Automi e Linguaggi Formali

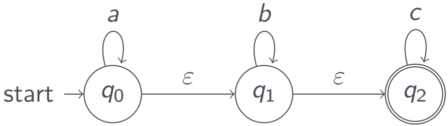
a.a. 2017/2018

LT in Informatica
8 Marzo 2018

Università degli Studi di Padova

Esercizio

1. Costruiamo un ε-NFA che riconosce le parole costituite da zero o più a, seguite da zero o più b, seguite da zero o più c



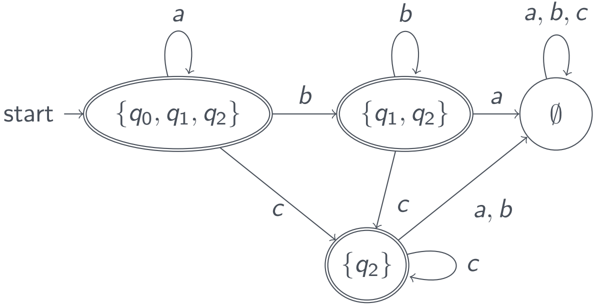
2. Calcolare la ε-chiusura di ogni stato

$$
\begin{aligned}
\mathrm{ECLOSE}(q_0) &= \{q_0, q_1, q_2\} \\
\mathrm{ECLOSE}(q_1) &= \{q_1, q_2\} \\
\mathrm{ECLOSE}(q_2) &= \{q_2\}
\end{aligned}
$$

3. Convertire l'ε-NFA in DFA

Università degli Studi di Padova

1. Costruiamo un ε-NFA che riconosce le parole costituite da zero o più *a*, seguite da zero o più *b*, seguite da zero o più *c*
2. Calcolare la ε-chiusura di ogni stato
3. Convertire l'ε-NFA in DFA



Theorem

Sia $D = (Q_D, \Sigma, S_0, F_D)$ il DFA ottenuto da un ε-NFA E con la costruzione a sottoinsiemi modificata. Allora $L(D) = L(E)$.

Dimostrazione: Prima mostriamo per induzione su $|w|$ che

$$\hat{\delta}_D(S_0, w) = \hat{\delta}_E(q_0, w)$$

Base: $w = \varepsilon$. L'enunciato segue dalla definizione:

- Lo stato iniziale di D è $S_0 = \mathrm{ECLOSE}(q_0)$;
- $\hat{\delta}_D(S_0, \varepsilon) = S_0 = \mathrm{ECLOSE}(q_0)$;
- $\hat{\delta}_E(q_0, \varepsilon) = \mathrm{ECLOSE}(q_0)$.

Induzione:

- Sia $|w| = n+1$ e supponiamo vero l'enunciato per la lunghezza n. Scomponiamo w in $w = xa$ (con $|x| = n$ e a simbolo finale)
- Per ipotesi induttiva $\hat{\delta}_D(S_0, x) = \hat{\delta}_E(q_0, x) = \{p_1, \dots, p_k\}$
- Per la definizione di $\hat{\delta}$ per gli ε-NFA

$$\hat{\delta}_E(q_0, xa) = \mathrm{ECLOSE}\left(\bigcup_{i=1}^{k} \delta_E(p_i, a)\right)$$

- Per la costruzione a sottoinsiemi

$$\delta_D(\{p_1, \dots, p_k\}, a) = \mathrm{ECLOSE}\left(\bigcup_{i=1}^{k} \delta_E(p_i, a)\right)$$

Induzione (continua):

- Per la definizione di $\hat{\delta}$ per i DFA

$$\hat{\delta}_D(S_0, xa) = \delta_D(\{p_1, \ldots, p_k\}, a) = \mathrm{ECLOSE}\left(\bigcup_{i=1}^{k} \delta_E(p_i, a)\right)$$

- Quindi abbiamo mostrato che $\hat{\delta}_D(S_0, w) = \hat{\delta}_E(q_0, w)$

Poiché sia D che E accettano se solo se $\hat{\delta}_D(S_0, w)$ e $\hat{\delta}_E(q_0, w)$ contengono almeno un stato in F_E, allora abbiamo dimostrato che $L(D) = L(N)$.

Teorema di equivalenza tra DFA e NFA

Università degli Studi di Padova

Theorem

Un linguaggio L è accettato da un DFA se e solo se è accettato da un ε-NFA.

Dimostrazione:

- La parte "se" è il teorema precedente
- La parte "solo se" si dimostra osservando che ogni DFA può essere trasformato in un ε-NFA modificando δ_D in δ_E con la seguente regola:

  Se $\delta_D(q, a) = p$ allora $\delta_E(q, a) = \{p\}$

Espressioni Regolari

- Un FA (NFA o DFA) è un metodo per costruire una macchina che riconosce linguaggi regolari
- Una espressione regolare è un modo dichiarativo per descrivere un linguaggio regolare.
- Esempio: $\mathbf{01}^* + \mathbf{10}^*$
- Le espressioni regolari sono usate, ad esempio, in:
  - comandi UNIX (`grep`)
  - strumenti per l'analisi lessicale di UNIX (`lex` (Lexical analyzer generator) e `flex` (Fast Lex))
  - editor di testo

Operazioni sui linguaggi

- **Unione:**

$$L \cup M = \{w : w \in L \text{ oppure } w \in M\}$$

- **Concatenazione:**

$$L.M = \{uv : u \in L \text{ e } v \in M\}$$

- **Potenze:**

$$L^0 = \{\varepsilon\} \qquad L^1 = L \qquad L^k = L.L^{k-1} = \underbrace{L.L.L.\dots.L}_{k \text{ volte}}$$

- **Chiusura (o Star) di Kleene:**

$$L^* = \bigcup_{i=0}^{\infty} L^i$$

Espressione regolari: sintassi

Le Espressioni Regolari sono costruite utilizzando

- un insieme di costanti di base:
  - ε per la stringa vuota
  - $\emptyset$ per il linguaggio vuoto
  - $\mathbf{a}, \mathbf{b}, \ldots$ per i simboli $a, b, \dots \in \Sigma$
- collegati da operatori:
  - $+$ per l'unione
  - $\cdot$ per la concatenazione
  - * per la chiusura di Kleene
- raggruppati usando le parentesi:
  - $(\quad)$

Se $\mathbf{E}$ è un espressione regolare, allora $L(\mathbf{E})$ è il linguaggio denotato da $\mathbf{E}$. La definizione di $L(\mathbf{E})$ è induttiva:

- **Caso Base:**
  - $L(\varepsilon) = \{\varepsilon\}$
  - $L(\emptyset) = \emptyset$
  - $L(\mathbf{a}) = \{a\}$
- **Caso induttivo:**
  - $L(\mathbf{E} + \mathbf{F}) = L(\mathbf{E}) \cup L(\mathbf{F})$
  - $L(\mathbf{EF}) = L(\mathbf{E}).L(\mathbf{F})$
  - $L(\mathbf{E}^*) = L(\mathbf{E})^*$
  - $L((\mathbf{E})) = L(\mathbf{E})$

Espressioni regolari: esempio

- Scriviamo l'espressione regolare per

$$L = \{w \in \{0,1\}^* : 0 \text{ e } 1 \text{ alternati in } w\}$$

$$(\mathbf{01})^* + (\mathbf{10})^* + \mathbf{1}(\mathbf{01})^* + \mathbf{0}(\mathbf{10})^*$$

oppure

$$(\varepsilon + \mathbf{1})(\mathbf{01})^*(\varepsilon + \mathbf{0})$$

Espressioni regolari: precedenza

Come per le espressioni aritmetiche, anche per le espressioni regolari ci sono delle regole di precedenza degli operatori:

1. Chiusura di Kleene
2. Concatenazione (punto)
3. Unione ($+$)

Esempio:

$$\mathbf{01}^* + \mathbf{1} \text{ è raggruppato in } (\mathbf{0}(\mathbf{1})^*) + \mathbf{1}$$

e denota un linguaggio diverso da

$$(\mathbf{01})^* + \mathbf{1}$$

Esercizi (1)

Per ognuno dei seguenti linguaggi, costruire una ER sull'alfabeto $\{a, b, c\}$ che li rappresenti:

1. Tutte le stringhe w che contengono un numero pari di a;
2. Tutte le stringhe w che contengono $4k + 1$ occorrenze di b, per ogni $k \geq 0$;
3. Tutte le stringhe la cui lunghezza è un multiplo di 3;

Esercizi (2)

Università degli Studi di Padova

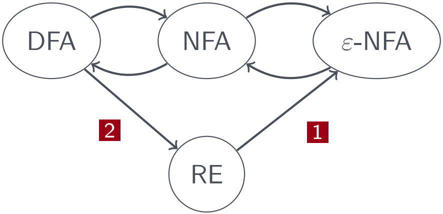
Per ognuno dei seguenti linguaggi, costruire una ER sull'alfabeto $\{0, 1\}$ che li rappresenti:

4. Tutte le stringhe w che contengono la sottostringa 101
5. Tutte le stringhe w che non contengono la sottostringa 101

Sfida!

Costruire una ER sull'alfabeto $\{0, 1\}$ per il linguaggio di tutti i numeri binari multipli di 3.

Sappiamo già che DFA, NFA, e ε-NFA sono tutti equivalenti.



Gli FA sono equivalenti alle espressioni regolari:

1. Per ogni espressione regolare $\mathbf{R}$ esiste un ε-NFA A, tale che $L(A) = L(\mathbf{R})$
2. Per ogni DFA A possiamo costruire un'espressione regolare $\mathbf{R}$, tale che $L(\mathbf{R}) = L(A)$

Theorem

Per ogni espressione regolare $\mathbf{R}$ *possiamo costruire un* ε*-NFA* A *tale che* $L(A) = L(\mathbf{R})$

Dimostrazione:

Costruiremo un ε-NFA A con:

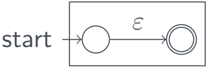
- un solo stato finale
- nessuna transizione entrante nello stato iniziale
- nessuna transizione uscente dallo stato finale

La dimostrazione è per induzione strutturale su $\mathbf{R}$

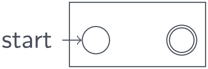
Da RE a ε-NFA

Caso Base:

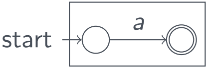
- automa per ε



- automa per $\emptyset$



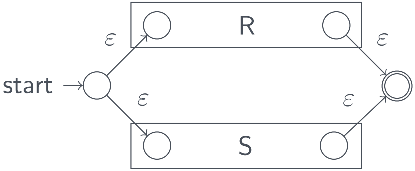
- automa per **a**



Da RE a ε-NFA

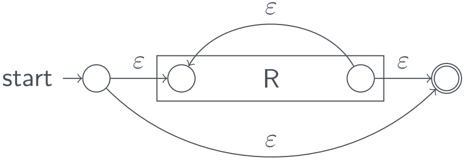
Caso Induttivo:

- automa per $\mathbf{R} + \mathbf{S}$



- automa per $\mathbf{RS}$

- automa per $\mathbf{R}^*$



Esercizio

Trasformiamo $(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})$ in ε-NFA