

Лекция

9–10 марта

Детерминированные автоматы



Языки и слова

Язык есть орган, образующий мысль.

— Вильгельм фон Гумбольдт

Что читает машина?

Программа всегда читает строку.

Число записано цифрами, граф — перечнем рёбер, программа — текстом.

Всё, что машина читает, — строка.

Пример

Задачи одного и того же вида:

- редактор находит в документе все номера телефонов;
- компилятор режет текст программы на ключевые слова и числа;
- протокол связи решает, корректно ли выглядит принятое сообщение.

Во всех трёх задачах машина читает строку слева направо и отвечает «да» или «нет».

Начнём с самого простого устройства, способного на такой ответ.

Алфавит, слово, язык

Определение 1

- **Алфавит** — конечное множество символов Σ .
- **Слово** — конечная последовательность символов алфавита. Длина слова w обозначается $|w|$. **Пустое слово** ε — единственное слово длины 0.
- **Язык** — произвольное множество слов: $L \subseteq \Sigma^*$.

Пример

Над алфавитом $\{0, 1\}$ слово 1001 имеет длину 4.

Язык слов, заканчивающихся на 01, содержит 1001 и не содержит 1000.

Язык — множество слов. Распознать его — уметь отвечать для любого слова, принадлежит ли оно.

Пустое слово и пустой язык

Три множества слов легко перепутать.

Язык	Что содержит
------	--------------

$\emptyset$	ни одного слова
$\{\epsilon\}$	одно слово длины 0
Σ^*	все слова алфавита

Язык $\{\epsilon\}$ и язык $\emptyset$ различны: в первом есть слово, во втором слов нет вовсе.

Пустое слово в языке L означает то же, что ноль элементов в множестве чисел: это содержательный факт.

Сколько слов над алфавитом?

Теорема 1 (Подсчёт слов)

Если $|\Sigma| = k$, то слов длины ровно n — ровно k^n .

Доказательство

Докажем подсчётом позиций.

Слово длины n — это n независимых выборов из k символов.

Всего выборов $k \cdot k \cdots k = k^n$.

Пример

Двоичный алфавит, слова длины не больше 2:

$$1 + 2 + 4 = 7.$$

Все ли языки можно описать?

Язык — подмножество Σ^* .

Выбрать язык — значит выбрать подмножество счётного множества.

Теорема 2 (Подсчёт языков)

Над непустым алфавитом языков ровно $2^{\aleph_0}$ — континуум.

Доказательство

Докажем подсчётом подмножеств. Языки — булеан Σ^* , равномогущий бесконечным двоичным последовательностям, а их несчётно много по диагональному аргументу Кантора.

Конечных строк над Σ — счётное число, а языков — континуум, и почти ни один язык нельзя описать конечным текстом. Курс изучает языки, которые описать *можно*.

Операции над языками

Определение 2

- **Конкатенация:** $L_1 L_2 = \{uv \mid u \in L_1, v \in L_2\}$.
- **Степень:** $L^0 = \{\varepsilon\}$, $L^{n+1} = L^n L$.
- **Звезда Клини:** $L^* = \bigcup_{n \geq 0} L^n$ — ноль и более повторений.

Пример

$\{a, ab\} \cdot \{c\} = \{ac, abc\}$.

$\{0, 1\}^2$ — четыре слова длины 2.

$\{a\}^*$ — слова $a, aa, aaa, \dots$ и пустое слово.

Звезда Клини — сжатая запись цикла: сколько угодно раз пройти по одному и тому же куску.

Алгебра языков

Языки — множества, поэтому над ними работают все операции множеств:

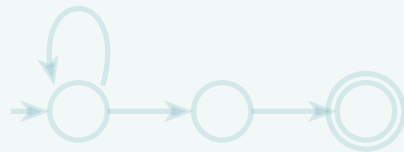
$$L_1 \cup L_2, \quad L_1 \cap L_2, \quad \overline{L} = \Sigma^* \setminus L.$$

Законы де Моргана из логики высказываний переносятся на языки дословно:

$$\overline{L_1 \cup L_2} = \overline{L_1} \cap \overline{L_2}.$$

Комбинаторика языков опирается на знакомый фундамент: операции — это связки, включение — импликация, булеан даёт верхнюю границу.

Нового здесь только одна операция: конкатенация, у числовых множеств аналога нет.



Детерминированные автоматы

Только разнообразие может поглотить разнообразие.

— Уильям Росс Эшби

Три гипотезы о конце слова

Хотим автомат для языка слов, заканчивающихся на 01.

Что нужно помнить о прочитанном префиксе?

- Ничего полезного для конца — состояние q_0 .
- Заканчивается на 0: возможно, это предпоследний символ — состояние q_1 .
- Заканчивается на 01 — цель достигнута — состояние q_2 .

Гипотезы о будущем превращаются в состояния автомата.

Переход — это обновление гипотезы при чтении очередного символа.

Детерминированный конечный автомат

Определение 3 (ДКА)

Пятёрка $M = (Q, \Sigma, \delta, q_0, F)$:

- Q — конечное множество состояний;
- Σ — алфавит;
- $\delta : Q \times \Sigma \rightarrow Q$ — всюду определённая функция переходов;
- $q_0 \in Q$ — начальное состояние;
- $F \subseteq Q$ — принимающие состояния.

Пример

Таблица переходов — задание δ в конечном виде.

Детерминизм: следующее состояние известно заранее — вся память автомата есть номер состояния.

Автомат как размеченный оргграф

У автомата есть графовая внешность:

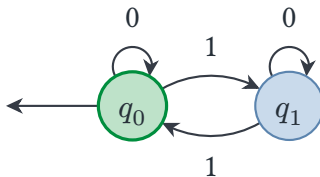
В автомате	В оргграфе
состояния	вершины
переходы	дуги
метки переходов	цвета дуг
начальное состояние	выделенная вершина
принимающие состояния	выделенное подмножество вершин

Вычисление — путь из выделенной вершины, метки пути складываются в слово.

Язык автомата — метки путей, кончающихся в принимающих вершинах.

Это тот же граф, с которым работала теория графов: достижимость, циклы, обходы.

Пример: чётное число единиц



Пример

Два состояния: q_0 — чётное число единиц, q_1 — нечётное.

Символ 0 состояние не меняет, символ 1 переключает.

Прогон 011: $q_0 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0$ — принято.

Прогон 01: $q_0 \rightarrow q_0 \rightarrow q_1$ — отвергнуто.

Пустое слово принимается: ноль единиц — чётное число.

Расширенная функция переходов

Определение 4 (Расширенная функция переходов)

$\hat{\delta} : Q \times \Sigma^* \rightarrow Q$ протягивает δ на целые слова:

- **Базис:** $\hat{\delta}(q, \varepsilon) = q$.
- **Шаг:** $\hat{\delta}(q, aw) = \hat{\delta}(\delta(q, a), w)$.

Пример

На автомате чётности:

$$\hat{\delta}(q_0, 011) = \hat{\delta}(\delta(q_0, 0), 11) = \hat{\delta}(q_0, 11) = q_0.$$

Слово 011 принято.

Принять слово — значит проверить одно неравенство: $\hat{\delta}(q_0, w) \in F$.

Язык автомата

Определение 5

$$L(M) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\}.$$

Язык называется **регулярным**, если он распознаётся некоторым ДКА.

Пример

Автомат чётности распознаёт язык слов с чётным числом единиц.

Три состояния из гипотез распознают язык слов, заканчивающихся на 01.

Два автомата эквивалентны, если распознают один и тот же язык.

Класс регулярных языков — это то, что описывают автоматы с конечной памятью.

Автомат в программе

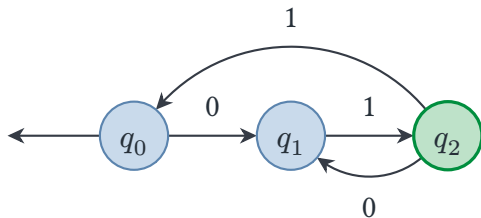
Принятие слова — цикл по символам, один шаг — один табличный переход.

```
let mut q = start;
for &sym in word {
    q = trans[q][sym];
}
accept[q]
```

Массив `trans` — это и есть таблица δ .

Каждый символ обрабатывается за $O(1)$: такая распознающая часть называется **конечным автоматом** и в программировании тоже.

Слова, заканчивающиеся на «01»



Пример

Три состояния: ожидание, «заканчивается на 0», «заканчивается на 01».

Прогон 1001: $q_0 \rightarrow q_0 \rightarrow q_1 \rightarrow q_1 \rightarrow q_2$ — принято.

Прогон 1000: $q_0 \rightarrow q_0 \rightarrow q_1 \rightarrow q_1 \rightarrow q_1$ — отвергнуто.

Три состояния хватает на любую длину: в памяти лишь важное для будущего.

Что у автомата в памяти?

Состояние — это сжатая история прочитанного префикса.

- Язык «заканчивается на 01» — три состояния: достаточно помнить последний символ или два.
- Язык «чётное число единиц» — два состояния: достаточно помнить чётность.
- Язык «в слове поровну нулей и единиц» — нужно помнить разность, а она не ограничена.

Конечность памяти — граница модели.

Насколько она жёсткая, станет ясно, когда появится инструмент доказательства невозможности.

Такой инструмент появится через две лекции.

Лекция 5: что мы поняли?

- Всё, что читает машина, — строка: алфавит, слово, язык $\subseteq \Sigma^*$.
- Слов счётное число, языков — континуум: конечным описанием описана малая доля языков.
- ДКА — пятёрка $(Q, \Sigma, \delta, q_0, F)$, вся память — номер состояния.
- Язык автомата собирается расширенной функцией $\hat{\delta}$ и условием $\hat{\delta}(q_0, w) \in F$.
- Автомат — размеченный орграф: распознавание — путь от выделенной вершины.

Вопросы для самопроверки

1. Над алфавитом $\{0, 1\}$ сколько слов длины 3? А длины не больше 3?
2. Постройте ДКА для языка слов над $\{0, 1\}$, содержащих подстроку 10.
3. Почему $\emptyset$ и $\{\varepsilon\}$ — разные языки? Какой из них распознаёт ДКА с недостижимым принимающим состоянием?
4. Слово ab длины 2 переводит автомат из q_0 в q_2 . Чему равно $\hat{\delta}(q_0, ab)$ и как это вычислить по шагам?

Чтение к следующей паре

- m23 — «Алфавит, слова и формальные языки», «Детерминированные конечные автоматы».
- Флеш-опросник — в начале лекции 6.