AUTOMI CON ε-TRANSIZIONI

un NFA con ε-transizioni e' una quintupla $\langle Q, \Sigma, \delta, q_0, F \rangle$, dove la funzione di transizione e' ora definita

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow P(Q)$$

dunque, e' possibile passare da uno stato all'altro anche senza "leggere" caratteri di input.

La definizione $\hat{\delta} : Q \times \Sigma^* \rightarrow P(Q)$ introduce la funzione ε-closure (= eseguire un certo numero di ε-step)

→ applicata ad uno stato, restituisce l'insieme degli stati raggiungibili da esso (compreso sè stesso) mediante ε-transizioni.

La costruzione e' equivalente a quella che permette di conoscere i nodi raggiungibili da un nodo in un grafo, si può calcolare a partire da δ

(un arco $p \rightarrow q$ si ha quando $q \in \delta(p, \varepsilon)$)

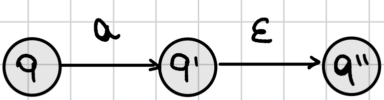
$$\varepsilon\text{-closure}(P) = \bigcup_{p \in P} \varepsilon\text{-closure}(p)$$

$\hat{\delta}$ si definisce come:

$$\begin{cases} \hat{\delta}(q, \varepsilon) = \varepsilon\text{-closure}(q) \\ \hat{\delta}(q, wa) = \bigcup_{p \in \hat{\delta}(q, w)} \varepsilon\text{-closure}(\delta(p, a)) \end{cases}$$

In questo caso $\hat{\delta}(q, a)$ può essere diverso da $\delta(q, a)$.

Esempio:



$$\delta(q, a) = \{q'\}$$

$$\hat{\delta}(q, a) = \bigcup_{p \in \hat{\delta}(q, \varepsilon)} \varepsilon\text{-closure}(\delta(p, a))$$

$= \{q', q''\}$

Definiamo il linguaggio accettato dall'automa

$$L(M) = \{ x \in \Sigma^* : \hat{\delta}(q_0, x) \cap F \neq \emptyset \}$$

La definizione di $\hat{\delta} : Q \times \Sigma^* \rightarrow p(Q)$ fa riferimento alle funzioni ausiliare

$\rightarrow$ $\varepsilon\text{-step} : \underbrace{p(Q)}_{\text{insieme di stati nel quale mi trovo}} \xrightarrow{\text{con un } \varepsilon\text{-step}} \underbrace{p(Q)}_{\text{posso andare in un altro insieme di stati}}$ (tipo funzione)

$\rightarrow$ $\varepsilon\text{-step}(S) = \{ q \in Q \mid \exists p \in S .\ q \in \delta(p, \varepsilon) \}$

dove posso arrivare partendo da S con <u>un</u> passo ε

$\rightarrow$ $\varepsilon\text{-step}^0(S) = S;$

(partendo da un insieme S faccio 0 ε-step rimango in S)

$\rightarrow$ $\varepsilon\text{-step}^{n+1}(S) = \varepsilon\text{-step}(\varepsilon\text{-step}^n(S))$

Abbiamo due varianti per l'ε-closure:

1) $\varepsilon\text{-closure} : Q \rightarrow p(Q)$ (:= lavora su insieme di stati

$$\varepsilon\text{-closure}(S) = \bigcup_{i \in \mathbb{N}} \varepsilon\text{-step}^i(S)$$

2) $\varepsilon\text{-closure} : p(Q) \rightarrow p(Q)$ (:= lavora su un singolo stato)

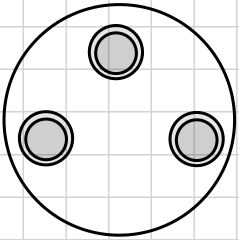
$$\varepsilon\text{-closure}(p) = \varepsilon\text{-closure}(\{p\})$$

OSS.

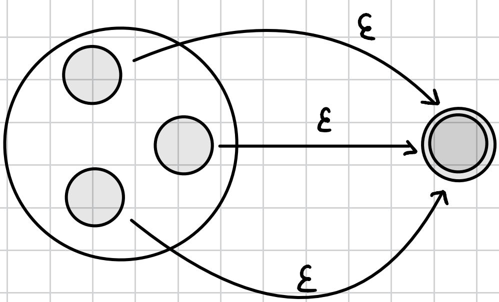
Per gli ε-NFA si potrebbe assumere che l'insieme F abbia esattamente un elemento.

OVVERO:

Supponiamo di avere un automa con 3 stati finali



Per ottenere un ε-NFA equivalente basta aggiungere un nuovo stato finale, trasformare gli stati finali precedenti in "non-finali" e collegarli al nuovo stato con una ε-transizione.



ESERCIZIO

Concatenazione di due linguaggi L_1 e L_2:

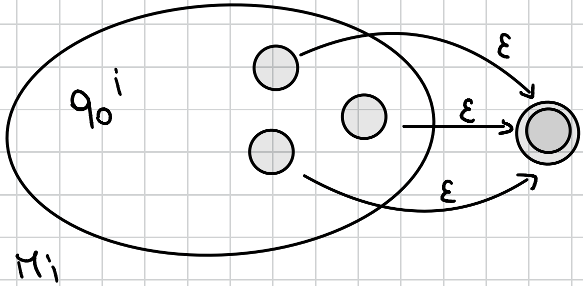
$$L_1L_2 = \{ w_1w_2 \in \Sigma^* \mid w_1 \in L_1 , w_2 \in L_2 \}$$

Se L_1 e L_2 sono regolari, cosa possiamo dire della loro concatenazione?

$\longrightarrow$ Se L_1 e L_2 sono regolari $\Rightarrow \exists M_1, M_2$ t.c. $L(M_i) = L_i$
$i = 1,2$

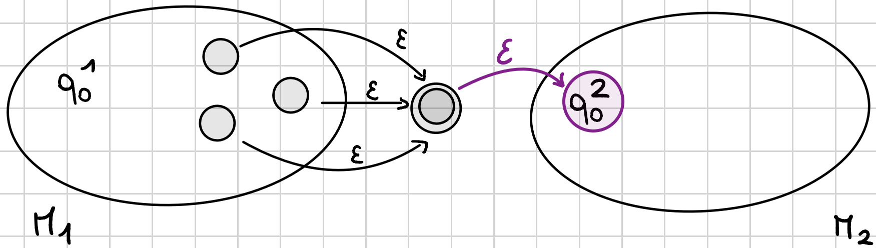
M_1, M_2 DFA

Supponendo che M_i abbia un certo numero di stati finali, possiamo trasformarlo in un ε-NFA:



$(i = 1, 2)$

La concatenazione allora potrà essere realizzata così:



Aggiungo un ARCO in modo che dopo aver finito con la stringa w_1, posso saltare con passo ε all'inizio dell'automa M_2 per riconoscere la stringa w_2.

Per affermare che $L_1 L_2$ e' regolare dobbiamo dimostrare che dato un qualunque ε-NFA lo possiamo tradurre in un NFA (che può' essere poi convertito in un DFA).

EQUIVALENZA DI ε-NFA e NFA

Ogni NFA e', per def., un caso particolare di un ε-NFA

TEOREMA

Sia $M = < Q, \Sigma, \delta, q_0, F >$ un ε-NFA. Allora esiste un NFA M' tale che $L(M) = L(M')$

DIMOSTRAZIONE: Definisco $M' = < Q', \Sigma', \delta', q_0', F' >$ come:

- $Q' = Q$ (non serve aggiungere nuovi stati)
- $\Sigma' = \Sigma$
- $q_0' = q_0$

$\bullet\ F' = \begin{cases} F \cup \{q_0\}, & \text{se } \varepsilon\text{-closure}(q_0) \cap F \neq \emptyset \\ F, & \text{altrimenti} \end{cases}$

L'insieme degli stati finali potrebbe cambiare:
- se partendo da q_0 e seguendo soltanto ε-transizioni, posso già arrivare ad uno stato accettante, allora q_0 dovrebbe accettare la stringa vuota

=> ma nell'automa che sto costruendo non sono possibili le ε-transizioni, dunque se la ε-closure(q_0) mi fa entrare in uno stato accettante q_0 deve entrare negli stati finali.

$\bullet\ \underbrace{\delta'(q,a)}_{} = \underbrace{\hat{\delta}(q,a)}_{}$

f. di transizione che prende un solo simbolo (del NFA) ⇓ ci dice quali sono gli archi dell'automa | f. di transizione che lavora sulle stringhe che prende la stringa formata dal simbolo "a" (f. che include l'ε-closure)

Dobbiamo dimostrare che

$$\hat{\delta}(q_0, x) \cap F \neq \emptyset \iff \hat{\delta}'(q_0', x) \cap F' \neq \emptyset$$

ε-NFA == NFA

Vediamo i casi: 1) $x = \varepsilon$
2) $x \neq \varepsilon$

1) Se $x = \varepsilon$

$\rightarrow \hat{\delta}(q_0, \varepsilon) = \varepsilon\text{-closure}(q_0)$ e $\hat{\delta}'(q_0', \varepsilon) = \{q_0\}$ per def.

non leggiamo nessun simbolo e restiamo in q_0

$(\Rightarrow)$ $\rightarrow$ Se ε-closure$(q_0) \cap F \neq \emptyset$ allora, per def. di F' vale che $q_0 \in F'$; dunque $\{q_0\} \cap F' \neq \emptyset$

$(\Leftarrow)$ $\rightarrow$ Sia ora $\{q_0\} \cap F' \neq \emptyset$. Allora $q_0 \in F'$. Due casi possibili:

1. Se $q_0 \in F$ allora ε-closure$(q_0) \cap F \neq \emptyset$ (in quanto $q_0 \in \varepsilon$-closure(q_0))

2. Altrimenti, se $q_0 \in F' \setminus F$ (ovvero e' stato aggiunto : $F \cup \{q_0\}$) per definizione di F' si ha che ε-closure$(q_0) \cap F \neq 0$

2) Se $x \neq \varepsilon$

Dim. per induzione su $|x| \geq 1$, che $\hat{\delta}'(q_0', x) = \hat{\delta}(q_0, x)$

<u>BASE</u>:

$|x| = 1$. Allora $x = a$ per qualche simbolo $a \in \Sigma$. Ma allora $\delta'(q_0', a) = \hat{\delta}(q_0, a)$ per definizione

<u>PASSO INDUTTIVO</u>:

Assumiamo che la tesi valga per tutte le stringhe x tali che $1 \leq |x| \leq m$.

Sia xa una stringa $m+1$. Allora:

$$\hat{\delta}'(q_0', xa) = \bigcup_{p \in \hat{\delta}'(q_0', x)} \delta'(p, a) \qquad \text{def. di } \hat{\delta}'$$

$$= \bigcup_{p \in \hat{\delta}'(q_0', x)} \hat{\delta}(p, a) \qquad \text{def di } \delta'$$

$$= \bigcup_{p \in \hat{\delta}(q_0, x)} \hat{\delta}(p, a) \qquad \text{ip. ind.}$$

$$\text{def. di } \hat{\delta} \qquad = \bigcup_{p \in \hat{\delta}(q_0, x)} \bigcup_{r \in \hat{\delta}(p, \varepsilon)} \varepsilon\text{-closure}(\delta(r, a))$$

$$\hat{\delta}(q_0, x) \text{ chiuso per } \varepsilon\text{-closure} \qquad = \bigcup_{p \in \hat{\delta}(q_0, x)} \varepsilon\text{-closure}(\delta(p, a))$$

$= \hat{\delta}(q_0, xa)$ def. di $\hat{\delta}$

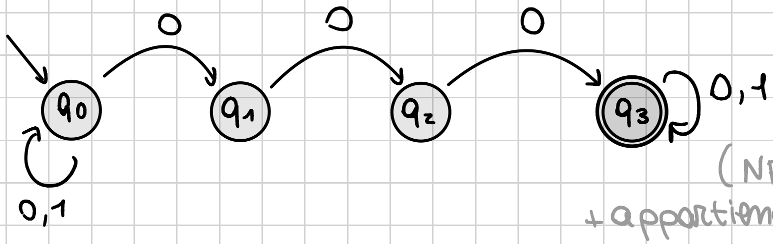
COROLLARIO 1

Le classi di linguaggio riconosciute da DFA, NFA e ε-NFA coincidono (linguaggi regolari)

ESERCIZIO

$\Sigma = \{0,1\}$

1) Insieme di tutte le stringhe che hanno 3 zero consecutivi



(NFA)
+ appartiene all'insieme degli ε-NFA

TABELLA DI TRANSIZIONE per ε-NFA

δ	0	1	ε
q_0	$\{q_0, q_1\}$	q_0	$\emptyset$
q_1	q_2	$\emptyset$	$\emptyset$
q_2	q_3	$\emptyset$	$\emptyset$
q_3	q_3	q_3	$\emptyset$

(non fa parte di Σ ma c'è come input di δ)

DFA:

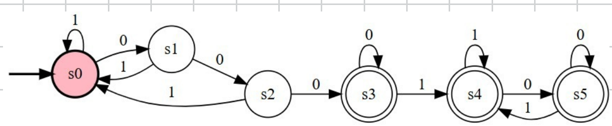
- STATO $\{q_0\}$:
  0: $\{q_0, q_1\}$ 1: $\{q_0\}$

- STATO $\{q_0, q_1\}$
  $0: \{q_0, q_1, q_2\}$ $1: \{q_0\}$
- STATO $\{q_0, q_1, q_2\}$
  $0: \{q_0, q_1, q_2, q_3\}$ $1: \{q_0\}$
- STATO $\{q_0, q_1, q_2, q_3\}$
  $0: \{q_0, q_1, q_2, q_3\}$ $1: \{q_0, q_3\}$
- STATO $\{q_0, q_3\}$
  $0: \{q_0, q_1, q_3\}$ $1: \{q_0, q_3\}$
- STATO $\{q_0, q_1, q_3\}$
  $0: \{q_0, q_1, q_3\}$ $1: \{q_0, q_3\}$

$S_0 = \{q_0\}$, $S_1 = \{q_0, q_1\}$, $S_2 = \{q_0, q_1, q_2\}$

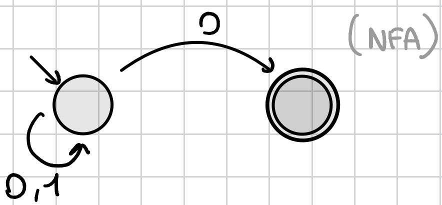
$S_3 = \{q_0, q_1, q_2, q_3\}$ $S_4 = \{q_0, q_3\}$ $S_5 = \{q_0, q_1, q_3\}$

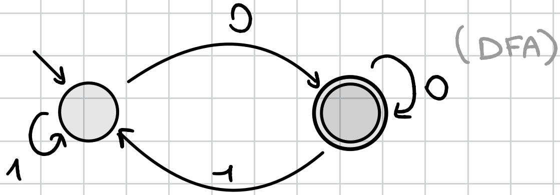
	0	1
S_0	S_1	S_0
S_1	S_2	S_0
S_2	S_3	S_0
S_3	S_3	S_4
S_4	S_5	S_4
S_5	S_5	S_4



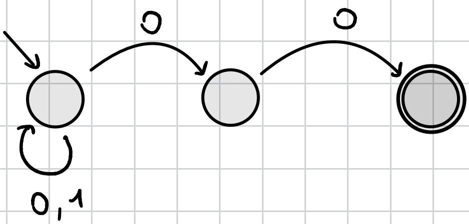
2) l'insieme di tutte le stringhe tali che se interpretate come numero intero binario sono divisibili per 2.
(ovvero finisce con zero)

$L = \{ w0 \mid w \in \Sigma^* \}$

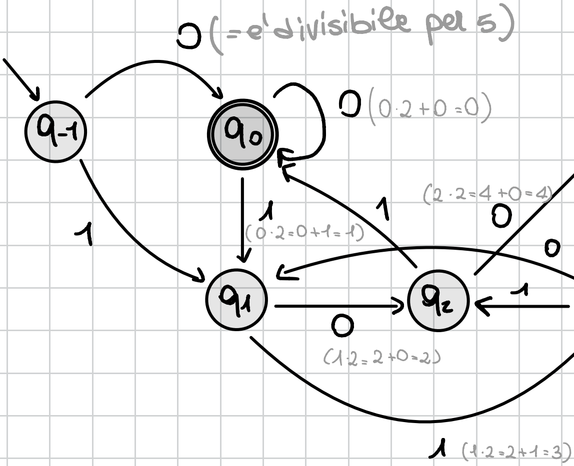




3) l'insieme di tutte le stringhe tali che se interpretate come numero binario sono divisibili per 4.



4) l'insieme di tutte le stringhe tali che se interpretate come numero binario sono divisibili per 5.
(ragioniamo modulo 5)



RAPPRESENTAZIONE DEI LINGUAGGI

Problema della rappresentazione FINITA di linguaggi INFINITI e' affrontabile da 3 punti di vista:

1) RICONOSCITIVO ANALITICO: dove il linguaggio e' visto come l'insieme delle stringhe riconosciute o accettate da strutture finite (AUTOMI)

2) GENERATIVO-SINTETICO: il linguaggio e' visto come l'insieme delle stringhe generate da strutture finite dette grammatiche

3) ALGEBRICO: il linguaggio e' rappresentato da un' espressione algebrica o e' la soluzione di un sistema di relazioni algebriche.

OPERAZIONI SUI LINGUAGGI

Sia Σ un alfabeto e L, L_1, L_2 insiemi di stringhe di Σ^*.

→ La concatenazione di L_1 e L_2 e' l'insieme:

$$L_1 L_2 = \{ xy \in \Sigma^* \mid x \in L_1, y \in L_2 \}$$

→ La chiusura (di Kleene) di L, e' l'insieme:

$$L^* = \bigcup_{i \geq 0} L^i \quad \text{dove} \quad L^0 = \{\varepsilon\}, \quad L^{i+1} = L L^i$$

→ La chiusura positiva di L, e' l'insieme:

$$L^+ = \bigcup_{i \geq 1} L^i$$

NOTA = $L^+ = LL^*$
(zucchero sintattico)

ESPRESSIONI REGOLARI

Sia Σ un alfabeto. Le espressioni regolari su Σ e gli insiemi che esse denotano sono definiti ricorsivamente come:

(differenza tra sintassi e semantica)

1) $\emptyset$ e' una espressione regolare che denota l'insieme vuoto (es. SINTASSI $\emptyset$ SEMANTICA $\{\}$)

2) ε e' una espressione regolare che denota l'insieme $\{\varepsilon\}$ (es. SIN: ε SEM: $\{\varepsilon\}$)

3) $\forall$ simbolo $a \in \Sigma$, a e' una espressione regolare che denota l'insieme $\{a\}$

4) Se r e s sono espressioni regolari denotanti rispettivamente gli insiemi R ed S, allora $(r+s)$, (rs), e (r^*) sono espressioni regolari che denotano gli insiemi $R \cup S$, RS, e R^* rispettivamente

Se r è un'espressione regolare, indicheremo $L(r)$ il linguaggio denotato da r.