результат: $P(x)=0 P(x)=0$ (ложь) или $P(x)=1 P(x)=1$ (истина).
- **T₁** — вычисляет функцию $f_1(x)f_1(x)$.
- **T₂** — вычисляет функцию $f_2(x)f_2(x)$.

Все три машины работают в одном внешнем алфавите.

2. Переобозначение состояний

Состояния машин переименовываются, чтобы в объединённой машине они не пересекались. Например:

- Состояния TPTP получают верхний индекс pp: q_{1p}, q_{2p}, ..., q_{1p}, q_{2p}, ...
 - Состояния T1T1 — верхний индекс 11: q₁₁, q₂₁, ..., q₁₁, q₂₁, ...
 - Состояния T2T2 — верхний индекс 22: q₁₂, q₂₂, ..., q₁₂, q₂₂, ...
-

3. Схема ветвления

Мы хотим создать новую машину **T**, которая вычисляет:

$$f(x) = \begin{cases} f_1(x), & \text{если } P(x)=0 \\ f_2(x), & \text{если } P(x)=1 \end{cases}$$

Алгоритм:

1. Дублирование данных

На ленте исходные данные xx копируются, получается две копии рядом.

2. Вычисление $P(x)P(x)$

Машине начинает с начального состояния q_{1q1} (которое было q_{1q1} машины TPTP) и обрабатывает **левую копию** данных, оставляя правую копию нетронутой.

3. Очистка левой части

После того как TPTP завершит работу, результат предиката (0 или 1) известен внутри состояния машины.

Затем стираются все ячейки **левее правой копии** (включая левую копию и мусор от вычисления РР).

4. Переход к правой копии

Указатель ставится на первую непустую ячейку **правой копии** данных xx.

- Если $P(x)=0$, то переходим в **начальное состояние машины T1T1** (qi_1qi_1).
- Если $P(x)=1$, то переходим в **начальное состояние машины T2T2** (qi_2qi_2).

5. Вычисление f1f1 или f2f2

Машина выполняет соответствующую программу T1T1 или T2T2 на правой копии данных.

6. Общее завершение

Заключительные состояния qf_1qf_1 и qf_2qf_2 машин T1T1 и T2T2 отождествляются в одно общее **заключительное состояние qfqf** машины ТТ. На ленте — результат $f_1(x)f_1(x)$ или $f_2(x)f_2(x)$.

4. Зачем дублирование данных?

Если бы мы вычисляли $P(x)P(x)$ прямо на исходных данных xx, то после вычисления предиката исходные данные могли быть испорчены, и для вычисления f1f1 или f2f2 они были бы недоступны.

Дублирование позволяет сохранить **оригинальные данные** для второй фазы вычислений.

5. Итог

Таким образом, **ветвление** в теории машин Тьюринга реализуется через:

1. Дублирование входа.
2. Вычисление условия на одной копии.
3. Очистку рабочей области и выбор соответствующей машины для работы со второй копией.

Это аналог условного оператора `if P(x) then f2(x) else f1(x)` в обычных языках программирования.