

ТЕОРІЯ ФОРМАЛЬНИХ МОВ

Лекція 4

Детерміновані скінченні автомати (ДСА)

Формальна модель DFA · стани й переходи · таблиця переходів · граф автомата



ПЛАН ЛЕКЦІЇ

Що ми розглянемо сьогодні

1

Поняття скінченного автомата

Розпізнавач мови, що читає слово посимвольно

2

Формальна модель ДСА

П'ятірка $(Q, \Sigma, \delta, q_0, F)$

3

Стани і переходи

Початковий стан і множина фінальних станів

4

Таблиця переходів

Табличне подання функції δ

5

Граф автомата

Наочна діаграма станів

6

Зв'язок з граматиками й виразами

Три способи задання регулярної мови

Ще один спосіб задати регулярну мову

У лекції 3 регулярну мову задавали граматикою і регулярним виразом. Третій, обчислювальний спосіб - скінченний автомат. Це абстрактна «машина», яка читає слово посимвольно зліва направо і видає відповідь «належить» / «не належить».

регулярна граматика ↔ регулярний вираз ↔ скінченний автомат

ДЕТЕРМІНОВАНИЙ скінченний автомат (ДСА / DFA (Deterministic Finite Automaton))

Найпростіший вид автомата, де для кожного стану і кожного вхідного символу визначено рівно один наступний стан.

Скінчений автомат

Задача

Для алфавіту $\Sigma = \{0, 1\}$ і мови $L = \{w \in \{0,1\}^* \mid w \text{ закінчується на } 1\}$ потрібен механізм, який для будь-якого слова (101, 1101, 000, 1110, ...) визначає, належить воно L чи ні.

Скінченний автомат

Абстрактна математична модель обчислювального пристрою, який читає вхідне слово символ за символом і переходить між скінченною кількістю станів.

Автомат має «пам'ять» дуже простої форми - він пам'ятає лише свій поточний стан.

Що означає «детермінований»

Головна властивість

Поведінка автомата повністю визначена, якщо автомат перебуває у певному стані та отримує певний символ, він має рівно один можливий наступний стан.

Приклад переходу

$$\delta(q_0, 0) = q_1$$

з q_0 за символом 0 автомат переходить у q_1 .



Не може бути одночасно $\delta(q_0, 0) = q_1$ і $\delta(q_0, 0) = q_2$ - для кожної пари (стан, символ) існує один і тільки один наступний стан.

Формальне визначення ДСА

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

 Q **Стани**

Скінченна непорожня
множина станів
автомата

 Σ **Алфавіт**

Вхідний алфавіт (той
самий, що й для мов)

 δ **Переходи**

$Q \times \Sigma \rightarrow Q$: рівно один
наступний стан

 q_0 **Старт**

Початковий стан, $q_0 \in Q$

 F **Фінали**

Множина приймальних
станів, $F \subseteq Q$

Слово w приймається, якщо після його прочитання автомат перебуває у стані з F :

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

Стан = «пам'ять» про прочитане

Кожен стан ДСА можна тлумачити як мінімальну інформацію про вже прочитану частину слова, потрібну для продовження розпізнавання.

ПРИКЛАД: мова слів над $\{0,1\}$, що закінчуються на 1

q_0

останній символ 0, або слово ще
порожнє

q_1^*

останній символ 1 (фінальний стан)

Початковий стан: q_0 . **Фінальний стан:** $F = \{q_1\}$.

Початковий стан q_0 єдиний: з нього автомат розпочинає роботу над кожним новим словом (якщо $q_0 \in F$, то і порожнє слово ϵ належить мові). Фінальні стани $F \subseteq Q$ визначають, чи приймається слово.

Графічне й табличне представлення

01

Граф переходів

- стан - коло;
- початковий стан - стрілка зліва;
- заключний стан - подвійне коло;
- перехід - стрілка з підписом символу.

02

Таблиця переходів

Рядки - стани, стовпці - символи алфавіту; клітинка містить наступний стан.

Важливо

Граф і таблиця - два еквівалентні способи задання функції δ . Таблиця фактично є «програмою» поведінки автомата.

Табличне подання функції δ

Рядки - стани, стовпці - символи алфавіту, у комірках - наступний стан.

Стан	$\delta(\cdot, 0)$	$\delta(\cdot, 1)$	Тип стану
q_0	q_0	q_1	початковий
q_1	q_0	q_1	фінальний (приймальний)

Щоб обробити слово, «рухаємось» по таблиці: беремо рядок поточного стану та стовпець наступного символу вхідного слова - і так до кінця слова.

4

ПРИКЛАД

Слова, що закінчуються на 1

Мова: $L = \{w \in \{0,1\}^* \mid w \text{ закінчується на } 1\}$

Стан	0	1
q0 (останній $\neq 1$)	q0	q1
q1 (останній = 1)	q0	q1

q0 — початковий, $F = \{q1\}$

1 → приймається

01 → приймається

101 → приймається

0 → відхиляється

100 → відхиляється

4

ПРИКЛАД

Як працює ДСА, трасування слова 110

1

Стан q_0 , читаємо символ 1

$$\delta(q_0, 1) = q_0$$

2

Стан q_0 , читаємо символ 1

$$\delta(q_0, 1) = q_0$$

3

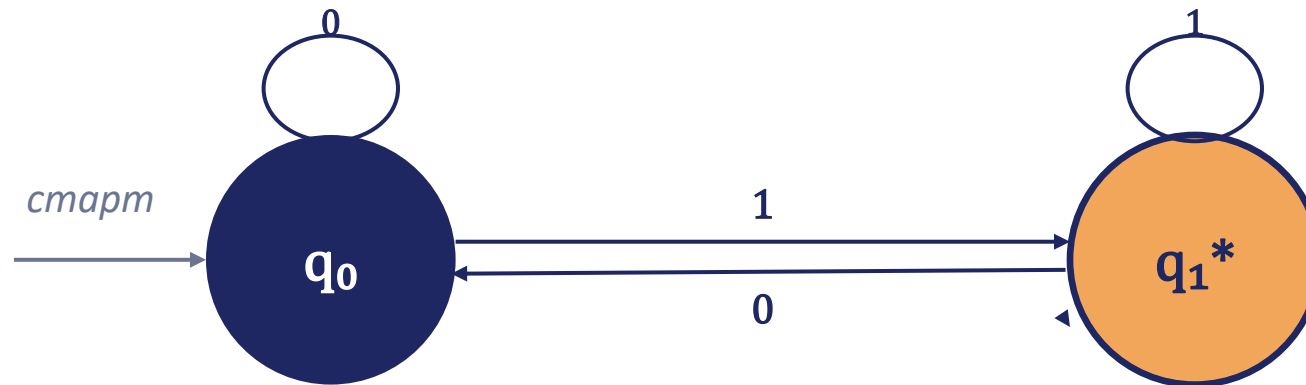
Стан q_0 , читаємо символ 0

$$\delta(q_0, 0) = q_1$$

Результат

$q_0 \rightarrow q_0 \rightarrow q_0 \rightarrow q_1$. Оскільки q_1 є заключним станом, слово 110 приймається.

Наочне подання переходів



Вершини - стани (фінальний стан - подвійним колом/зіркою); дуги підписані символами; початковий стан - вхідна стрілка без джерела.

Відповідна регулярна граматика (лекція 3): $q_0 \rightarrow 0q_0 \mid 1q_1$; $q_1 \rightarrow 0q_0 \mid 1q_1 \mid \varepsilon$

Перевірка слів «101» і «100»

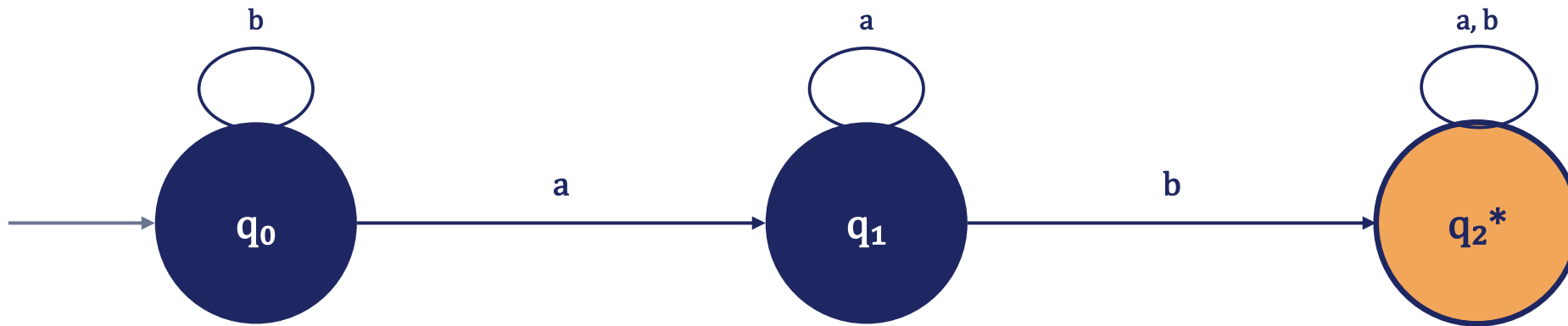
Крок	Поточний стан	Символ	Наступний стан
0	q_0 (старт)	-	-
1	q_0	1	q_1
2	q_1	0	q_0
3	q_0	1	q_1

«101» → завершує в $q_1 \in F$ → ПРИЙМАЄТЬСЯ

«100» → останній перехід $q_1 \rightarrow (0)q_0$ → завершує в $q_0 \notin F$ → ВІДХИЛЯЄТЬСЯ

Мова, де слова, що містять підрядок «ab»

$Q = \{q_0, q_1, q_2^*\}$, $\Sigma = \{a, b\}$, q_0 - початковий, $F = \{q_2\}$



«bbaab»: $q_0 \rightarrow q_0 \rightarrow q_1 \rightarrow q_1 \rightarrow q_2 \rightarrow q_2$ - ПРИЙМАЄТЬСЯ

«aaa»: $q_0 \rightarrow q_1 \rightarrow q_1 \rightarrow q_1$ - ВІДХИЛЯЄТЬСЯ (підрядок «ab» так і не з'явився)

Стан q_2 - «пастка успіху»: щойно підрядок «ab» зустрівся, автомат залишається фінальним назавжди.

Мова автомата та функція δ^*

Мова, яку розпізнає автомат $A = (Q, \Sigma, \delta, q_0, F)$

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

ДСА приймає слово тоді і тільки тоді, коли після читання всіх його символів він опиняється у заключному стані.

Розширена функція переходів δ^* (рекурсивне означення)

$$\delta^*(q, \epsilon) = q$$

$$\delta^*(q, wa) = \delta(\delta^*(q, w), a)$$

δ^* дозволяє обробляти не один символ, а ціле слово.

«Пам'ять» ДСА

ДСА не зберігає все прочитане слово

Він зберігає лише інформацію, потрібну для прийняття рішення. Для мови «слова з парною кількістю 1» автомату не потрібно знати, скільки саме одиниць було — 2, 4, 100 чи 1000.

Йому достатньо знати: кількість одиниць парна чи непарна. Тому потрібні лише два стани.

Це одна з головних ідей теорії скінченних автоматів.

Повний ДСА та поглинаючий стан

Повнота

Для стандартного (повного) ДСА функція переходів визначена для кожної пари $(q,a) \in Q \times \Sigma$. Якщо $|Q|=3$ і $|\Sigma|=2$, мають бути визначені всі $3 \times 2 = 6$ переходів.

Поглинаючий стан (пастка, dead state)

Якщо перехід логічно означає «слово більше не може бути прийняте», вводять стан-пастку q_dead , з якого немає виходу: $\delta(q_dead, a) = q_dead$ для будь-якого a .

Три еквівалентні способи задання регулярної мови

РЕГУЛЯРНА ГРАМАТИКА

$$S \rightarrow aS \mid b$$

Нетермінал = стан; правило $A \rightarrow aB$ = перехід $A \rightarrow B$ по a

РЕГУЛЯРНИЙ ВИРАЗ

$$a^*b$$

Компактний запис тієї самої мови

СКІНЧЕННИЙ АВТОМАТ

$$q_0 \rightarrow(a) q_0 \rightarrow(b) q_1^*$$

Обчислювальна модель розпізнавання

Обернене перетворення (автомат $\rightarrow$ регулярний вираз) і строге доведення еквівалентності - теорема Кліні (наступні лекції). Далі - недетерміновані автомати (НСА), де можливо кілька наступних станів або переходи без символу.

Ключові тези

- ДСА $M = (Q, \Sigma, \delta, q_0, F)$ - розпізнавач регулярної мови: слово приймається, якщо після його прочитання автомат перебуває у стані з F .
- Функція переходів δ визначає рівно один наступний стан для кожної пари (стан, символ) - це властивість детермінованості.
- Таблиця переходів і граф автомата - дві еквівалентні форми подання δ .
- Стан автомата відповідає мінімальній інформації про вже прочитану частину слова.
- ДСА, регулярна граматика й регулярний вираз - три еквівалентні способи задання одного класу регулярних мов.