Fondamenti d'Informatica: Linguaggi Regolari

Barbara Re, Phd

Automi a stati finiti

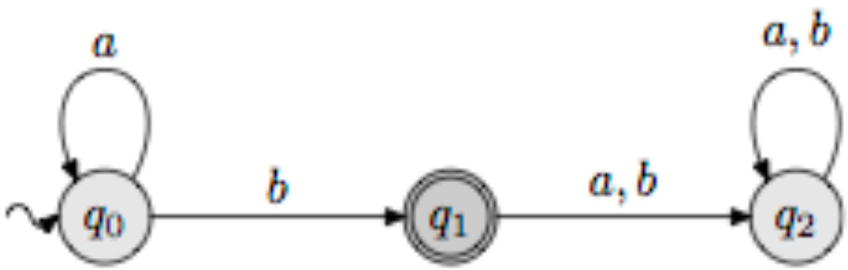
- Gli AUTOMI A STATI FINITI si possono pensare come dispositivi che, mediante una testina, leggono la stringa di input scritta su un nastro e la elaborano facendo uso di un elementare meccanismo di calcolo e di una memoria finita e limitata
- L'esame della stringa avviene un carattere alla volta, mediante una sequenza di passi di computazione, ognuno dei quali comporta lo spostamento della testina sul carattere successivo e l'aggiornamento dello stato della memoria
- Possono essere di due tipologie
  - Automi a stati finiti deterministici
  - Automi a stati finiti non deterministici

Definizione di automa a stati finito deterministico

- Un automa a stati finiti deterministico (ASFD) è una quintupla $A = \langle \Sigma, Q, \delta, q_0, F \rangle$, dove:
  - $\Sigma = \{a_1, \ldots, a_n\}$ è l'alfabeto di input,
  - $Q = \{q_0, \ldots, q_m\}$ è un insieme finito e non vuoto di stati,
  - $F \subseteq Q$ è un insieme di stati finali,
  - $q_0 \in Q$ è lo stato iniziale
  - $\delta : Q \times \Sigma \rightarrow Q$ è la funzione (totale) di transizione che ad ogni coppia ⟨stato, carattere in input⟩ associa uno stato successivo

ASFD - Esempio

- Un automa a stati finiti deterministico (ASFD) è una quintupla $A = \langle \Sigma, Q, \delta, q_0, F \rangle$, dove
  - $\Sigma = \{a, b\}$ è l'alfabeto di input,
  - $Q = \{q_0, q_1, q_2\}$ è un insieme finito e non vuoto di stati,
  - $F \subseteq \{ q_1\}$ è un insieme di stati finali,
  - $q_0 \in Q$ è lo stato iniziale
  - $\delta : Q \times \Sigma \rightarrow Q$ è la funzione (totale) di transizione che ad ogni coppia $\langle$stato, carattere in input$\rangle$ associa uno stato successivo



Diagrammi degli stati

δ	a	b
q_0	q_0	q_1
q_1	q_2	q_2
q_2	q_2	q_2

Tabella di transizione

Configurazioni

- Dato un automa a stati finiti $A = \langle \Sigma, Q, \delta, q_0, F \rangle$, una configurazione di A è una coppia (q, x), con $q \in Q$ e $x \in \Sigma^*$

- Una configurazione $\langle q, x \rangle$, $q \in Q$ ed $x \in \Sigma^*$, di A, è detta:
  - Iniziale se $q = q0$;
  - Finale se $x = \varepsilon$;
  - Accettante se $x = \varepsilon$ e $q \in F$

Transizioni tra configurazioni

- La funzione di transizione permette di definire la relazione di transizione tra configurazioni, che associa ad una configurazione la configurazione successiva

- Dato un ASFD $A = \langle \Sigma, Q, \delta, q_0, F \rangle$ e due configurazioni (q, x) e (q', y) di A, avremo che

  $(q, x) \mid\!\text{---}\ (q', y)$

  se e solo se valgono le due condizioni:
- Esiste $a \in \Sigma$ tale che $x = ay$;
- $\delta(q, a) = q'$

Accettazione stringa

- A questo punto possiamo dire che, dato un automa a stati finiti deterministico $A = \langle \Sigma, Q, \delta, q_0, F \rangle$, una stringa $x \in \Sigma^*$ è accettata da A se e solo se (q_0, x) |--A-- (q, ε), con $q \in F$,

- Possiamo definire il linguaggio riconosciuto da A

$$L(\mathcal{A}) = \left\{ x \in \Sigma^* \mid (q_0, x) \overset{*}{\underset{\mathcal{A}}{\vdash}} (q, \varepsilon), q \in F \right\}$$

Linguaggi riconoscibili

- L'insieme R dei linguaggi riconoscibili da automi a stati finiti deterministici, cioè l'insieme

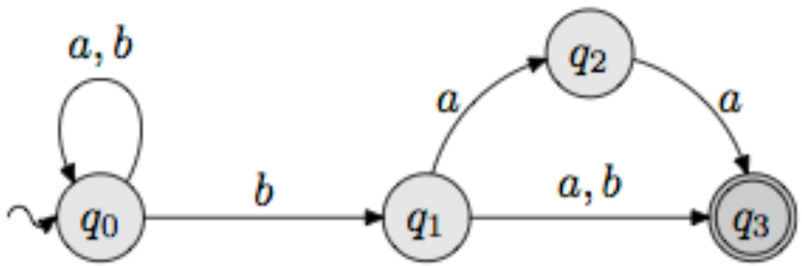
$$R = \{L \mid L \subseteq \Sigma^* \; ed\ esiste\ un\ automa\ A\ tale\ che\ L = L(A)\}$$

Automi a stati finiti non deterministrici

- Un automa a stati finiti non deterministico (ASFND) è una quintupla $A = \langle \Sigma, Q, \delta_N, q_0, F \rangle$, dove
  - $\Sigma = \{a_1, ..., a_n\}$ è l'alfabeto di input,
  - $Q = \{q_0,...,q_m\}$ è un insieme finito e non vuoto di stati interni,
  - $F \subseteq Q$ è un insieme di stati finali,
  - $q_0 \in Q$ è lo stato iniziale
  - $\delta : Q \times \Sigma \to P(Q)$ è la funzione (parziale) di transizione che ad ogni coppia ⟨stato, carattere in input⟩ associa un sottoinsieme di Q (eventualmente vuoto)

ASFND

- Un automa a stati finiti non deterministico (ASFND) è una quintupla $A = \langle \Sigma, Q, \delta_N, q_0, F \rangle$, dove
  - $\Sigma = \{a, b\}$ è l'alfabeto di input,
  - $Q = \{q_0, q_1, q_2, q_3\}$ è un insieme finito e non vuoto di stati interni,
  - $F = \{q_3\}$ è un insieme di stati finali,
  - $q_0 \in Q$ è lo stato iniziale
  - $\delta : Q \times \Sigma \rightarrow P(Q)$ è la funzione (parziale) di transizione che ad ogni coppia $\langle$stato, carattere in input$\rangle$ associa un sottoinsieme di Q (eventualmente vuoto)

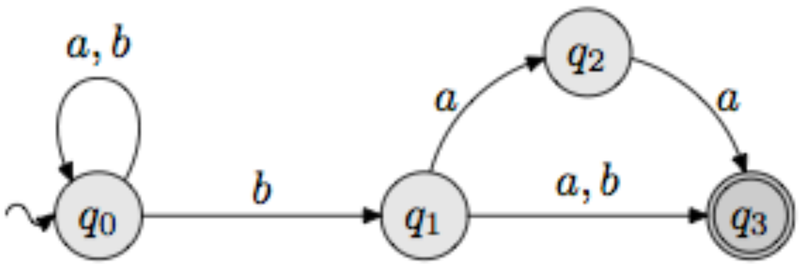


Diagrammi degli stati

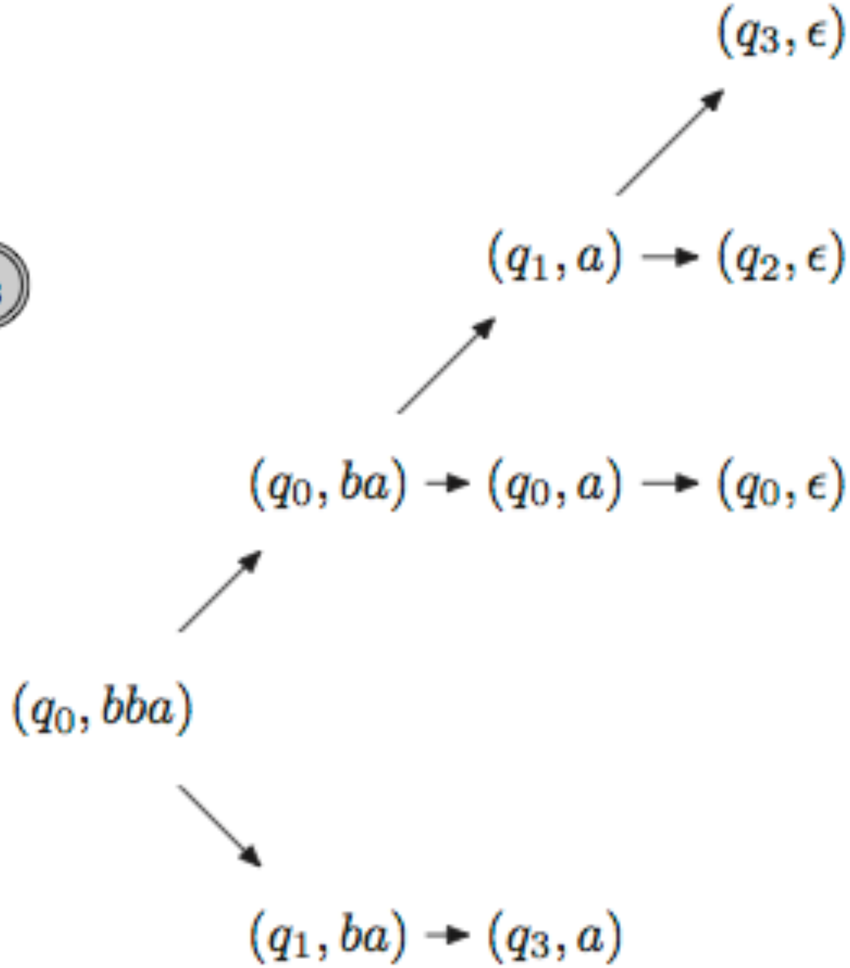
δ_N	a	b
q_0	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_2, q_3\}$	$\{q_3\}$
q_2	$\{q_3\}$	$\emptyset$
q_3	$\emptyset$	$\emptyset$

Tabella di transizione

Albero di Computazione



Automa



Albero di computazione
sulla stringa bba

Una stringa viene accettata da un automa a stati finiti non deterministico <u>se almeno una delle computazioni</u> definite per la stringa stessa è di accettazione

Il linguaggio L(A) accettato da un ASFND A

$$L(\mathcal{A}) = \left\{ x \in \Sigma^* \mid (\{q_0\}, x) \stackrel{*}{\longmapsto} (Q, \varepsilon), Q \cap F \neq \emptyset \right\}$$

Relazioni tra ASFD, ASFND

- In base alle definizioni date si potrebbe pensare che gli ASFND siano dispositivi più potenti degli ASFD, nel senso che riconoscono una classe di linguaggi più ampia

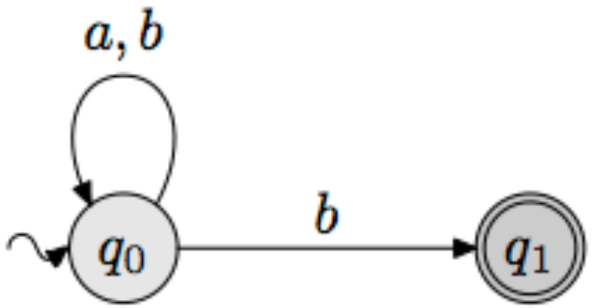
- In altri tipi di macchine astratte il non determinismo accresce effettivamente il potere computazionale, nel caso degli automi a stati finiti non è così

- Si può dimostrare che gli ASFD e gli ASFND riconoscono la medesima classe di linguaggi

- Teorema: Dato un ASFD che riconosce un linguaggio L, esiste corrispondentemente un ASFND che riconosce lo stesso linguaggio L; viceversa, dato un ASFND che riconosce un linguaggio L', esiste un ASFD che riconosce lo stesso linguaggio L'

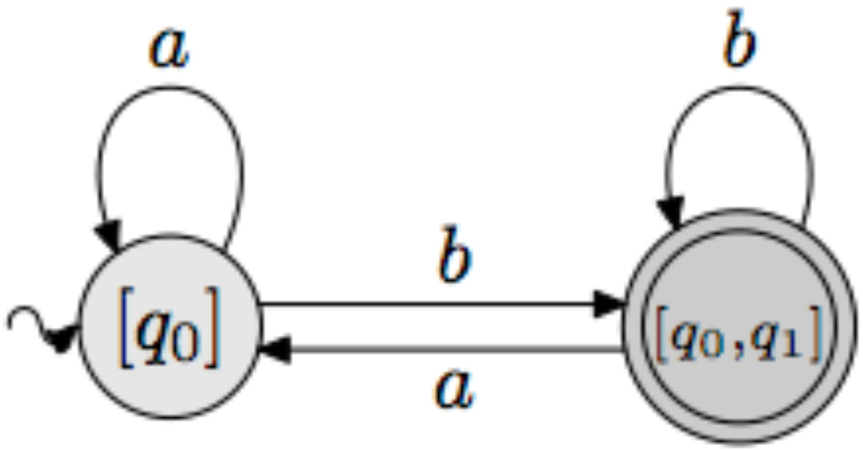
Corrispondenza ASFND e ASFD

- Dato il seguente ASFND che riconosce le stringhe in {a, b}* terminanti con b determinare il corrispondente ASFD



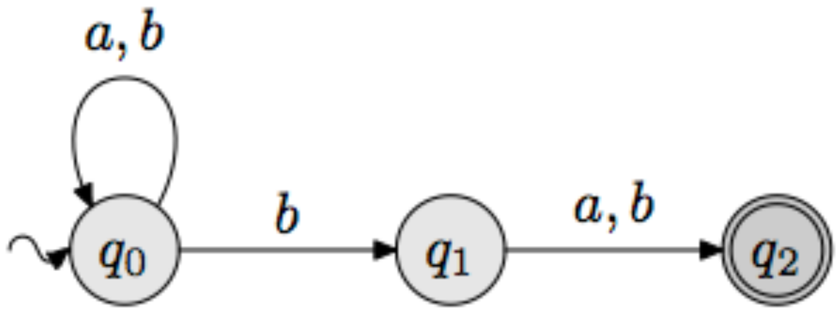
Soluzione

- ASFD che riconosce le stringhe in {a, b}* terminanti con b



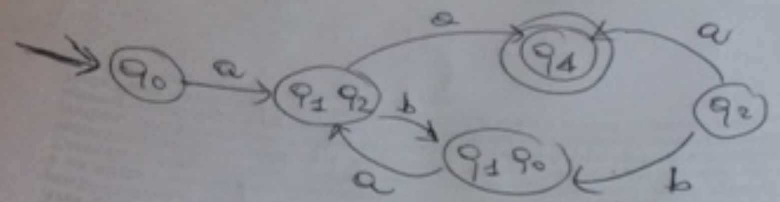
Corrispondenza ASFND e ASFD (II)

- Dato il seguente ASFND che riconosce le stringhe date da $(a + b)^*b(a + b)$ determinare il corrispondente ASFD



ASFD e ASFND

- Gli automi deterministici e quelli non deterministici hanno lo stesso potere espressivo!!!
- Questo significa che se un linguaggio può essere accettato da un ASFND allora esiste anche un ASFD che lo accetta, e viceversa
- Esiste un algoritmo noto come costruzione dei sottoinsiemi (subset construction) che serve per costruire, a partire da un ASFND dato, un ASFD equivalente che simula il non determinismo, ma è deterministico



1) Tabella Transizione con righe tutti gli stati ASFND e colonne alfabeto ASFND.

2) Vedo nuovi stati sulla tabella e li considero come righe (FINO A CHE SI GENERA NUOVI STATI)

3) Uso tabella ASFD.

↓

a) stato iniziale ASFD = stato iniziale ASFND

b) stato finale ~~ASFND~~ ASFD = tutti gli stati che contengono stato finale ASFND

c) transizioni sulla tabella.

Idea

- Partiamo da un insieme di stati DStates contenente solo uno stato iniziale non marcato
- Ad ogni passo selezioniamo uno stato non marcato da DStates e ne calcoliamo le transizioni uscenti aggiungendo a DStates, non marcati, eventuali nuovi stati trovati
- Prima o poi, dato che il numero di stati possibili è non ci saranno più stati non marcati da considerare e l'algoritmo terminerà

Algoritmo di traduzione

LINGUAGGIO: Pseudocodice.
INPUT: Un automa non deterministico N.
OUTPUT: Un automa deterministico equivalente D.

Sia $\{s_0\}$ l'unico stato non marcato di $DStates$;
while c'e' uno stato non marcato T in $DStates$ **do**
    begin
        marca T;
        for each simbolo di input $x \in \Sigma$ **do**
            begin
                $U := \overline{move}(T, x)$;
                if U non e' in $DStates$ **then**
                    begin
                        aggiungi U, non marcato, a $DStates$;
                    end
                $Dtran(T, x) := U$;
            end
    end

lo stato iniziale di D e' $\{s_0\}$
gli stati finali di D sono tutti quelli che contengono almeno uno stato finale di N

$$\overline{move}: (\wp(S) \times \Sigma) \longrightarrow \wp(S)$$

$$\overline{move}(T, x) = \bigcup_{s \in T} move(s, x)$$

Relazioni tra automi e grammatiche

- Teorema: Data una grammatica regolare $G = \langle V_T, V_N, P, S\rangle$, esiste un ASFND $AN = \langle \Sigma, Q, \delta N, q_0, F \rangle$ che riconosce il linguaggio che essa genera; viceversa, dato un ASFND esiste una grammatica regolare che genera il linguaggio che esso riconosce

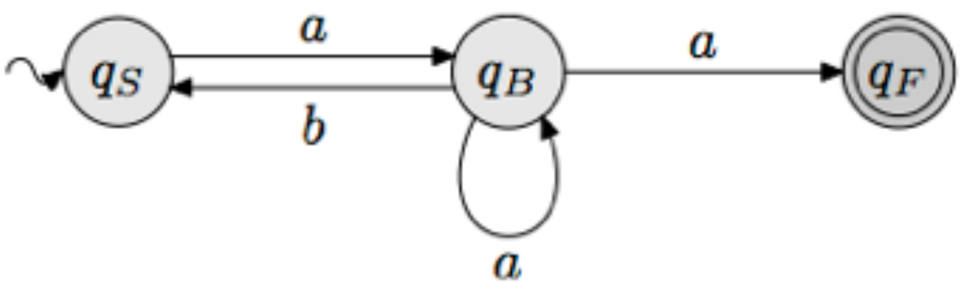
In pratica

$G = \langle\{a, b\}, \{S, B\}, P, S\rangle$

in cui P è dato da

S ☾ aB

$B \rightarrow aB \mid bS \mid a$

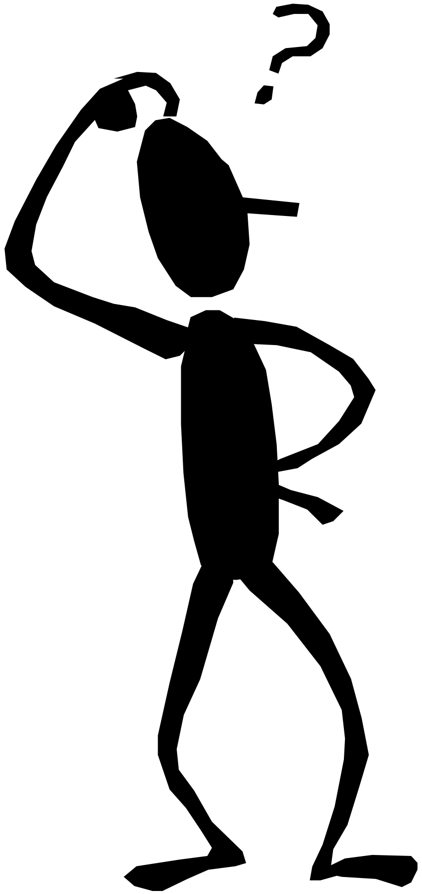


- $\Sigma = V_T$
- $Q = \{q_s, q_B, q_F\}$
- $q_0 = q_s$ *come stato iniziale*
- $F = \{q_F\}$
- δ_N è *definita come segue*

$\delta_N(q_S, a) = \{qB\}$

$\delta_N(q_B, a) = \{qB, qF\}$

$\delta_N(qB, b) = \{qS\}$



Questions?