Da Espressioni Regolari a NFA equivalenti

Teorema Data una espressione regolare s, possiamo costruire un NFA N[s] tale che

$$\mathcal{L}[s] = L[N[s]],$$

(ovvero gli NFA riconoscono "tutti" i linguaggi regolari!! Vedremo che gli NFA riconoscono "solo" linguaggi regolari)

Dimostrazione per induzione sulla sintassi (astratta) della espressione regolare s

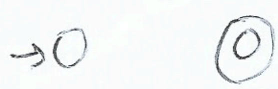
Costruiremo N[s], cioè un possibile NFA associato alla espressione regolare s, in modo da mantenere i seguenti due invarianti (per semplificare la costruzione):

(1) lo stato iniziale non ha archi entranti

(2) N[s] ha un solo stato finale senza archi uscenti

Esaminiamo i vari casi:

- $s = \phi$ $N[s] =$



(due stati non connessi)

Osserva che $\mathcal{L}[\phi] = \phi = L[N[s]]$

- $s = \varepsilon$ $N[s] =$

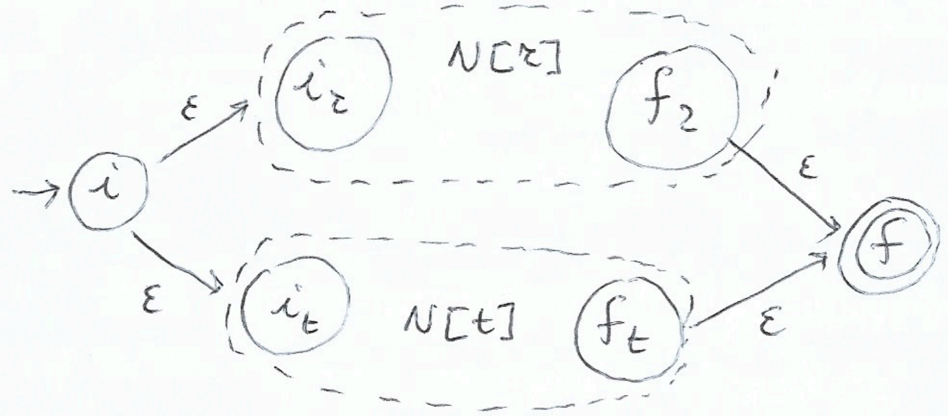
Osserva che $\mathcal{L}[\varepsilon] = \{\varepsilon\} = L[N[s]]$

- $s = a$ $\quad N[s] =$

Osserva che $\mathcal{L}[a] = \{a\} = L[N[s]]$

- $s = r|t$

$N[s] =$

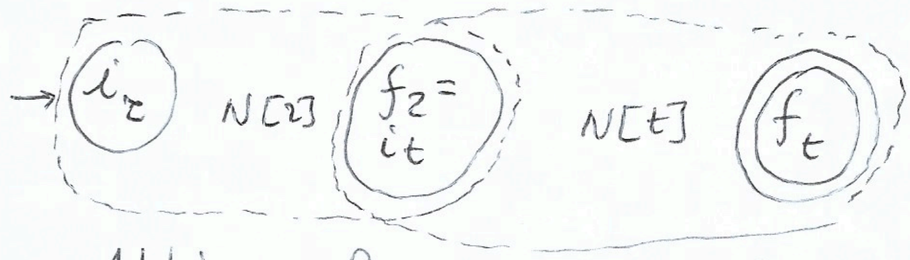


Osserva che

$$\mathcal{L}[r|t] = \mathcal{L}[r] \cup \mathcal{L}[t] \overset{\text{per ipotesi induttiva}}{=} L[N[r]] \cup L[N[t]] = L[N[r|t]]$$

- $s = r \cdot t$

$N[s] =$



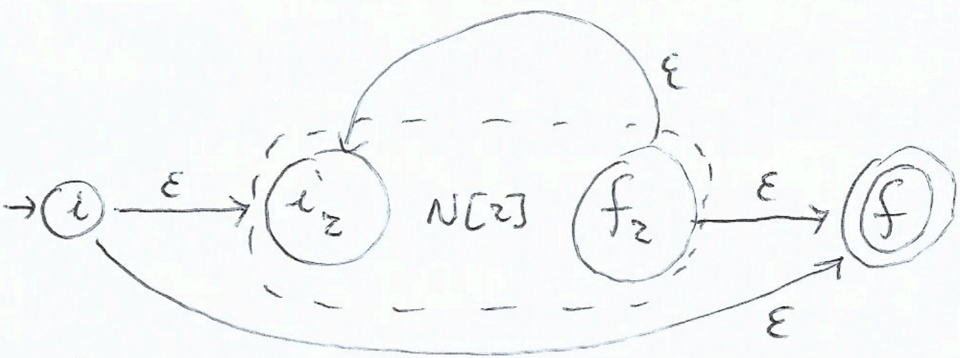
Abbiamo fuso insieme il finale di N[r] con l'iniziale di N[t]

Osserva che

$$\mathcal{L}[r \cdot t] = \mathcal{L}[r] \cdot \mathcal{L}[t] \overset{\text{per ipotesi induttiva}}{=} L[N[r]] \cdot L[N[t]] = L[N[r \cdot t]]$$

- $s = r^*$

$N[s] =$



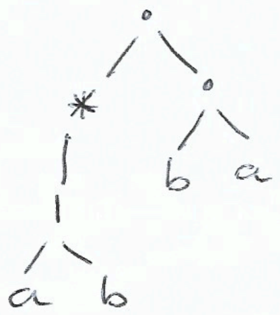
Osserva che

$$\mathcal{L}[r^*] = (\mathcal{L}[r])^* \underset{\substack{\uparrow \\ \text{per ipotesi induttiva}}}{=} (L[N[r]])^* = L[N[r^*]]$$

c.v.d

Esempio

$S = (a|b)^* ba$



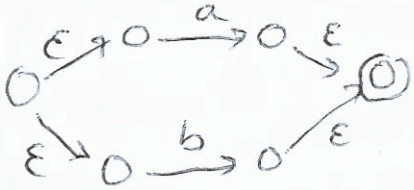
albero sintattico

Partiamo dalle foglie e risaliamo alla radice (bottom-up)

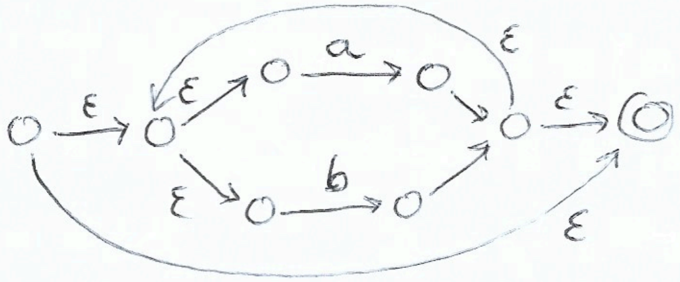
N[a]

N[b]

N[a|b]

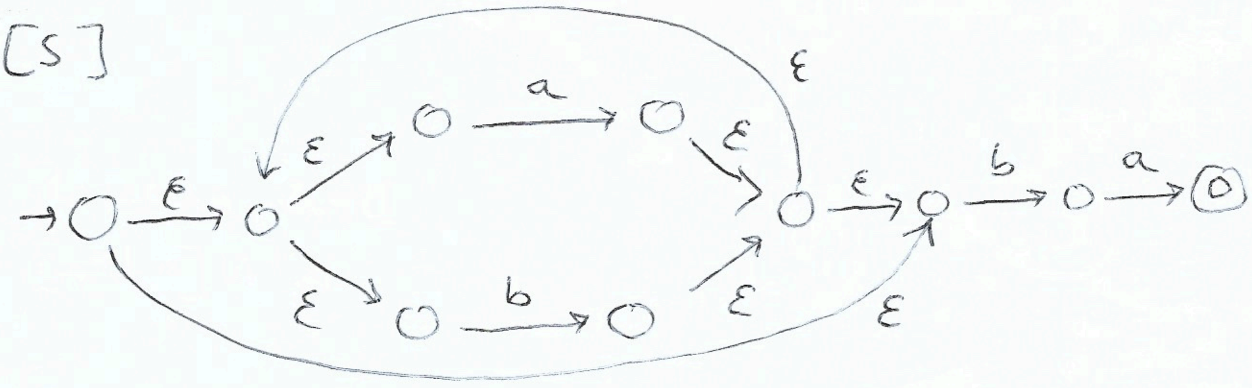


N[(a|b)*]



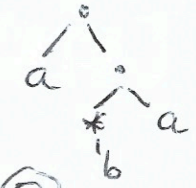
N[ba]

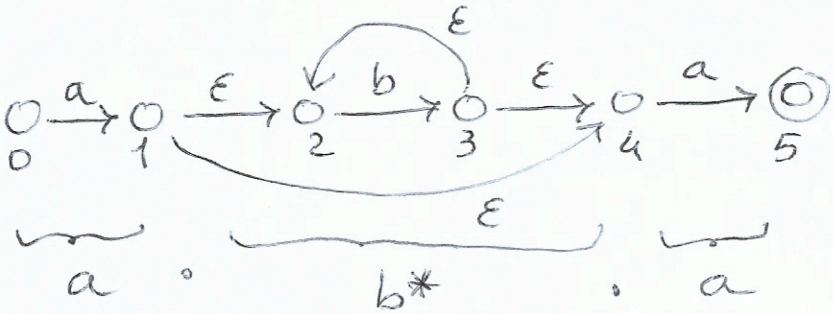
N[s]



Altro Esempio

$S = a\, b^* a$





$N[S]$

Costruiamo il DFA associato a $N[S]$

$A = \varepsilon\text{-closure}(0) = \{0\}$

$\Delta(A, a) = \varepsilon\text{-closure}(\{1\}) = \{1, 2, 4\} = B$

$\Delta(A, b) = \varepsilon\text{-closure}(\emptyset) = \emptyset = C$

$\Delta(B, a) = \varepsilon\text{-closure}(\{5\}) = \{5\} = D$

$\Delta(B, b) = \varepsilon\text{-closure}(\{3\}) = \{3, 4, 2\} = E$

$\Delta(D, a) = \emptyset = C$

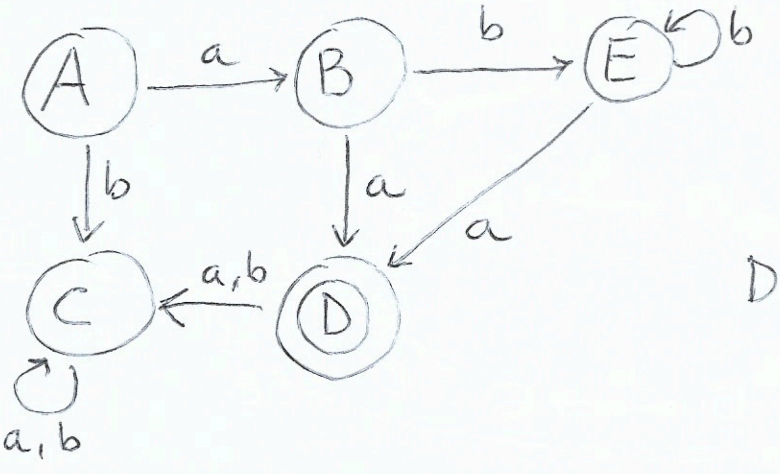
$\Delta(D, b) = \emptyset = C$

$\Delta(E, a) = \varepsilon\text{-closure}(\{5\}) = D$

$\Delta(E, b) = \varepsilon\text{-closure}(\{3\}) = E$

$\Delta(C, a) = \emptyset = C$

$\Delta(C, b) = \emptyset = C$



DFA associato a $N[a\, b^* a]$

Grammatiche Regolari

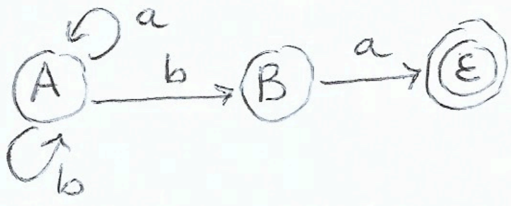
Def: Una grammatica libera è regolare sse ogni produzione è della forma $V \to aW$ oppure $V \to a$ dove $V, W \in NT$ e $a \in T$. Per il simbolo iniziale S è ammessa anche la produzione $S \to \varepsilon$.

(**N.B.** A volte useremo una definizione più lasca che permette produzioni $V \to \varepsilon$ anche per nonterminali diversi da S.)

Esempio (1)

$$\left.\begin{array}{l} A \to aA \mid bB \mid bA \\ B \to a \end{array}\right] \quad \begin{array}{l} G \text{ è regolare} \\ L(G) = (a|b)^*ba \text{ !!} \end{array}$$

Proviamo ad associare un NFA a G:



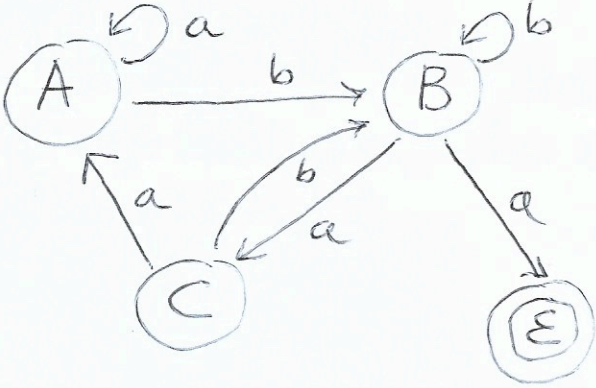
(solito automa NFA per $(a|b)^*ba$)

- Abbiamo associato ad ogni nonterminale uno stato dell'NFA
- abbiamo aggiunto uno stato finale "ε"

È possibile, in generale, associare ad ogni grammatica regolare un NFA equivalente?

Esempio (2)

$$\left.\begin{array}{l} A \to aA \mid bB \\ B \to bB \mid aC \mid a \\ C \to aA \mid bB \end{array}\right] \quad G \text{ è regolare}$$



Non è immediato riconoscere che $L[N] = (a|b)^* ba$

Teorema Data una grammatica regolare G si può costruire un NFA N_G equivalente.

Dimostrazione Sia $G = (NT, T, R, S)$, allora $N_G = (T, Q, \delta, S, \{\varepsilon\})$ è definito come segue

- $Q = NT \cup \{\varepsilon\}$ ($\cdot$ $F = \{\varepsilon\}$ $\cdot$ $q_0 = S$)
- δ è definita come:
  - $Z \in \delta(V, a)$ se $V \to aZ \in R$
  - $\varepsilon \in \delta(V, a)$ se $V \to a \in R$
  - $\varepsilon \in \delta(S, \varepsilon)$ se $S \to \varepsilon \in R$

Si può dimostrare che

$$S \overset{*}{\underset{G}{\Longrightarrow}} w \quad \text{(con la grammatica } G\text{)}$$

sse

$$(S, w) \vdash^{*}_{N_G} (\varepsilon, \varepsilon) \quad \text{(con l'automa } N_G\text{)}$$

(non lo dimostreremo)

Da DFA a grammatiche regolari

Teorema

Da un DFA M, possiamo definire una grammatica regolare G_M tale che $L[M] = L(G_M)$.

Dimostrazione

Sia $M = (\Sigma, Q, \delta, q_0, F)$ il DFA. La grammatica $G_M = (Q, \Sigma, R, q_0)$ ha:

- per nonterminali gli stati di M
- per terminali, l'alfabeto di M
- per simbolo iniziale, lo stato iniziale di M
- per produzioni R
  - per ogni $\delta(q_i, a) = q_j$, la produzione $q_i \to a q_j \in R$

    inoltre se $q_j \in F$, anche $q_i \to a \in R$
  - se $q_0 \in F$, allora $q_0 \to \varepsilon \in R$

(Versione alternativa che usa la def. di grammatica regolare più lasca:

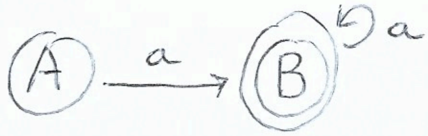
- per ogni $\delta(q_i, a) = q_j$ la produzione $q_i \to a q_j \in R$
- se $q \in F$, allora $q \to \varepsilon \in R$)

Si può dimostrare che

$$w \in L[M] \text{ sse } w \in L(G_M)$$

Esempi

M



aa^* è l'espressione regolare che descrive L[M]

Secondo la costruzione appena descritta

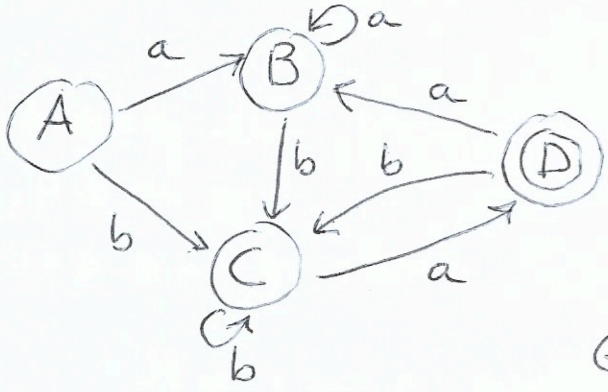
$$G_M \left[\begin{array}{l} A \to aB \mid a \\ B \to aB \mid a \end{array}\right.$$

Secondo la variante, invece

$$G_M \left[\begin{array}{l} A \to aB \\ B \to aB \mid \varepsilon \end{array}\right.$$

a rigore questa non è regolare perché ammette $B \to \varepsilon$

M



$(a|b)^* ba$

$$G_M \left[\begin{array}{l} A \to aB \mid bC \\ B \to aB \mid bC \\ C \to bC \mid aD \mid a \\ D \to aB \mid bC \end{array}\right.$$

Alternativamente

$$G_M \left[\begin{array}{l} A \to aB \mid bC \\ B \to aB \mid bC \\ C \to bC \mid aD \\ D \to aB \mid bC \mid \varepsilon \end{array}\right.$$

Domanda: È possibile costruire una G_M regolare partendo da un NFA (anziché da un DFA)?

Suggerimento: pensate alle transizioni ε.

Grammatiche Regolari ed Espr. Regolari

Teorema Il linguaggio definito da una grammatica regolare G è un linguaggio regolare, cioè è possibile costruire una espressione regolare s_G tale che

$$L(G) = \mathcal{L}[s_G].$$

Sketch della dimostrazione

Idea della prova:

Caso semplice: un solo nonterminale

$$A \rightarrow aA \mid b \mid \varepsilon$$

È intuitivo vedere che $a^*(b|\varepsilon)$ è la espressione regolare associata.

Caso medio: due nonterminali

$$\left.\begin{array}{l} A \rightarrow aA \mid bB \mid c \\ B \rightarrow cA \mid aB \mid d \end{array}\right\}$$ lo si può vedere come un sistema da risolvere

Ricaviamo B dalla seconda "equazione"

$$B \approx a^*(cA \mid d)$$ dove A compare nella espr. regolare

Ora sostituiamo B nella prima "equazione"

$$A \approx aA \mid ba^*(cA \mid d) \mid c$$

Con opportune manipolazioni su espr. regolari, usando leggi che abbiamo visto, possiamo scrivere

$$A \approx aA \mid ba^*cA \mid ba^*d \mid c$$

e quindi

$$A \approx (a \mid ba^*c)A \mid ba^*d \mid c$$

Ora siamo nella forma "<u>semplice</u>" e sappiamo come fare

A ha associata la esp. regolare

$$(a \mid ba^*c)^* (ba^*d \mid c)$$

<u>In generale</u>

$$A_1 \approx a_{11} A_1 \mid \dots \mid \cancel{A_1} a_{1m} A_m \mid b_{11} \mid \dots \mid b_{1p_1}$$

$$A_2 \approx a_{21} A_1 \mid \dots \mid a_{2n} A_n \mid b_{21} \mid \dots \mid b_{2p_2}$$

$$\vdots$$

$$A_m \approx a_{n1} A_1 \mid \dots \mid a_{mn} A_n \mid b_{n1} \mid \dots \mid b_{n p_n}$$

Si parte con

$$A_m \simeq S_m[A_1, \dots, A_{n-1}]$$

cioè si costruisce una espr. regolare per A_n che usa $A_1 \dots, A_{n-1}$

Poi si procede sostituendo A_n (o meglio $S_n[A_1, \dots A_n]$ al posto di A_n) nell'equazione per A_{n-1}, cioè

$$A_{n-1} \simeq S_{n-1}[A_1, \dots, A_{n-2}]$$

e così via fino ad arrivare ad A_1 (che è il simbolo iniziale)

Esempi

(1) $$\left.\begin{array}{l} A \to aB|\varepsilon \\ B \to bA|\varepsilon \end{array}\right] G$$

- Per B la espr. regolare associata è $(bA|\varepsilon)$
- Sostituisco questa al posto di B nella prima "equazione"

$$A \simeq a(bA|\varepsilon)|\varepsilon$$

- manipolo la espr. regolare

$$A \simeq abA|a|\varepsilon$$

- siamo ora nel caso semplice

$$A \simeq (ab)^*(a|\varepsilon)_{(S_G)}$$

---

(2) $$\left.\begin{array}{l} A \to aB|bA|d \\ B \to aB|c \end{array}\right] G$$

- Per B ottengo subito l'espr. regolare a^*c
- Sostituisco nella prima equazione

$$A \simeq a(a^*c)|bA|d$$

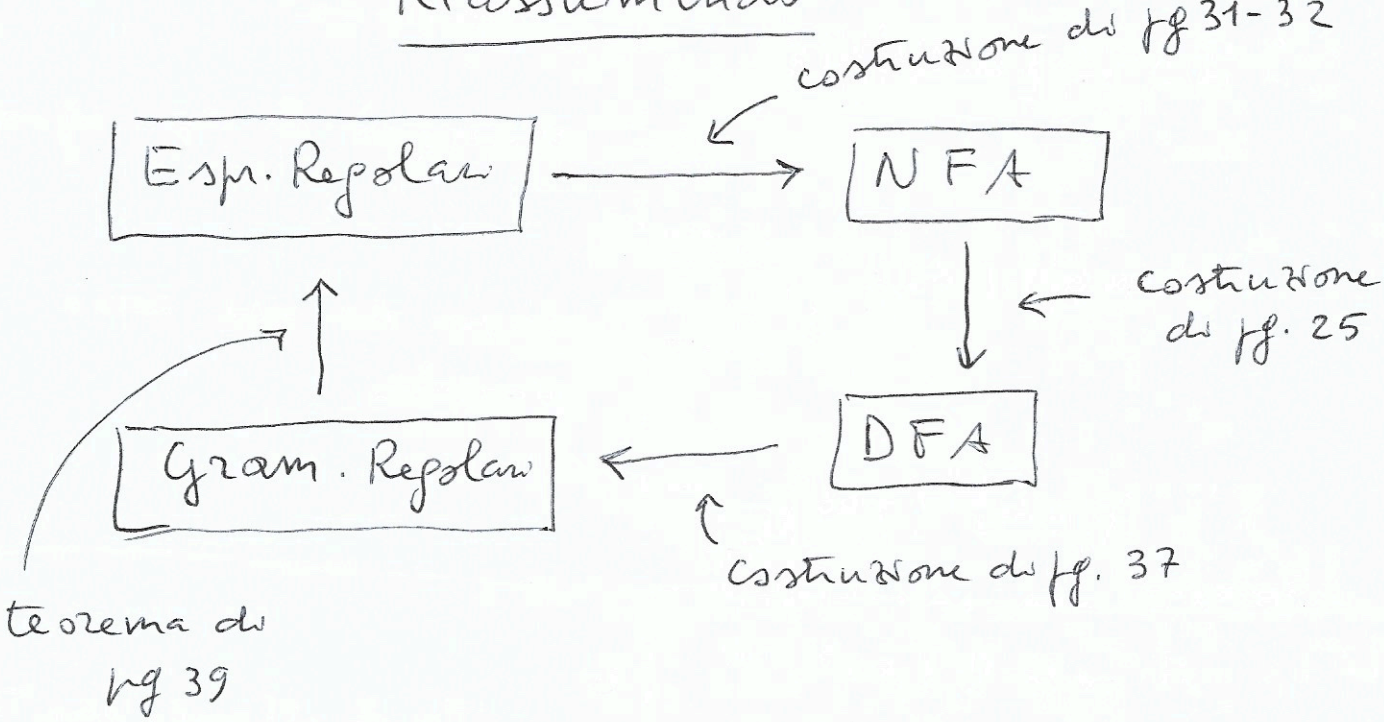
- ora siamo nel caso semplice

$$A \simeq b^*(a(a^*c)|d)_{(S_G)}$$

OSS: La grammatica regolare con unica produzione $A \to aA$] G definisce il linguaggio vuoto, non a^*

Quindi $s_G = \emptyset$.

Riassumendo



Conseguenza: Tutti questi formalismi sono equivalenti.

Tutti generano / riconoscono / descrivono

gram. reg. ↑ generano, NFA/DFA ↑ riconoscono, esp. reg. ↑ descrivono

la stessa classe di linguaggi, ovvero i linguaggi REGOLARI

Per costruire uno scanner (analizzatore lessicale), si parte dalla specifica dei pattern associati alle categorie sintattiche del linguaggio, mediante espressioni regolari

Expr. Reg → NFA → DFA
(pattern)

↑ ma se l'NFA ha n stati, il DFA potrebbe avere fino a 2^n stati!

⇒ serve trovare un DFA equivalente più piccolo possibile

min-DFA (minimizzato)

- per occupare meno memoria (minor numero possibile di stati)
- tale DFA minimo è unico (a meno di isomorfismo)
- ed è equivalente al DFA da cui siamo partiti