Linguaggi di Programmazione

Capitolo 6 – Automi a stati finiti (deterministici e non deterministici)

Automi a stati finiti deterministici

- Definizione di automa a stati finiti o accettore a stati finiti o FSA
  Sia X un alfabeto. Un *automa a stati finiti* (*FSA*) è una quadrupla:
  $$M = (Q, \delta, q_0, F)$$
  ove:
  - X è detto *alfabeto di ingresso*;
  - Q è un insieme finito e non vuoto di *stati*;
  - δ è una funzione da in Q, detta *funzione di transizione*:
    $$\delta : Q \times X \rightarrow Q$$
  - q_0 è lo *stato iniziale*;
  - $F \subseteq Q$ è l'insieme degli *stati di accettazione* o *finali*.

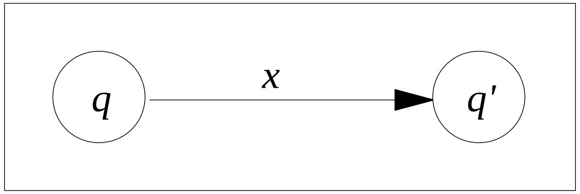
Automi a stati finiti deterministici

- Talora i valori della funzione di transizione δ non sono definiti per tutte le coppie (stato-simbolo di ingresso) (q, x). In tal caso, si dice che δ è una funzione parziale o definita parzialmente. Questo significa che la lettura di x dà luogo in q ad un comportamento dell’automa che non si ritiene utile descrivere ai fini del riconoscimento (nel senso che produrrebbe stringhe non accettate).
- Evidentemente questo fatto può essere descritto in modo equivalente, seguendo la definizione data di automa a stati finiti, passando da q, per effetto di x, in uno stato dal quale non si possa mai raggiungere uno stato finale (*stato pozza*).

Rappresentazione di FSA

- Un FSA può essere rappresentato mediante:
  - *Grafo degli Stati* o *Diagramma di Transizione* o *Diagramma di Stato*

    È una rappresentazione grafica in cui:
    - ogni stato $q \in Q$ è rappresentato da un cerchio (*nodo*) con etichetta *q*;
    - lo stato iniziale (nodo q_0) ha un arco orientato entrante libero (ossia, che non proviene da nessun altro nodo);
    - per ogni stato $q \in Q$ e per ogni simbolo *x* dell'alfabeto di ingresso, $x \in X$, se $\delta(q, x) = q'$,esiste un arco orientato etichettato con *x* uscente dal nodo *q* ed entrante nel nodo *q'*.



Rappresentazione di FSA

- Un FSA può essere rappresentato mediante:
  - *Tavola di Transizione*

    È una tabella in cui sono riportati gli stati sulle righe e i simboli dell'alfabeto di ingresso sulle colonne.

    Per ogni coppia (stato-ingresso) si legge nella tavola lo stato successivo:

δ	x_1	x_2	...	....	x_n
q_0	q_0^1	q_0^2	...	...	q_0^n
q_1	q_1^1	q_1^2	...	...	q_1^n
...	...	...	...	...	...
...	...	...	...	...	...
q_m	q_m^1	q_m^2	...	...	q_m^n

dove l'alfabeto di ingresso e l'insieme degli stati sono rispettivamente:

$$X = \{x_1, x_2, ..., x_n\} \text{ e } Q = \{q_0, q_1, ..., q_m\}$$

Quindi si ha che:

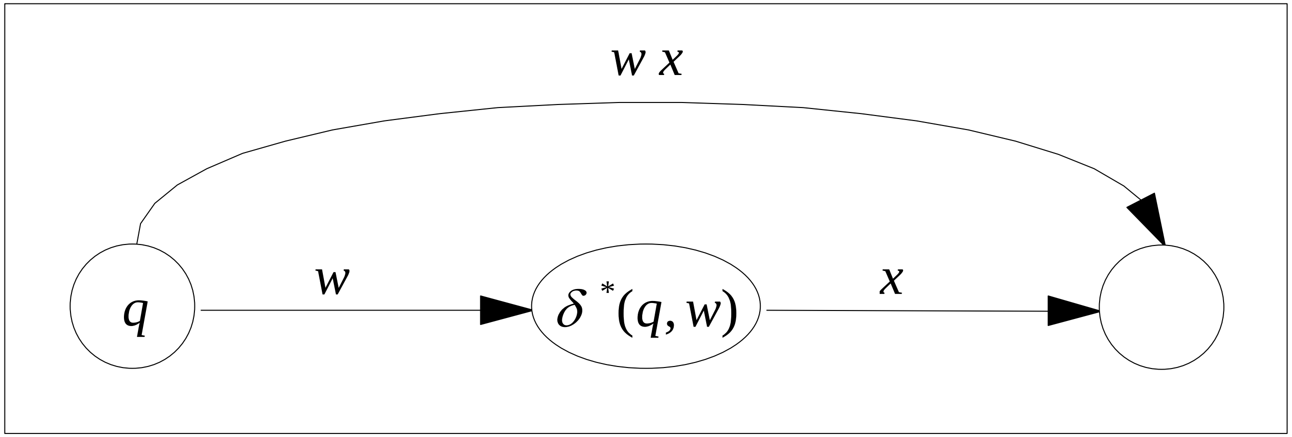
$$\delta(q_i, x_j) = q_i^j \quad \text{con} \quad q_i, q_i^j \in Q, \ \ x_j \in X$$

Estensione della funzione di transizione

- Si può definire un'estensione della funzione di transizione δ come segue:
  Definizione: δ^* *per FSA*
  Dato un FSA $M = (Q, \delta, q_0, F)$ con alfabeto di ingresso X, definiamo per induzione la funzione:
  $$\delta^* : Q \times X^* \to Q$$
  tale che $\delta^*(q, w)$, per $q \in Q$ e $w \in X^*$, sia lo stato in cui *M* si porta avendo in ingresso la parola *w* su *X* e partendo dallo stato *q*.
  $$\begin{cases} \delta^*(q, \lambda) = q \\ \delta^*(q, wx) = \delta\left(\delta^*(q, w), x\right) \end{cases} \qquad \text{per ogni } q \in Q,\ x \in X,\ w \in X^*$$

Estensione della funzione di transizione

- La figura sottostante riporta una descrizione grafica della definizione precedente.



Parola accettata o riconosciuta da un FSA

- Definizione

  Sia $M = (Q, \delta, q_0, F)$ un FSA con alfabeto di ingresso X. Una *parola* $w \in X^*$ è *accettata* (o *riconosciuta*) da M se, partendo dallo stato iniziale q_0, lo stato q in cui l'automa si porta alla fine della sequenza di ingresso w è uno stato finale.

$$w \ \text{accettata} \ \overset{def}{\Leftrightarrow} \ \delta^*(q_0, w) \in F$$

Linguaggio accettato o riconosciuto da un FSA

- Definizione

  Sia $M = (Q, \delta, q_0, F)$ un FSA con alfabeto di ingresso X. Il *linguaggio accettato* o *riconosciuto* da M è il seguente sottoinsieme di X^*:

$$T(M) = \{w \in X^* \mid \delta^*(q_0, w) \in F\}$$

  (l'insieme delle parole accettate da M).

FSA equivalenti

- Definizione

  Sia $M_1 = (Q_1, \delta_1, q_1, F_1)$ ed $M_2 = (Q_2, \delta_2, q_2, F_2)$ due FSA di alfabeto di ingresso X. M_1 ed M_2 si dicono *equivalenti* se:

$$T(M_1) = T(M_2)$$

Linguaggi a stati finiti

- Definizione

  Dato un alfabeto X, un linguaggio L su X è un *linguaggio a stati finiti* (o FSL - Finite State Language) se esiste un FSA M con alfabeto di ingresso X tale che $L = T(M)$.

Classe dei linguaggi a stati finiti

- Definizione

  Di seguito la definizione della *Classe dei Linguaggi a Stati Finiti*

$$\mathcal{L}_{FSL} = \left\{ L \in 2^{X^*} \mid \exists M,\ M \text{ è un } \mathcal{FSA} : L = T(M) \right\}$$

Proposizione

- I linguaggi a stati finiti sono *chiusi rispetto al complemento*.

  Dimostrazione

  Sia $L \in \mathcal{L}_{FSL}$ un linguaggio a stati finiti sull'alfabeto *X*. Dalla definizione di linguaggio a stati finiti, *L* = *T*(*M*), ove $M = (Q, \delta, q_0, F)$.
  Consideriamo il complemento di *L*: $\overline{L} = X^* - L$ e l'automa a stati finiti $\overline{M} = (Q, \delta, q_0, Q - F)$. Si ha: $\overline{L} = T(\overline{M})$ (si dimostra per induzione sulla lunghezza di una parola *w*).

M è un automa stati finiti con funzione di transizione δ **definita totalmente** sul proprio dominio.

Automa a stati finiti non deterministico o accettore a stati finiti non deterministico

- Definizione
  Un *automa a stati finiti non deterministico* (*NDA*) con alfabeto di ingresso X è una quadrupla:
  $$M = (Q, \delta, q_0, F)$$
  ove:
  - per Q, q_0 ed F valgono le definizioni date per gli FSA;
  - $\delta : Q \times X \rightarrow 2^Q$ è la funzione di transizione che assegna ad ogni coppia (stato-simbolo di ingresso) (q, x) un insieme $\delta(q, x) \subseteq Q$ di possibili stati successivi.

Osservazione

- Un NDA è un FSA con l’unica eccezione che, in corrispondenza di una coppia (stato-simbolo di ingresso) (q, x), vi è un insieme di stati in cui l’automa può transitare (*stati successivi possibili*).

Estensione della funzione di transizione per un NDA

- Come per gli FSA si può definire un'estensione della funzione di transizione δ come segue:
  Definizione: *δ^* per NDA*
  Dato un NDA $M=(Q,\delta,q_0,F)$ con alfabeto di ingresso X, definiamo per induzione la funzione:

$$\delta^*: 2^Q \times X^* \to 2^Q$$

$$\begin{cases} \delta^*(p,\lambda) = p \\ \delta^*(p,wx) = \bigcup_{q\in\delta^*(p,w)} \delta(q,x) \end{cases} \quad \text{per ogni} \;\; p \in 2^Q \; (p \subset Q), \;\; x \in X, \;\; w \in X^*$$

Osservazione

- Analogamente a quanto fatto per gli FSA, si dovrebbero riformulare le definizioni di parola accettata e di linguaggio accettato da un NDA.
- La complicazione, rinveniente dalla computazione non deterministica dello stato successivo in cui un NDA transita, comporta che una stessa parola può indurre cammini multipli attraverso un NDA, alcuni che terminano in stati di accettazione, altri che terminano in stati di non accettazione.

Esempio

Parola accettata o riconosciuta da un NDA

- Definizione

  Sia $M = (Q, \delta, q_0, F)$ un NDA con alfabeto di ingresso X. Una *parola* $w \in X^*$ è *accettata* (o *riconosciuta*) da M se, partendo dallo stato iniziale q_0, esiste almeno un modo per M di portarsi in uno stato di accettazione alla fine della sequenza di ingresso w. In formule:

$$w \text{ accettata } \overset{def}{\Leftrightarrow} \exists\, p : p \in \delta^*(\{q_0\}, w) \cap F \;\Leftrightarrow\; \delta^*(\{q_0\}, w) \cap F \neq \emptyset$$

Esempio

- Sia dato l'NDA dell'es. precedente. Calcoliamo $\delta^*(\{S_1\}, aba)$

$$\delta^*(\{S_1\}, aba) = \bigcup_{q \in \delta^*(\{S_1\}, ab)} \delta(q, a)$$

$$\delta^*(\{S_1\}, ab) = \bigcup_{q' \in \delta^*(\{S_1\}, a)} \delta(q', b)$$

$$\delta^*(\{S_1\}, a) = \bigcup_{q'' \in \delta^*(\{S_1\}, \lambda)} \delta(q'', a)$$

$$\delta^*(\{S_1\}, \lambda) = \{S_1\}$$

$$\delta^*(\{S_1\}, a) = \delta(S_1, a) = \{A, S_2\}$$

$$\delta^*(\{S_1\}, ab) = \delta(A, b) \cup \delta(S_2, b) = \{A\} \cup \varnothing = \{A\}$$

$$\delta^*(\{S_1\}, aba) = \delta(A, a) = \{A, S_2\}$$

Poiché $F = \{A\}$, si ha:

$$\delta^*(\{S_1\}, aba) \cap F = \{A\} \neq \varnothing$$

e w = aba è accettata da M_1.

Linguaggi accettati o riconosciuto da un NDA

- Definizione

  Sia $M = (Q, \delta, q_0, F)$ un NDA con alfabeto di ingresso X. Il *linguaggio accettato* o *riconosciuto* da M è l'insieme delle parole su X accettate da M:

$$T(M) = \left\{ w \in X^* \mid \delta^*(\{q_0\}, w) \cap F \neq \varnothing \right\}$$

  (è l'insieme delle parole w per le quali esiste almeno un cammino, etichettato con lettere di w nell'ordine da sinistra a destra, attraverso il diagramma degli stati che porta M dallo stato iniziale ad uno degli stati di accettazione).

NDA equivalenti

- Definizione
  Siano $M_1 = (Q_1, \delta_1, q_1, F_1)$ ed $M_2 = (Q_2, \delta_2, q_2, F_2)$ due NDA di alfabeto di ingresso X. M_1 ed M_2 si dicono *equivalenti* se:

$$T(M_1) = T(M_2)$$

Classe dei linguaggi riconosciuti da automi a stati finiti non deterministici

- Definizione

  Di seguito la definizione della *Classe dei Linguaggi riconosciuti da automi a Stati Finiti non deterministici*

$$\mathcal{L}_{NDL} = \left\{ L \in 2^{X^*} \mid \exists M,\ M \text{ è } un\ \mathcal{NDA} : L = T(M) \right\}$$

Equivalenza delle classi di linguaggi accettati da automi a stati finiti deterministici e non deterministici.

- Teorema
  Le classi dei linguaggi $\mathcal{L}_{FSL}$ e $\mathcal{L}_{NDL}$ sono equivalenti (1ª formulazione).
- Sia *L* un linguaggio su *X*. *L* è un linguaggio a stati finiti se e solo se *L* = *T*(*M*) per qualche NDA *M* (2ª formulazione).

Dimostrazione (2ª formulazione)

$\Rightarrow$) Sia $L \in 2^{X^*}$ ed $L \in \mathcal{L}_{FSL}$. Dalla definizione di linguaggio a stati finiti, si ha:

$$\exists M_1 : M_1 \text{ è un FSA, } M_1 = (Q_1, \delta_1, q_1, F_1)$$

con alfabeto di ingresso *X* : $L = T(M_1)$.

Equivalenza delle classi di linguaggi accettati da automi a stati finiti deterministici e non deterministici.

- Dimostrazione (2ª formulazione, continuazione)
  Sulla base di M_1, definiamo il seguente NDA con alfabeto di ingresso X:

$$M_2 = (Q_2, \delta_2, q_2, F_2)$$

ove:

- $Q_2 = Q_1$;
- δ_2 è così definita:

$$\delta_2 : Q_2 \times X \rightarrow 2^{Q_2},$$

$$\delta_2(q,x) = \{\delta_1(q,x)\} \quad \forall q \in Q_2 = Q_1, \; x \in X;$$

- $q_2 = q_1$;
- $F_2 = F_1$

M_2 è un NDA ed inoltre accetta lo stesso linguaggio accettato da M_1, ossia si ha: $T(M_2) = T(M_1)$

Equivalenza delle classi di linguaggi accettati da automi a stati finiti deterministici e non deterministici.

- Dimostrazione (2ª formulazione)

$\Leftarrow$) Sia $L \in 2^{X^*}$ *ed* $L \in \mathcal{L}_{NDL}$. Dalla definizione della classe di linguaggi $\mathcal{L}_{NDL}$, si ha:

$$\exists M,\ M \text{ è un NDA},\ M = (Q, \delta, q_0, F)$$

con alfabeto di ingresso X : $L = T(M)$.

Si può definire allora il seguente algoritmo per la costruzione di un FSA equivalente all'NDA M:

Trasformazione di un automa a stati finiti non deterministico in un automa deterministico equivalente

- Sia $M = (Q, \delta, q_0, F)$ un automa accettore a stati finiti non deterministico di alfabeto di ingresso X. M può essere trasformato in un automa deterministico M' di alfabeto di ingresso X come segue: $M' = (Q', \delta', q_0', F')$

  ove:
  - $Q' = 2^Q$
  - $q_0' = \{q_0\}$
  - $F' = \{p \subseteq Q \mid p \cap F \neq \varnothing\}$
  - $\delta' : Q' \times X \to Q' \quad \ni' \quad \forall q' = \{q_1, q_2, \ldots, q_i\} \in Q', \ \forall x \in X :$

  $$\delta'(q', x) = \delta'(\{q_1, q_2, \ldots, q_i\}, x) = \bigcup_{j=1}^{i} \delta(q_j, x) = \bigcup_{q \in q'} \delta(q, x).$$

  Si può dimostrare che M' è equivalente a M, ossia che

  $$T(M') = T(M)$$

Esercizi

Esercizi