

Дискретная математика

Машины Тьюринга

Константин Чухарев

Осень 2026

Определение

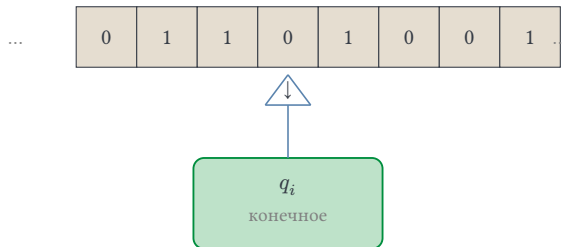
Машины удивляют меня с большой частотой.

— Алан Тьюринг

Определение МТ

Прежде чем строить модель, уточним, что значит «вычислить». Нужна модель проще компьютера, но не слабее по возможностям.

Лента:



Формальное определение

Определение 1 (Машина Тьюринга)

Семёрка $M = (Q, \Gamma, \Sigma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$:

- Q — конечное множество состояний.
- Γ — алфавит ленты, содержащий Σ и пробел.
- δ — частичная функция переходов: на паре (состояние, символ) она может быть не определена.

Машина состоит из ленты, головки и конечного управления.

Простота и сила

Машина Тьюринга — модель алгоритма из двух частей:

- бесконечная лента — память: ячейки с символами, головка читает и пишет.
- конечное управление — программа: правила «вижу символ a в состоянии q — пишу b , сдвигаюсь, перехожу в q' ».

Этой простоты хватает на всё: по тезису Чёрча–Тьюринга любая вычислимая функция реализуется машиной Тьюринга.

Один шаг

Определение 2 (Переход)

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}.$$

- Прочитать символ.
- Записать новый.
- Сдвинуть головку.
- Сменить состояние.

Символ и состояние определяют следующее действие.

Конфигурация записывается как uqv :

- u — содержимое ленты слева от головки.
- q — текущее состояние.
- v — содержимое справа вместе с читаемой ячейкой.

Исходы

Определение 3

- МТ **принимает** вход, если достигает принимающего состояния.
- МТ **отвергает**, если достигает отвергающего.
- Машина может **зациклиться**.

Остановка без перехода (δ не определена) — тоже отвергание.

Зацикливание — третий исход: не принять, не отвергнуть.

Распознаваемость и разрешимость

Определение 4 (Распознаваемость и разрешимость)

L **распознаваем** (рекурсивно перечислим), если есть МТ, принимающая w в точности тогда, когда $w \in L$.

На $w \notin L$ машина может отвергнуть или зациклиться.

L **разрешим** (*decidable*), если есть МТ, всегда останавливающаяся: принимает $w \in L$, отвергает $w \notin L$.

Разрешимый язык распознаваем.

Обратное — нет: пример распознаваемого, но неразрешимого языка — проблема остановки — появится в следующей лекции.

Строки, заканчивающиеся на 0

Пример (Семёрка в деле)

Язык слов над $\{0, 1\}$, заканчивающихся на 0.

$$Q = \{q_0, q_1, q_{\text{accept}}, q_{\text{reject}}\}, \Sigma = \{0, 1\}.$$

Состояния:

- q_0 — «последний символ 1».
- q_1 — «последний символ 0».

Переходы:

- $(q_0, 1) \rightarrow (q_0, 1, R)$.
- $(q_0, 0) \rightarrow (q_1, 0, R)$.
- $(q_1, 0) \rightarrow (q_1, 0, R)$.
- $(q_1, 1) \rightarrow (q_0, 1, R)$.

Остановка и трасса

Пример (Остановка)

На пробеле:

- из q_1 — принять.
- из q_0 — отвергнуть.

Пример (Трасса на 10)

$q_0 10 \rightarrow 1q_0 0 \rightarrow 10q_1$ (головка на пробеле) — принять.

Всегда останавливается — язык разрешим.

Конечная семёрка — кодируемый объект: её можно записать строкой.

Примеры

Инверсия битов

Пример

Заменить 0 на 1 и 1 на 0.

Одно состояние:

- $(q_0, 0) \rightarrow (q_0, 1, R)$.
- $(q_0, 1) \rightarrow (q_0, 0, R)$.

На пробеле — принять.

Эта машина вычисляет функцию, а не распознаёт язык.

$0^n 1^n$

Пример

Разрешим: многократно вычёркиваем 0 слева и 1 справа.

0011: q_0 читает 0, вычёркивает, идёт вправо. q_2 находит 1 справа, вычёркивает. Повтор.

$X011 \rightarrow X01X \rightarrow XX1X \rightarrow XXXX$ — принять.

МТ сравнивает количества нулей и единиц, вычёркивая их попарно.

Палиндромы

Пример

Алгоритм:

1. Сравнить первый и последний символ.
2. Стереть их.
3. Повторить проход для оставшейся части слова.

Чтобы запомнить символ, машина переходит в отдельное состояние для каждого символа. При несовпадении машина отвергает.

Стандартные приёмы:

- Запоминание через состояние.
- Движение к краю.
- Возврат.

МТ видит оба конца строки — ценой проходов.

Стандартные приёмы

Запоминание и маркеры

Определение 5 (Запоминание через состояние)

Чтобы запомнить «текущий символ — a », машина переходит в состояние «запомнила a » и несёт эту информацию, пока она не понадобится.

Состояние работает как переменная, принимающая конечное число значений.

Определение 6 (Вычёркивание и маркеры)

Обработанные символы заменяются маркером (X в примере с $0^n 1^n$).

Маркер отделяет «уже учтено» от «ещё нет» без дополнительной памяти.

Счёт на МТ — это расстановка и пересчёт маркеров.

Проход к краю и возврат

Определение 7 (Проход к краю и возврат)

Состояния-двигатели: «идти вправо, пока не пробел», затем шаг влево и переключение.

Проход по ленте — цикл, условие завершения — край строки.

Эти приёмы — прообраз процессора:

- Состояние играет роль счётчика команд.
- Маркер — флага.
- Проход — цикла.

Композиция машин

Определение 8 (Композиция машин)

Если M_1 переводит ленту из u в v , а M_2 — из v в w , состояние останова M_1 отождествляется с начальным M_2 .

Из простых блоков собираются большие программы:

- «найти край».
- «скопировать».
- «стереть».

Программа на ленте — те же символы, которые конечное управление интерпретирует как инструкции.

Варианты

Многоленточная МТ

Определение 9 (Многоленточная МТ)

k лент, k головок.

За один шаг каждая головка читает и пишет свой символ, каждая сдвигается независимо.

Удобнее программировать, но не сильнее.

Все ленты — на одну попеременно, позиции головок — маркеры.

Симуляция одного шага обходит всю уже записанную часть ленты, поэтому $T(n)$ шагов превращаются в $O(T(n)^2)$.

Недетерминированная МТ

Определение 10 (НМТ)

δ возвращает множество возможных следующих конфигураций.

Принимает, если существует хотя бы одна принимающая ветвь.

Выглядит как угадывание — на деле квантор существования.

Детерминированная МТ обходит дерево конфигураций в ширину, чтобы не застрять в бесконечной ветви.

Замедление — экспоненциальное в худшем случае.

Устойчивость модели

Теорема 1 (Устойчивость модели)

Многоленточные, недетерминированные, двухстековые, счётчиковые (Минского) и регистровые машины вычисляют тот же класс функций, что и одноленточная детерминированная МТ.

Набросок: многоленточная МТ переплетает ленты в одну и помечает позиции головок маркерами. Недетерминизм обходится поиском в ширину по дереву конфигураций.

Каждое расширение симулируется базовой машиной.

Плата — только время: квадратичное для лент, экспоненциальное для недетерминизма.

Тезис Чёрча–Тьюринга

Тезис

Тезис Чёрча–Тьюринга: класс функций, вычислимых на МТ, совпадает с классом интуитивно вычислимых. Это тезис, а не теорема: он фиксирует границу интуитивной вычислимости, его не доказывают.

Если задача неразрешима на МТ — она неразрешима нигде.

Три формализма 1936 года эквивалентны:

- МТ.
- Лямбда-исчисление.
- Рекурсивные функции.

Гёделя убедила именно механическая наглядность МТ.

Тезис принят: ни одна альтернативная модель не вышла за класс тьюринг-вычислимых.

Частичная вычислимость

Определение 11 (Вычислимая функция)

МТ **вычисляет** $f : \Sigma^* \rightarrow \Sigma^*$, если на входе w останавливается с $f(w)$ на ленте в точности тогда, когда $f(w)$ определено.

Функция может быть **частичной**: где не определена — машина не останавливается.

Пример

- Инверсия битов — всюду определённая (*total*): останавливается на каждом входе.
- МТ, закливающаяся на части входов, вычисляет частичную функцию.

Тезис Чёрча–Тьюринга — про частичные функции.

«Вычислима» и «всюду определена» — разные свойства.

Универсальная МТ

Теорема 2 (Универсальная МТ)

Существует МТ U , которая по кодировке произвольной МТ M и строке w симулирует работу M на w .

Программа лежит на ленте как данные: таблица переходов МТ кодируется строкой символов.

Одна универсальная машина исполняет любую программу.

Интерпретатор

Пример

Универсальная МТ — прообраз интерпретатора.

Программа читает программу:

- Java.
- .NET.
- WebAssembly.

Архитектура фон Неймана — программа и данные в общей памяти.

Что мы поняли?

- Машина Тьюринга — бесконечная лента и конечное управление. Этой простоты хватает на весь класс вычислимого.
- Разрешимость — всегда останавливаться. Распознаваемость допускает заикливание на $w \notin L$.
- Многоленточность и недетерминизм симулируются базовой машиной: тезис Чёрча–Тьюринга фиксирует единый класс вычислимого.
- Программа как данные: универсальная МТ симулирует любую машину — прообраз интерпретатора.

Вопросы для самопроверки

- Определите машину Тьюринга.
- Что такое конфигурация и как записывается?
- Чем распознаваемость отличается от разрешимости?
- Опишите МТ для $\{0^n 1^n\}$.
- Почему многоленточная МТ не сильнее одноленточной?
- Как детерминированная МТ симулирует недетерминированную?
- Что такое частичная и всюду определённая функция?
- Сформулируйте тезис Чёрча–Тьюринга.
- Что такое универсальная МТ?
- Почему «программа как данные» важна для компьютеров?