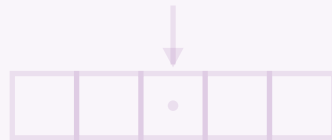


Лекция

6–7 апреля

Машина Тьюринга



Машина Тьюринга

Вычисление обычно выполняется записыванием определённых символов на бумаге.

— Алан Тьюринг

Чего не хватило автомату?

В первом семестре язык $a^n b^n$ поставил конечные автоматы на место.

Лемма о накачке и теорема Майхилла–Нерода доказали: этот язык нерегулярен.

Причина честная: у автомата только конечное число состояний, и бесконечный счёт в них не помещается.

Люди считают: карандаш, бумага, стирание и переписывание.

Модель вычисления обязана уметь то же самое — запоминать неограниченно много.

Человек за столом

Тьюринг в 1936 году описал вычисление так, как его делает человек с карандашом:

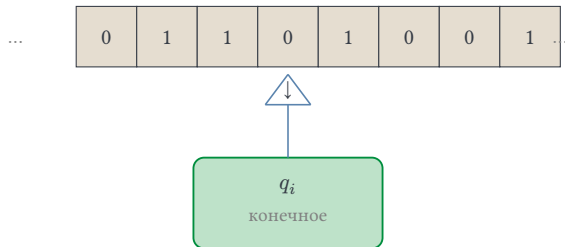
- конечное число «состояний души» — то, что человек помнит
- бумага, разбитая на клетки — то, куда он смотрит
- простые шаги — стереть, написать, сдвинуть взгляд.

Никакой магии: только символы, клетки и правила.

Всё остальное вычисление получает из их комбинаций.

Устройство машины

Лента:



Три части с чёткими ролями: бесконечная лента — память, головка — взгляд, конечное управление — программа.

Семёрка машины Тьюринга

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, содержит входной алфавит Σ и пробел.
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ — частичная функция переходов.
- q_0 — начальное состояние, q_{accept} и q_{reject} — принимающее и отвергающее.

Состояния конечны, лента — нет: весь неограниченный счёт живёт на ленте.

Один шаг

Определение 2 (Переход)

На паре (состояние, символ) правило δ даёт тройку (новое состояние, новый символ, сдвиг):

- прочитать символ под головкой
- записать новый
- сдвинуть головку влево или вправо
- сменить состояние.

δ частична: правила для текущей пары может не быть.

Остановка вне финальных состояний по соглашению считается отверганием.

Конфигурация

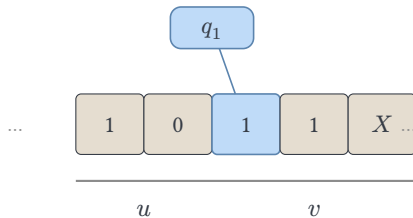
Определение 3 (Конфигурация)

Мгновенное описание работы — запись uqv :

- u — содержимое ленты слева от головки
- q — текущее состояние
- v — лента от головки вправо, включая читаемую ячейку.

Начальная конфигурация на входе w — это q_0w : головка на первом символе.

Анатомия конфигурации



Состояние стоит над читаемой ячейкой, u — всё слева, v — от головки до последнего непустого символа.

Три исхода

Определение 4 (Исходы работы)

- МТ **принимает** вход, если достигает q_{accept} .
- МТ **отвергает**, если достигает q_{reject} .
- МТ может **зациклиться** — не достигнуть ни того, ни другого.

Третий исход — не каприз определения, а суть модели.

Программа, которая крутится вечно, должна описываться честно.

Что значит решить язык?

Определение 5 (Распознаваемость и разрешимость)

Язык L **распознаваем**, если есть МТ, принимающая w в точности тогда, когда $w \in L$.

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

Язык L **разрешим**, если есть МТ, которая всегда останавливается: принимает $w \in L$, отвергает $w \notin L$.

Разрешимый язык распознаваем, а обратное — нет.

Пример распознаваемого, но неразрешимого языка появится через две лекции.

Пример: слова, заканчивающиеся на ноль

Пример (Семёрка в деле)

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

Состояния: q_0 — «последний символ 1», q_1 — «последний символ 0».

Переходы:

- $(q_0, 1) \rightarrow (q_0, 1, R)$
- $(q_0, 0) \rightarrow (q_1, 0, R)$
- $(q_0, \text{пробел}) \rightarrow (q_{\text{reject}}, \text{пробел}, R)$
- $(q_1, 0) \rightarrow (q_1, 0, R)$
- $(q_1, 1) \rightarrow (q_0, 1, R)$
- $(q_1, \text{пробел}) \rightarrow (q_{\text{accept}}, \text{пробел}, R)$

Трасса и разрешимость

Пример (Трасса на 10)

$$q_0 10 \rightarrow 1q_0 0 \rightarrow 10q_1 \square \rightarrow \text{принять.}$$

Последний шаг: q_1 видит пробел и принимает.

Машина читает $|w|$ символов, видит пробел и всегда останавливается.

Значит, язык разрешим — конечная память тут справляется без ленты.

Приёмы программирования

Примеры ниже собираются из повторяющихся приёмов.

Приём	Идея
Запоминание через состояние	«вижу символ a » — перейди в состояние «помню a » и неси его, пока не понадобится
Маркеры	обработанные символы заменяются на X : «уже учтено» отделяется от «ещё нет»
Проход к краю и возврат	«иди вправо, пока не пробел», шаг влево, переключение — цикл с условием на краю

Счёт на МТ — это всегда расстановка и пересчёт маркеров.

Память машины растёт не числом состояний, а занятой частью ленты.

Машина как процессор

Приёмы — прообраз настоящего процессора:

- состояние играет роль счётчика команд
- маркер — флага
- проход по ленте — цикла.

Программа на МТ — конечная таблица, и её можно записать строкой символов.

К чему это ведёт — в конце лекции.



Программы на ленте

Программы должны писаться для того, чтобы их читали люди, и лишь попутно — чтобы их исполняли машины.

— Гарольд Абельсон и Джеральд Сассман

Инверсия битов

Пример (Первый алгоритм)

Заменить каждый 0 на 1 и каждую 1 на 0.

Одно состояние, три правила:

- $(q_0, 0) \rightarrow (q_0, 1, R)$
- $(q_0, 1) \rightarrow (q_0, 0, R)$
- $(q_0, \text{пробел}) \rightarrow (q_{\text{accept}}, \text{пробел}, R)$

Эта машина вычисляет функцию, а не распознаёт язык: на ленте остаётся результат.

Разрешимость — про ответ «да или нет», вычислимость — про выход.

Инкремент двоичного числа

Пример (Прибавить единицу)

Число на ленте, младший бит слева.

Пока читаем 1 — пишем 0 и уносим перенос вправо.

Первая 0 меняется на 1, и работа закончена.

- $(q_0, 1) \rightarrow (q_0, 0, R)$
- $(q_0, 0) \rightarrow (q_{\text{accept}}, 1, R)$
- $(q_0, \text{пробел}) \rightarrow (q_{\text{accept}}, 1, R)$

Арифметика на ленте — один цикл с одним состоянием.

Инкремент: трасса

Пример (Трасса на 11)

1. $q_0 11$: читаем младшую 1, пишем 0, перенос вправо.
2. $0q_0 1$: снова 1, пишем 0, перенос вправо.
3. $00q_0 \square$: пробел — пишем 1 и принимаем.

На ленте 001: ровно $3 + 1$.

Все единицы исчерпаны — машина дописала новый разряд.

Лента сама расширяет число: ничего «перевыделять» не нужно.

Та самая $a^n b^n$

Пример (Вычёркивание пар)

Язык $a^n b^n$, не по силам конечным автоматам в первом семестре.

Стратегия: многократно вычеркнуть один a слева и одну b справа.

Трасса на $aabb$:

$Xa\ bb \rightarrow Xa\ bX \rightarrow XX\ bX \rightarrow XXXX \rightarrow \text{принять.}$

Конечному автомату не хватило состояний на счёт, а машине хватает ленты.

Сравнить два счётчика — значит стереть их попарно.

Палиндромы

Пример (Сравнение концов)

Алгоритм для палиндромов над $\{a, b\}$:

1. Запомнить первый символ — перейти в состояние «видел a » или «видел b » — и стереть его.
2. Дойти до последнего нестертого символа.
3. Совпал — стереть и вернуться к началу, нет — отвергнуть.
4. Всё стёрто — принять.

МТ видит оба конца строки — ценой многократных проходов.

Запоминание символа через состояние — тот же приём, что у автомата, но возвраты делает головка.

Композиция машин

Определение 6 (Композиция)

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

Из простых блоков собираются большие программы:

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

Удвоение унарного числа, затем инверсия битов — составная машина $1^n \rightarrow 0^{2n}$.

Таблица композиции — объединение таблиц с перенумерацией состояний.

Программа — конечный текст

Таблица переходов — конечный объект: её можно выписать строкой символов.

Значит, программу можно положить на ленту и читать её как данные.

Одна машина, которая делает это по правилам, исполнила бы любую программу.

Как такая машина устроена — тема следующей лекции.

Лекция 9: что мы поняли?

- МТ — бесконечная лента плюс конечное управление: неограниченная память при финитном описании.
- Конфигурация uqv — полный снимок работы, вычисление — цепочка конфигураций.
- Три исхода: принять, отвергнуть, заиклиться. Разрешимость — остановка с ответом на каждом входе.
- Программы пишутся приёмами: символ в состоянии, маркеры, проходы, композиция.
- Язык $a^n b^n$, не по силам автоматам, разрешается вычёркиванием пар — лента заменяет счётчик.

Вопросы для самопроверки

1. Почему $a^n b^n$ нерегулярен, но разрешим машиной Тьюринга?
2. Назовите все семь компонентов семёрки МТ и роль каждого.
3. Что записывает конфигурация uqv и как выглядит начальная конфигурация для входа 10?
4. Чем распознаваемость языка отличается от разрешимости?
5. Как машина запоминает символ, если у неё нет переменных?
6. Опишите машину инкремента двоичного числа: состояния и переходы.
7. Что значит скомпоновать две машины в одну?

Чтение к следующей паре

- m25 — «Определение машины Тьюринга», «Примеры машин Тьюринга».
- Флеш-опросник — в начале лекции 10.
- Теормин 1 (графы и автоматы, устно) — на отдельной паре этой недели.