Automi a stati finiti

- Il modello: la definizione formale, esempi.
- Le definizioni utili per descrivere e provare proprietà degli automi: configurazioni, relazione “porta a” e relative definizioni di linguaggio accettato.

Prime definizioni e convenzioni

Σ = {*a,b,c*} è un esempio di alfabeto, che è semplicemente un insieme finito.

x = *abbaacba* è un esempio di parola, che è semplicemente una sequenza di lettere dell'alfabeto

ε (o λ) è la parola vuota, senza lettere

Σ^* = {ε,*a,b,c,aa,ab,ac,ba,bb,bc,*…} è l'insieme di tutte le parole che si possono costruire sull'alfabeto {*a,b,c*}

Prime definizioni e convenzioni - 2

$|x|$ = lunghezza di x, cioè il numero dei simboli della parola

Definizione ricorsiva di parola:

ε è una parola

se x è una parola e *a* è una lettera dell'alfabeto

allora xa è una parola

Definizione ricorsiva di lunghezza:

$|\varepsilon| = 0$

$|xa| = |x| + 1$, per x in Σ^* e a in Σ.

Concatenazione e potenza di parole

Concatenazione di due parole:

date $x = a1a2...an$ e $y = b1b2...bm$ in Σ^*, $n,m \geq 0$

$xy = a1a2...anb1b2...bm$ è la loro concatenazione

Definizione ricorsiva di concatenazione:

per x,y in Σ^* e a in Σ

$x\varepsilon = x$

$x(ya) = (xy)a$

Potenza di una parola

per x in Σ^* $x^n = xx...x$ (n volte)

Definizione ricorsiva di potenza:

$x^0 = \varepsilon$

$x^{n+1} = x^n x$ per $n \geq 0$

Concatenazione di linguaggi

Estensione della concatenazione ai linguaggi:

$L, L' \subseteq \Sigma^*$

$LL' = \{w \text{ in } \Sigma^* \mid \exists\, x \text{ in } L,\ y \text{ in } L' \text{ e } w=xy\}$

Esempi:

bab è già presente

1. Se L = {a, ab,ba}, L' = {ab,b}, allora LL' = {aab,ab,abab,abb,baab,bab}.

2. Se L = {b,ba}, L' = {ab,b}, allora LL' = {bab,bb,baab}.

Proprietà:

$\{\varepsilon\}L = L = L\{\varepsilon\}$

$\emptyset L = \emptyset = L\emptyset$

Potenza di linguaggi

Estensione della potenza ai linguaggi:

per $L \subseteq \Sigma^*$ $L^n = LL...L$ (n volte)

$L^0 = \{\varepsilon\}$

Esempi:

1. se L = {a,ab,ba} allora
   L^2 = {aa,aab,aba, abab,abba,baa,baab,baba}.

aba

2. Se L = {ab,b} allora
   L^3 = {ababab,abbab,babab,bbab,ababb,abbb,
babb,bbb}.

Stella di Kleene

$$L \subseteq \Sigma^*, \quad L^* = \bigcup_{n \geq 0} L^n$$

Quindi $L^* = \{\varepsilon\} \cup L \cup LL \ldots$

Esempi:

1. $\{a,b\}^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, abb, baa, \ldots\}$

2. se $L=\{a,ab,ba\}$,
alloraL*=$\{\varepsilon, a, ab, ba, aa, aab, aba, abab, abba, baa, baab, baba, \ldots\}$.

Stephen Cole Kleene (1909 - 1994)

Ha insegnato alla Wisconsin University.

La sua attività di ricerca riguarda la teoria degli algoritmi, i modelli di calcolo e le funzioni ricorsive. A lui si devono alcuni dei più importanti risultati della teoria degli automi.

Altre operazioni sui linguaggi

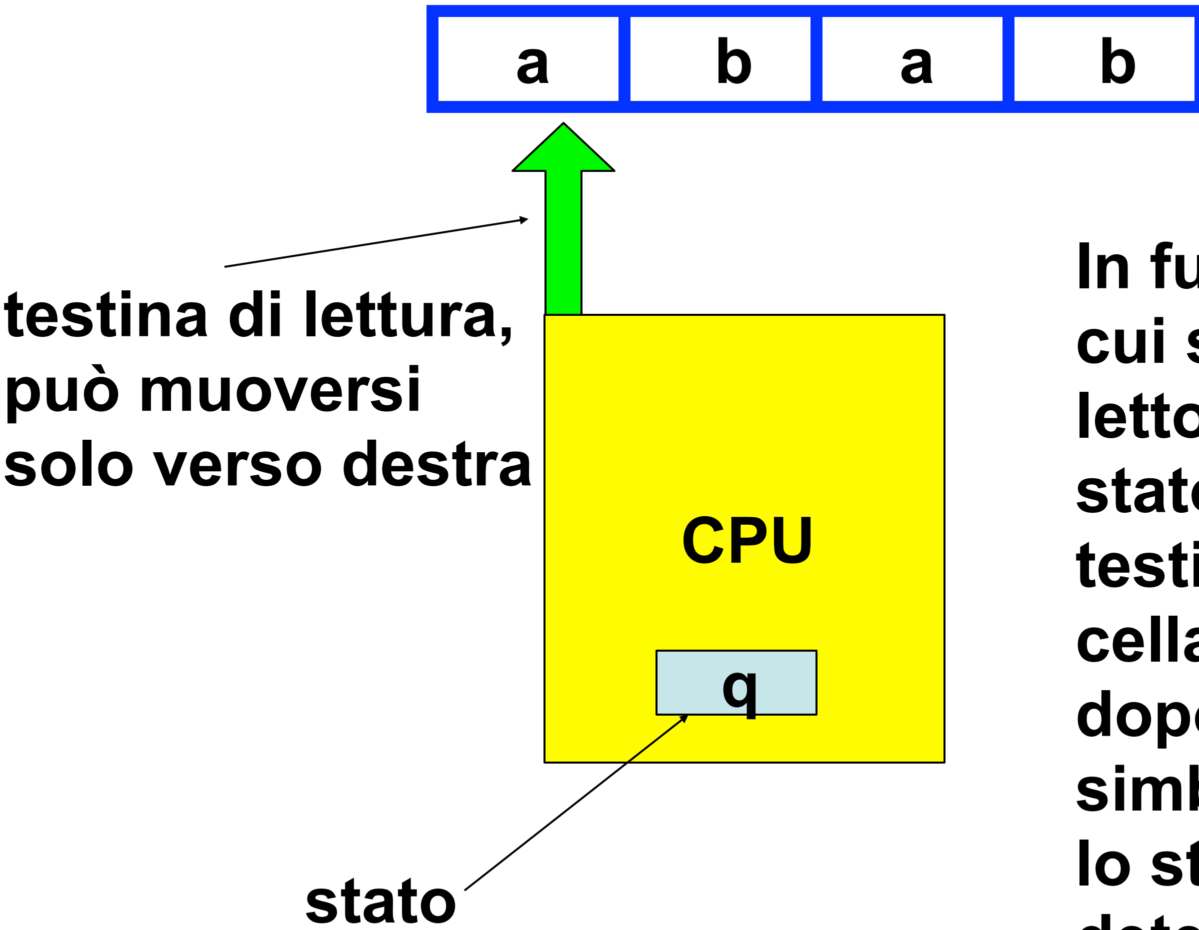
Ricordiamo le operazioni insiemistiche:

$L, L' \subseteq \Sigma^*$

- **unione: $L \cup L'$**
- **intersezione: $L \cap L'$**
- **complemento: $\neg L = \Sigma^* - L$**

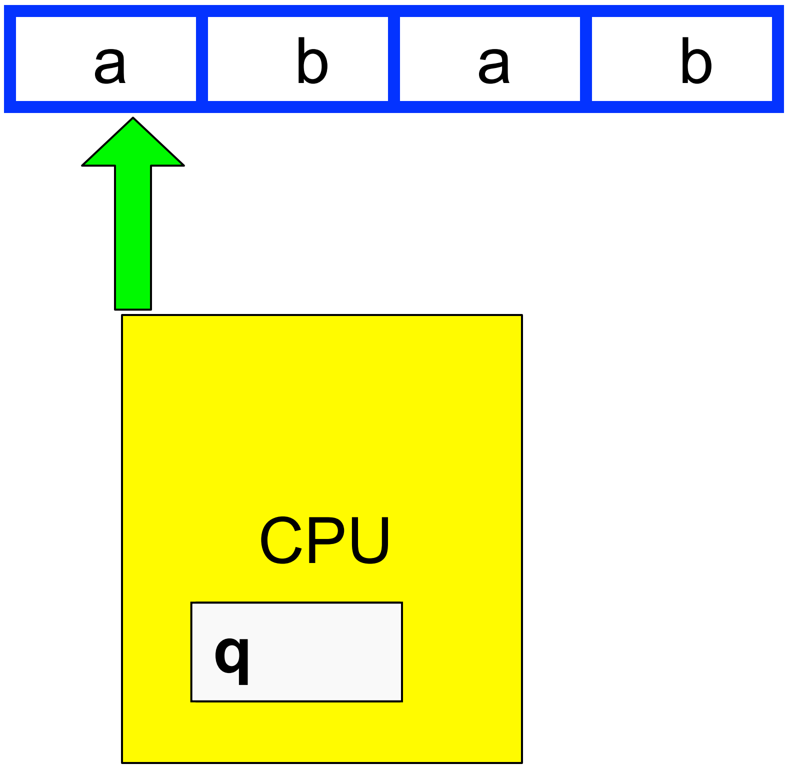
- **Ricordiamo che queste operazioni non sono indipendenti:**
- **$\neg(X \cup X') = \neg X \cap \neg X'$**
- **$\neg(X \cap X') = \neg X \cup \neg X'$**

Il MODELLO MENTALE



In funzione dello stato in cui si trova e dell'input letto la macchina cambia stato, poi sposta la testina di lettura sulla cella successiva. Termina dopo aver letto l'ultimo simbolo sul nastro input; lo stato raggiunto determina se la parola è accettata o no.

Un esempio di esecuzione



La macchina incorpora un programma con istruzioni del tipo:

se nello stato q e leggi a vai nello stato p

se nello stato p e leggi b vai nello stato q ...

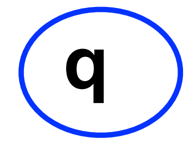
A ogni passo la testina di lettura si sposta di una cella a destra.

La macchina si ferma quando ha letto tutto l'input.

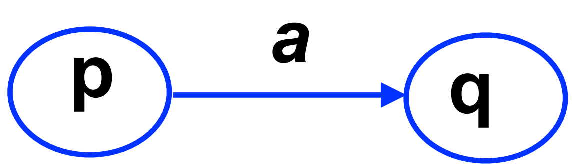
DIAGRAMMA DEGLI STATI

È un grafo diretto in cui

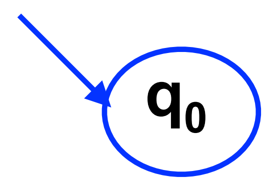
- **ogni nodo corrisponde a uno stato**



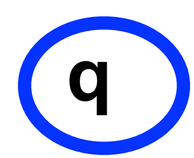
- **tra una coppia di nodi corrispondenti agli stati p e q c'è un arco con etichetta a se c'è la regola se nello stato p e leggi *a* vai nello stato q**



- **lo stato iniziale q_0 è denotato**

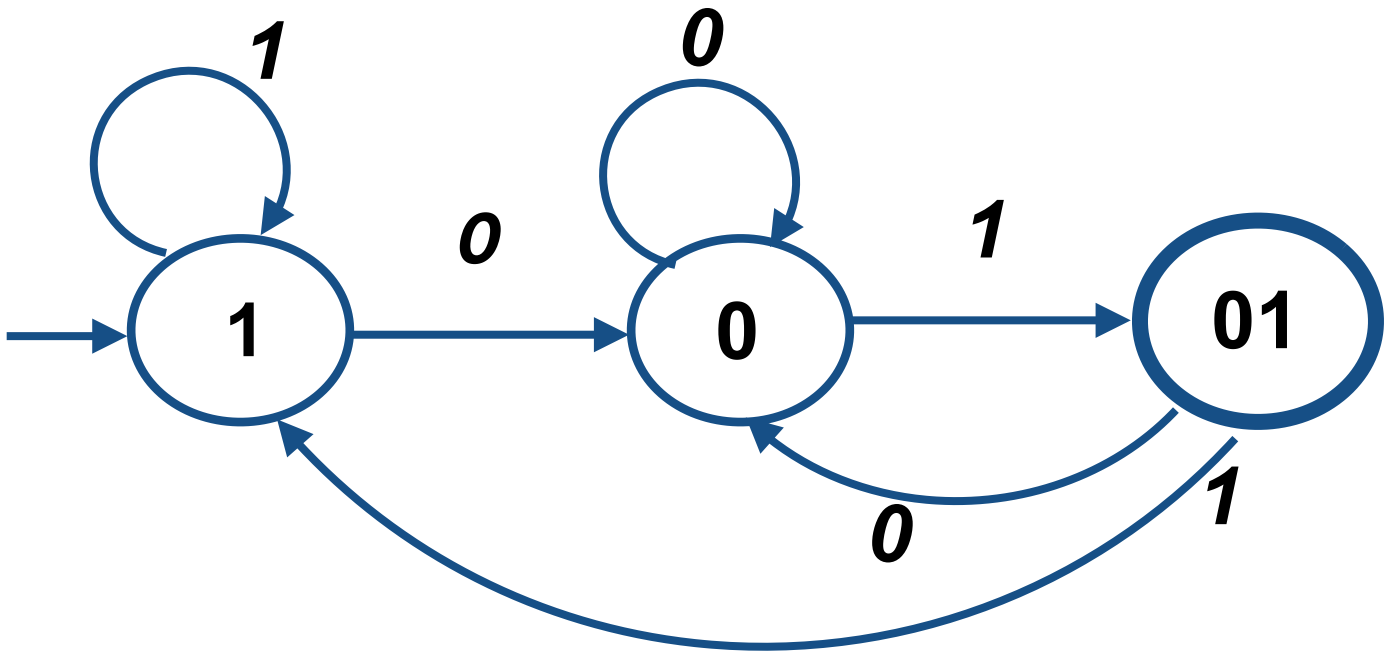


- **uno stato finale q è denotato**



Esempio 1

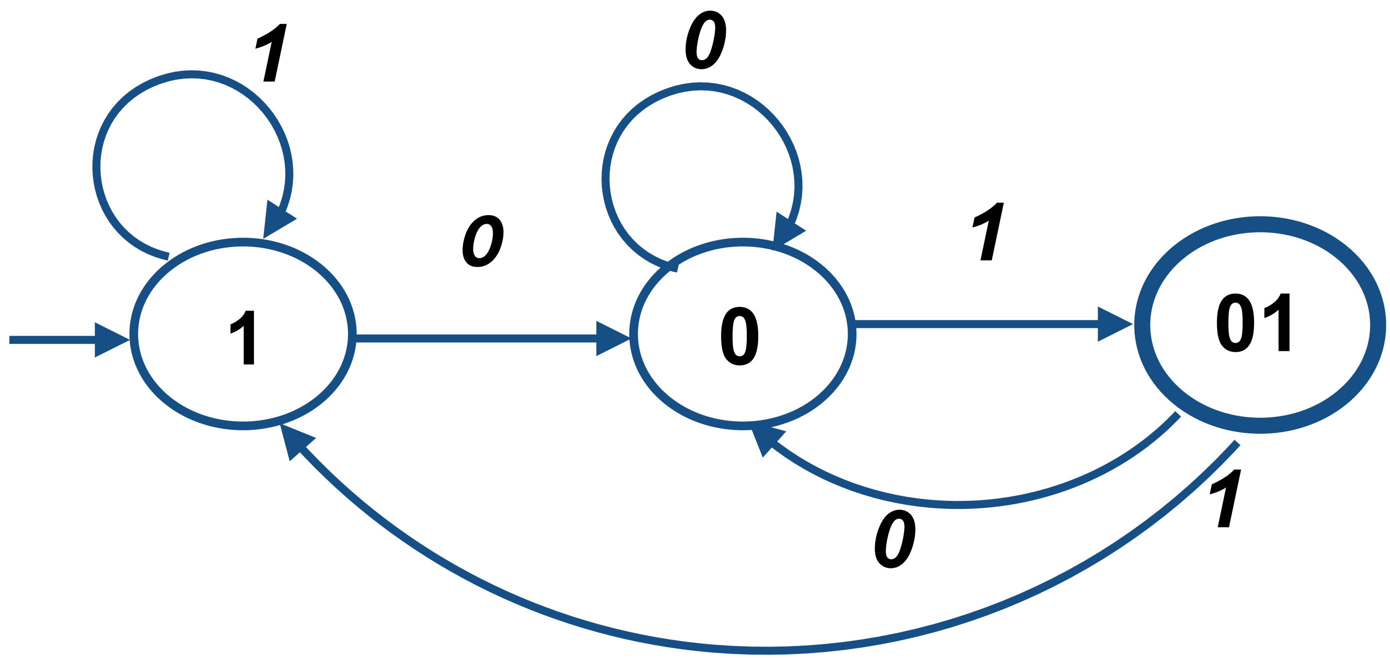
Vogliamo costruire un DFA che accetta tutte le stringhe binarie che terminano con 01



Ogni stringa che determina un cammino sul grafo diretto dallo stato iniziale a quello finale è accettata dall'automa

Esempio 1

Ogni stringa binaria w determina un cammino di lunghezza |w|+1, la lunghezza di w più 1



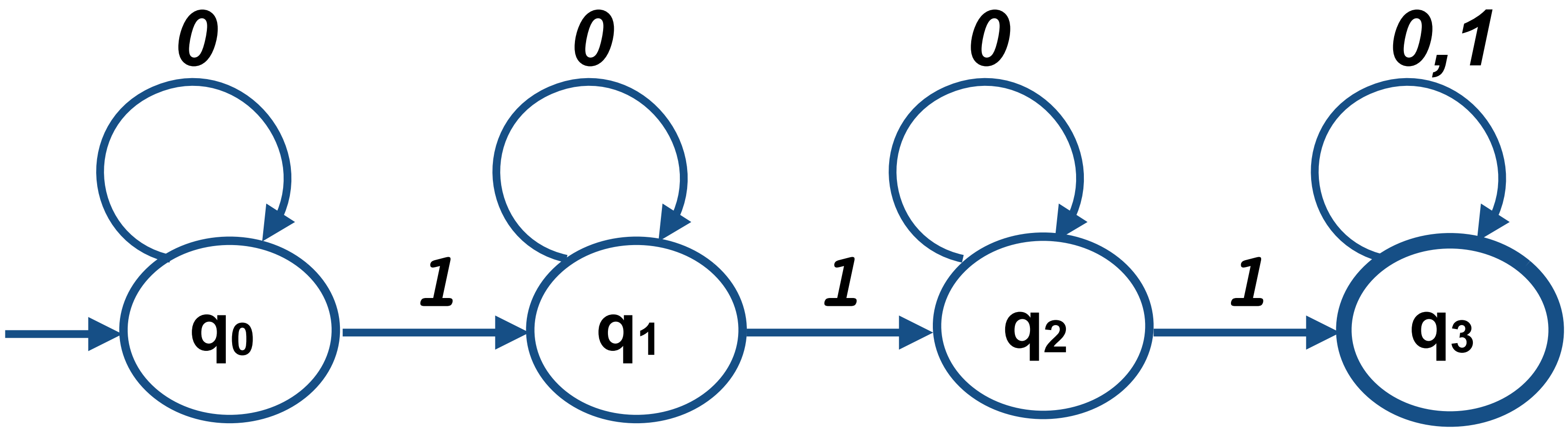
Definizione ricorsiva di lunghezza:

$|\varepsilon| = 0$

$|wa| = |w|+1$, per a in $\{0,1\}$

Esempio 2

Vogliamo costruire un DFA che accetta tutte le stringhe binarie che contengono almeno tre 1



la stringa

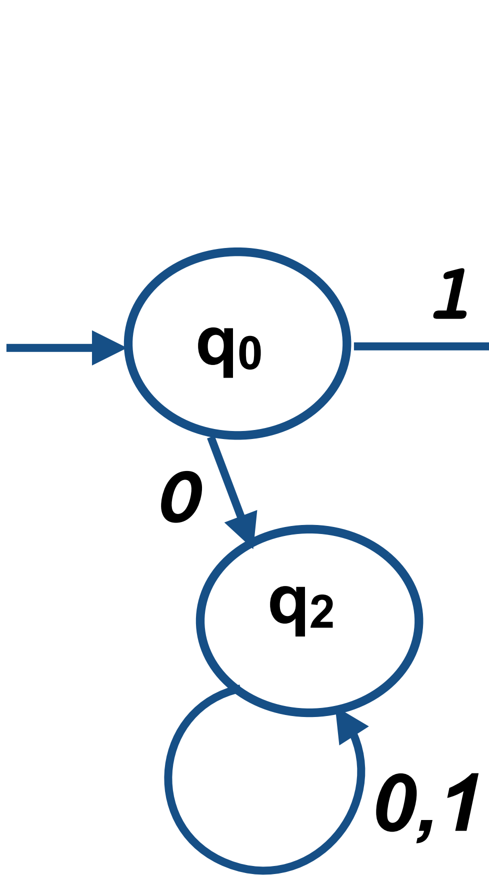
0 0 1 0 1 1 0 0 determina il cammino

q_0 q_0 q_0 q_1 q_1 q_2 q_3 q_3 q_3

Ci sono tanti archi uscenti da ogni nodo quante sono le lettere dell'alfabeto

Esempio 3

Vogliamo costruire un DFA che accetta tutte le stringhe binarie che cominciano per 1



E la parola 010?

Così manteniamo la proprietà che ogni parola determina un cammino sul grafo, proprietà comoda nelle costruzioni

Ma normalmente non si disegna questa parte dell'automa

Modello formale

Un **automa a stati finiti deterministico**, in breve DFA (Deterministic Finite Automaton), è una quintupla

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

- Q è l' insieme finito degli stati;
- Σ è l' insieme finito dei simboli di input;
- $\delta : Q \times \Sigma \rightarrow Q$ è la funzione (totale) di transizione;
- q_0 è lo stato iniziale;
- $F \subseteq Q$ è l' insieme degli stati finali o di accettazione

CONFIGURAZIONI

Lo stato di una macchina descrive la condizione in cui si trova la CPU, la configurazione descrive anche lo stato dei dispositivi input e, se presenti, delle memorie aggiuntive.

Una configurazione di un DFA $M = (Q,\Sigma,\delta,q_0,F)$ indica lo stato in cui si trova la macchina e la porzione di input ancora da leggere, è quindi un elemento di $Q\times\Sigma^*$.

Per un input $x \in \Sigma^*$, la configurazione iniziale o di partenza è (q_0,x).

Descrizione calcolo via configurazioni

La relazione "porta a" (in un passo) $\Rightarrow_M$ è una relazione binaria su $Q \times \Sigma^*$, così definita:

$$(p, ax) \Rightarrow_M (q, x) \quad \text{se e solo se } \delta(p, a) = q$$

dove $p, q \in Q$, $a \in \Sigma$ e $x \in \Sigma^*$.

Esempio

δ	*0*	*1*
q_0	q_2	q_1
q_1	q_1	q_1
q_2	q_2	q_2

$$(q_0, 101) \Rightarrow_M (q_1, 01)$$

Tutta il calcolo su *010* è descritto come segue:

$$(q_0, 101) \Rightarrow_M (q_1, 01) \Rightarrow_M (q_1, 0) \Rightarrow_M (q_1, \varepsilon)$$

Chiusura riflessiva e transitiva di una relazione binaria

Data una relazione binaria R su X, cioè $R \subseteq X^2$, la chiusura riflessiva e transitiva di R si ottiene semplicemente aggiungendo a R quello che manca perché sia riflessiva e transitiva, quindi:

R* = R ∪

{(x,x) | x è in X} (per la riflessività) ∪

{(x,z) | ∃ y (x,y) e (y,z) sono in R} (per la transitività)

Esempio: X = {a,b,ab,ba}, R= {(a,ab),(ab,ba)},

R* = R ∪ {(a,a),(b,b),(ab,ab),(ba,ba)} ∪ {(a,ba)}

Il caso della “porta a”

Nel caso di nostro interesse X è l’insieme delle configurazioni $Q \times \Sigma^*$ e R è la relazione “porta a”, $\Rightarrow_M$, e la chiusura riflessiva e transitiva consente di evitare di descrivere il calcolo nel dettaglio degli stati intermedi.

La riflessività esprime il fatto che senza leggere un input la macchina non cambia stato:

$$(q,x) \Rightarrow_{M^*} (q,x) \text{ , per ogni } q \in Q \text{ e } x \in \Sigma^*$$

la transitività consente di esprimere succintamente due o più passi di calcolo:

se $(p,aby) \Rightarrow_M (q,by)$ e $(q,by) \Rightarrow_M (r,y)$ allora $(p,aby) \Rightarrow_M^* (r,y)$.

Esempio

δ	*0*	*1*
q_0	q_2	q_1
q_1	q_1	q_1
q_2	q_2	q_2

La computazione su *010* è descritta succintamente come segue:

$(q_0,\mathit{10111111}) \Rightarrow_M^* (q_1,\varepsilon)$

DEFINIZIONE FORMALE DI ACCETTAZIONE - 1

Una parola $x \in \Sigma^*$ è accettata da un DFA $M = (Q,\Sigma,\delta,q_0,F)$ se

$$(q_0,x) \Rightarrow_M^* (q,\varepsilon) \text{ e } q \in F.$$

Il linguaggio L(M) accettato dal DFA M è l'insieme di tutte le parole accettate da M:

$$L(M)= \{x \in \Sigma^* \mid (q_0,x) \Rightarrow_M^* (q,\varepsilon) \text{ e } q \in F \}$$

FUNZIONE DI TRANSIZIONE ESTESA

Per indicare lo stato raggiunto a partire dallo stato iniziale su un input qualsiasi introduciamo l'estensione della funzione di transizione alle parole sull'alfabeto input:

$$\delta^* : Q \times \Sigma^* \rightarrow Q$$

così definita

se $|x| = 0$, allora $\delta^*(q,x) = q$.

se $|x| > 0$, allora $x=ya$, dove $y \in \Sigma^*$ e $a \in \Sigma$,

allora $\delta^*(q,x)= \delta(\delta^*(q,y),a)$.

Nota che $\delta^*(q,a) = \delta(q,a)$ per ogni $a \in \Sigma$.

DEFINIZIONE FORMALE DI ACCETTAZIONE - 2

Una parola $x \in \Sigma^*$ è accettata da un DFA $M = (Q,\Sigma,\delta,q_0,F)$ se

$$\delta^*(q_0,x) = r \in F.$$

Il linguaggio L(M) accettato dal DFA M è l'insieme di tutte le parole accettate da M:

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

LINGUAGGI REGOLARI

$\mathcal{L}$(DFA) sia la classe dei linguaggi **accettati** da un DFA, formalmente

$$\mathcal{L}(DFA) = \{L \mid \exists\, M \in DFA \text{ e } L(M) = L\}$$

Esempio 3

Si costruisca un DFA che accetta le stringhe binarie, in cui ogni sottoparola di soli 1, delimitata da 0, ha lunghezza dispari