

Automi a pila

a.a. 2020-2021

Corso di Fondamenti di Informatica - 1 modulo

Corso di Laurea in Informatica

Università di Roma "Tor Vergata"

Prof. Giorgio Gambosi



Automi a pila

Un automa a pila (o automa push-down) è definito come una settupla $\mathcal{M} = \langle \Sigma, \Gamma, Z_0, Q, q_0, F, \delta \rangle$ dove Σ è l'**alfabeto di input**, Γ è l'**alfabeto dei simboli della pila**, $Z_0 \in \Gamma$ è il **simbolo iniziale di pila**, Q è un insieme finito e non vuoto di **stati**, $q_0 \in Q$ è lo **stato iniziale**, $F \subseteq Q$ è l'insieme degli **stati finali**, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \longrightarrow Q \times \Gamma^*$ è la **funzione (parziale) di transizione**.

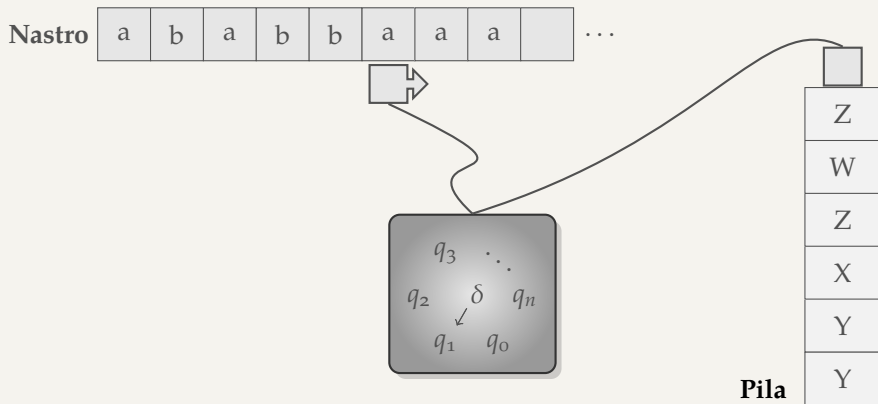
L'automa introdotto non è deterministico, in quanto la presenza di ε -transizioni può comportare che esistano continuazioni diverse di una stessa computazione anche in presenza di uno stesso carattere letto.

Per ottenere un comportamento deterministico dobbiamo fare l'ulteriore ipotesi che se, per una coppia $q \in Q, Z \in \Gamma$, è definita $\delta(q, \varepsilon, Z)$ allora la funzione di transizione $\delta(q, a, Z)$ non deve essere definita per nessun $a \in \Sigma$.

Ad ogni passo l'automa, a partire dallo stato attuale, dal carattere letto sul nastro di input e dal carattere affiorante sulla pila, sostituisce il simbolo affiorante nella pila con una stringa di caratteri e si porta in un nuovo stato.

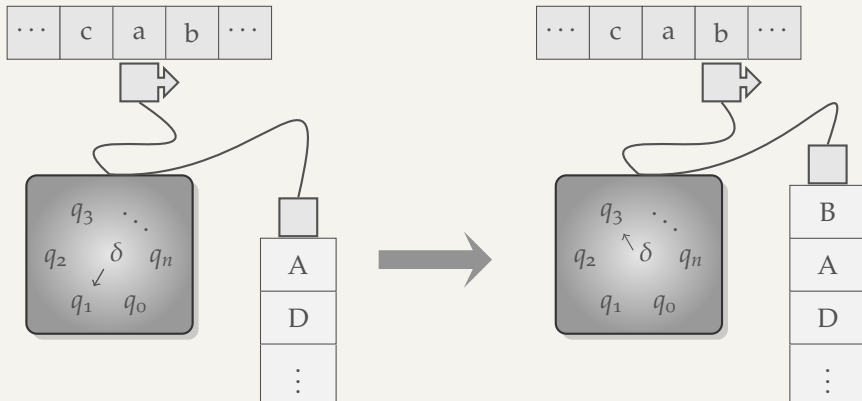
La convenzione è che il primo carattere della stringa diventi il simbolo di pila affiorante. Si noti che se la stringa che viene inserita nella pila è la stringa vuota, ciò equivale a dire che il simbolo precedente affiorante nella pila è stato cancellato.

Automi a pila



Automi a pila

$$\delta(q_1, a, A) = (q_3, BA)$$



Dato un automa a pila $\mathcal{M} = \langle \Sigma, \Gamma, Z_0, Q, q_0, F, \delta \rangle$, una *configurazione* di \mathcal{M} è data dalla tripla $\langle q, x, \gamma \rangle$, dove $q \in Q$, $x \in \Sigma^*$ e $\gamma \in \Gamma^*$.

Automi a pila

Sia $\mathcal{M} = \langle \Sigma, \Gamma, Z_0, Q, q_0, F, \delta \rangle$ un automa a pila e siano (q, x, γ) e (q', x', γ') due configurazioni di \mathcal{M} : avremo allora che $(q, x, \gamma) \xrightarrow[\mathcal{M}]{} (q', x', \gamma')$ se e solo se valgono le tre condizioni:

1. esiste $a \in \Sigma$ tale che $x = ax'$;
2. esistono $Z \in \Gamma$ e $\eta, \zeta \in \Gamma^*$ tali che $\gamma = Z\eta$ e $\gamma' = \zeta\eta$;
3. $\delta(q, a, Z) = (q', \zeta)$;

oppure le tre condizioni:

1. $x = x'$;
2. esistono $Z \in \Gamma$ e $\eta, \zeta \in \Gamma^*$ tali che $\gamma = Z\eta$ e $\gamma' = \zeta\eta$;
3. $\delta(q, \epsilon, Z) = (q', \zeta)$.

Una computazione è definita come una sequenza c_0, \dots, c_k di configurazioni di \mathcal{M} tale che $c_i \xrightarrow[\mathcal{M}]{} c_{i+1}$.

Automi a pila: accettazione per pila vuota

Sia \mathcal{M} un automa a pila. Una configurazione (q, x, γ) di \mathcal{M} è di accettazione se $x = \gamma = \varepsilon$. Secondo tale definizione, una stringa x è quindi **accettata** da \mathcal{M} se e solo se al termine della scansione della stringa la pila è vuota.

Indichiamo con $N(\mathcal{M})$ il linguaggio accettato per pila vuota dall'automata \mathcal{M} .

Automi a pila: accettazione per stato finale

Sia \mathcal{M} un automa a pila. Una configurazione (q, x, γ) di \mathcal{M} è di accettazione se $x = \varepsilon$ e $q \in F$. Secondo tale definizione, una stringa x è quindi **accettata** da \mathcal{M} se e solo se al termine della scansione della stringa l'automa si trova in uno stato finale.

Indichiamo con $L(\mathcal{M})$ il linguaggio accettato per stato finale dall'automa

Automi a pila nondeterministici

Un **automa a pila non deterministico** è definito come una settupla $\mathcal{M} = \langle \Sigma, \Gamma, Z_0, Q, q_0, F, \delta \rangle$ dove Σ è l'**alfabeto di input**, Γ è l'**alfabeto dei simboli della pila**, $Z_0 \in \Gamma$ è il **simbolo iniziale di pila**, Q è un insieme finito e non vuoto di **stati**, $q_0 \in Q$ è lo **stato iniziale**, $F \subseteq Q$ è l'insieme degli **stati finali**, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \mapsto \mathcal{P}(Q \times \Gamma^*)$ è la **funzione (parziale) di transizione**.

Automi a pila nondeterministici

Nel caso degli automi a pila la presenza del non determinismo comporta un aumento del potere computazionale.

Mentre gli automi a pila non deterministici riconoscono la classe dei linguaggi non contestuali, gli automi a pila deterministici riconoscono un sottoinsieme proprio di tali linguaggi, la classe dei linguaggi non contestuali deterministici.

Esempio: accettazione del linguaggio non contestuale $\{w\tilde{w} \mid w \in \{a,b\}^+\}$, dove \tilde{w} indica la stringa riflessa di w

Esempio di accettazione per pila vuota

Tabella di transizione dell'automa a pila che accetta per pila vuota il linguaggio $\{w\tilde{w} \mid w \in \{a,b\}^+\}$

	A		B		Z_0	
	a	b	a	b	a	b
q_0	(q_0, AA) (q_1, ε)	(q_0, BA)	(q_0, AB)	(q_0, BB) (q_1, ε)	(q_0, A)	(q_0, B)
q_1	(q_1, ε)			(q_1, ε)		

Esempio di accettazione per stato finale

Tabella di transizione dell'automa a pila che accetta per stato finale il linguaggio $\{w\tilde{w} \mid w \in \{a,b\}^+\}$, con $q_F = q_2$

	A		B		Z_0		
	a	b	a	b	a	b	ε
q_0	(q_0, AA) (q_1, ε)	(q_0, BA)	(q_0, AB)	(q_0, BB) (q_1, ε)	(q_0, AZ_0)	(q_0, BZ_0)	
q_1	(q_1, ε)			(q_1, ε)			(q_2, ε)

Teorema

Dato un automa a pila non deterministico $\mathcal{M} = \langle \Sigma, \Gamma, Z_0, Q, q_0, F, \delta \rangle$ che accetta un linguaggio per stato finale, esiste un automa a pila non deterministico $\mathcal{M}' = \langle \Sigma', \Gamma', Z'_0, Q', q'_0, \emptyset, \delta' \rangle$ che accetta lo stesso linguaggio per pila vuota, vale a dire tale che $L(\mathcal{M}) = N(\mathcal{M}')$.

Equivalenza tra condizioni di accettazione

\mathcal{M}' opera in modo simile a \mathcal{M} , secondo lo schema seguente:

1. All'inizio \mathcal{M}' ha nella pila un simbolo speciale X non appartenente a Γ ed inserisce al di sopra di X il simbolo Z_0 ;
2. Quindi, \mathcal{M}' esegue gli stessi passi di \mathcal{M} . Si noti che nel corso di tale fase la pila di \mathcal{M}' non sarà mai vuota;
3. Se, alla fine della stringa di input, \mathcal{M} raggiunge un suo stato finale (che quindi è raggiunto anche da \mathcal{M}'), \mathcal{M}' provvede ad eliminare tutti i simboli presenti in pila, incluso X .

Equivalenza tra condizioni di accettazione

\mathcal{M}' allora definito come $\Sigma' = \Sigma$, $\Gamma' = \Gamma \cup \{X\}$ (X non in Γ), $Z'_0 = X$, $Q' = Q \cup \{q'_0, q_f\}$ (q'_0 e q_f non in Q) e, per quanto riguarda la funzione di transizione δ' :

1. $\delta'(q, a, Z) = \delta(q, a, Z)$ per ogni $q \in Q, a \in \Sigma, Z \in \Gamma$
2. $\delta'(q, \varepsilon, Z) = \delta(q, \varepsilon, Z)$ per ogni $q \in Q - F, Z \in \Gamma$
3. $\delta'(q, \varepsilon, Z) = \delta(q, \varepsilon, Z) \cup \{(q_f, \varepsilon)\}$ per ogni $q \in F, Z \in \Gamma$
4. $\delta'(q'_0, \varepsilon, X) = \{(q_0, Z_0 X)\}$
5. $\delta'(q_f, \varepsilon, Z) = \{(q_f, \varepsilon)\}$ per ogni $Z \in \Gamma'$

Equivalenza tra condizioni di accettazione

- Per effetto di 1 e 2 \mathcal{M}' simula perfettamente \mathcal{M} se questo non si trova in uno stato finale.
- Per effetto di 1 e 3 \mathcal{M}' in uno stato finale può eseguire tutte le transizioni definite per \mathcal{M} con l'ulteriore possibilità di effettuare una ε -transizione verso il suo stato q_f lasciando la pila immutata.
- La condizione 4 fa sì che il simbolo X rimanga in fondo alla pila mentre l'automa \mathcal{M}' effettua la simulazione di \mathcal{M}
- La condizione 5 assicura lo svuotamento della pila di \mathcal{M}' nel caso in cui \mathcal{M} raggiunga uno stato finale.

Teorema

Dato un automa a pila non deterministico $\mathcal{M} = \langle \Sigma, \Gamma, Z_0, Q, q_0, F, \delta \rangle$ che accetta un linguaggio per pila vuota, esiste un automa a pila non deterministico $\mathcal{M}' = \langle \Sigma', \Gamma', Z'_0, Q', q'_0, \emptyset, \delta' \rangle$ che accetta lo stesso linguaggio per stato finale, vale a dire tale che $N(\mathcal{M}) = L(\mathcal{M}')$.

Equivalenza tra condizioni di accettazione

\mathcal{M}' opera similmente a \mathcal{M} , secondo lo schema:

1. All'inizio \mathcal{M}' ha nella pila un simbolo speciale X non appartenente a Γ ed inserisce Z_0 al di sopra di X ;
2. Quindi, \mathcal{M}' esegue gli stessi passi di \mathcal{M} . Si noti che nel corso di tale fase la pila di \mathcal{M}' non sarà mai vuota;
3. Se, alla fine della stringa di input, \mathcal{M} raggiunge la condizione di pila vuota, \mathcal{M}' entra nello stato finale q_f .

Equivalenza tra condizioni di accettazione

\mathcal{M}' allora definito come $\Sigma' = \Sigma$, $\Gamma' = \Gamma \cup \{X\}$ (X non in Γ), $Z'_0 = X$, $Q' = Q \cup \{q'_0, q_f\}$ (q'_0 e q_f non in Q) e, per quanto riguarda la funzione di transizione δ' :

1. $\delta'(q, a, Z) = \delta(q, a, Z)$ per ogni $q \in Q, a \in \Sigma \cup \{\varepsilon\}, Z \in \Gamma$,
2. $\delta'(q, \varepsilon, X) = \{(q_f, X)\}$ per ogni $q \in Q$
3. $\delta'(q'_0, \varepsilon, X) = \{(q_0, Z_0X)\}$.

Equivalenza tra condizioni di accettazione

- Per effetto di 1 \mathcal{M}' simula perfettamente \mathcal{M} se la pila di questo non è vuota
- Per effetto di 2 \mathcal{M}' , se la pila di \mathcal{M} è vuota (e quindi la propria contiene solo X), può effettuare una ε -transizione verso il suo stato finale q_f lasciando la pila immutata
- La 3 fa sì che il simbolo X rimanga in fondo alla pila mentre l'automa \mathcal{M}' effettua la simulazione di \mathcal{M} .

Equivalenza CFG-NPDA

Se un linguaggio è generato da una grammatica \mathcal{G} non contestuale, esiste un automa a pila \mathcal{M} tale che $L(\mathcal{G}) = \mathcal{N}(\mathcal{M})$.

La dimostrazione è costruttiva: a partire da \mathcal{G} deriviamo \mathcal{M} e mostriamo poi l'equivalenza.

Equivalenza CFG-NPDA

Consideriamo dapprima il caso in cui $\varepsilon \notin L(\mathcal{G})$. A partire da \mathcal{G} , costruiamo una grammatica $\mathcal{G}' = \langle \Sigma, V'_N, P', S' \rangle$ equivalente a \mathcal{G} , ma in Forma Normale di Greibach.

Per l'automa, poniamo $\Gamma = V'_N, Q = \{q_0\}, Z_0 = S'$.

Funzione di transizione: per ogni produzione $A \longrightarrow a\gamma, \gamma \in (V'_N)^*$ introduciamo la regola $\delta(q_0, a, A) = (q_0, \gamma)$.

L'equivalenza deriva dalla dimostrazione che

$$(q, x, S') \stackrel{*}{\vdash} (q, \varepsilon, \alpha)$$

se e solo se $S' \stackrel{*}{\Rightarrow} x\alpha$, dove $q \in Q$, $x \in \Sigma^*$ e $\alpha \in (V'_N)^*$

Equivalenza CFG-NPDA

In due passi:

1. Dimostrazione, per induzione su i (numero di passi della computazione), che

$$\text{se } (q, x, S') \vdash^i (q, \varepsilon, \alpha) \quad \text{allora} \quad S' \xRightarrow{*} x\alpha.$$

2. Dimostrazione, per induzione su i (numero di passi della derivazione), che

$$\text{se } S' \xRightarrow{i} x\alpha \quad \text{allora} \quad (q, x, S') \vdash^* (q, \varepsilon, \alpha).$$

Se $\varepsilon \in L(\mathcal{G})$ possiamo costruire una grammatica \mathcal{G}' in GNF che genera il linguaggio $L(\mathcal{G}') = L(\mathcal{G}) - \{\varepsilon\}$ e quindi applicare la procedura descritta per ottenere un automa a pila $\langle \Sigma, \Gamma, Z_0, \delta, \{q_0\}, \emptyset \rangle$ che riconosce $L(\mathcal{G}')$.

Un automa che riconosce $L(\mathcal{G})$ può essere ottenuto aggiungendo uno nuovo stato iniziale q'_0 e la transizione $\delta(q'_0, \varepsilon, Z_0) = \{(q'_0, \varepsilon), (q_0, Z_0)\}$.

Equivalenza NPDA-CFG

Sia L un linguaggio accettato mediante pila vuota da un automa a pila $\mathcal{M} = \langle \Sigma, \Gamma, Q, \delta, Z_0, q_0, \emptyset \rangle$, allora esiste una grammatica non contestuale \mathcal{G} che lo genera, cioè $L = N(\mathcal{M}) = L(\mathcal{G})$.

La dimostrazione è costruttiva: a partire da \mathcal{M} deriviamo \mathcal{G} e mostriamo poi l'equivalenza.

Equivalenza NPDA-CFG

Definiamo

$$V_N = \{[q_i, A, q_j] \mid \text{per ogni } q_i, q_j \in Q, A \in \Gamma\} \cup \{S\}.$$

dove S è l'assioma.

- Idea di base: $[q_i, A, q_j] \xRightarrow[\mathcal{E}]{*} \sigma$ se e solo se $(q_i, \sigma, A) \xRightarrow[\mathcal{M}]{*} (q_j, \varepsilon, \varepsilon)$
- quindi $[q_i, A, q_j] \xRightarrow[\mathcal{E}]{*} \sigma$ se e solo se \mathcal{M} , a partire dalla configurazione (q_i, σ, A) , può raggiungere (per una sequenza opportuna di scelte) la configurazione $(q_j, \varepsilon, \varepsilon)$

Equivalenza NPDA-CFG

Definiamo le produzioni P nel modo seguente:

1. per ogni $q \in Q$:

$$S \longrightarrow [q_0, Z_0, q] \in P$$

- per ipotesi, $[q_0, Z_0, q] \xRightarrow[\mathcal{G}]{*} \sigma$ se e solo se \mathcal{M} , a partire dalla configurazione (q_0, σ, Z_0) , può raggiungere (per una sequenza opportuna di scelte) la configurazione $(q, \varepsilon, \varepsilon)$ e quindi se \mathcal{M} accetta σ per pila vuota (q è lo stato in cui \mathcal{M} si viene a trovare)
- quindi, $S \implies [q_0, Z_0, q] \xRightarrow{*} \sigma$ se e solo se \mathcal{M} accetta σ e quindi $\sigma \in L(\mathcal{G})$ se e solo se $\sigma \in N(\mathcal{M})$

2. per ogni $q \in Q, A \in \Gamma, a \in \Sigma \cup \{\varepsilon\}$

se $(q', \varepsilon) \in \delta(q, a, A)$ allora $[q, A, q'] \longrightarrow a \in P$

- mantiene l'ipotesi, in quanto $[q, A, q'] \xRightarrow{*} a$ e \mathcal{M} , a partire da (q, A, a) , può raggiungere la configurazione $(q', \varepsilon, \varepsilon)$

Equivalenza NPDA-CFG

3. per ogni $q \in Q, A \in \Gamma, a \in \Sigma \cup \{\varepsilon\}$

se $(q', B_1 \dots B_m) \in \delta(q, a, A)$ allora

$$[q, A, q_{i_m}] \longrightarrow a[q' B_1 q_{i_1}][q_{i_1} B_2 q_{i_2}] \dots [q_{i_{m-1}} B_m q_{i_m}] \in P$$

per ogni possibile sequenza q_{i_1}, \dots, q_{i_m} di m stati (non distinti) in Q ,

- Ipotesi: $[q, A, q_{i_m}] \xrightarrow[\mathcal{E}]{*} \sigma$ se e solo se \mathcal{M} , a partire da (q, σ, A) , può giungere a $(q_{i_m}, \varepsilon, \varepsilon)$
- Consideriamo il caso in cui il primo passo di \mathcal{M} è $\delta(q, a, A) = (q', B_1 \dots B_m)$: allora \mathcal{M} potrà raggiungere $(q_{i_m}, \varepsilon, \varepsilon)$ soltanto attraverso una sequenza di passi di cui la prima parte (leggendo una stringa σ_1 e giungendo a uno stato q_{i_1}) porterà all'eliminazione di B_1 dallo stack, la seconda (leggendo una stringa σ_2 e giungendo a uno stato q_{i_2}) all'eliminazione di B_2 e così via

Equivalenza NPDA-CFG

- Quindi, si ha che:
 - \mathcal{M} può passare da $(q_{i_{k-1}}, \sigma_k B_k)$ a $(q_{i_k}, \varepsilon, \varepsilon)$ per $k = 1, \dots, m$
 - $\sigma = a\sigma_1\sigma_2\sigma_m$
- Corrispondentemente, la grammatica \mathcal{G} può generare σ a partire da $[q, A, q_{i_m}]$ producendo in un passo la frase $a[q'B_1q_{i_1}][q_{i_1}B_2q_{i_2}] \dots [q_{i_{m-1}}B_mq_{i_m}]$ e utilizzando le derivazioni $[q_{i_{k-1}}B_kq_{i_k}] \xRightarrow{*} \sigma_k$

Consideriamo l'automa a pila avente la funzione di transizione

$$\delta(q_0, 0, Z_0) = \{(q_0, XZ_0)\}$$

$$\delta(q_0, 0, X) = \{(q_0, XX)\}$$

$$\delta(q_0, 1, X) = \{(q_1, \varepsilon)\}$$

$$\delta(q_1, 1, X) = \{(q_1, \varepsilon)\}$$

$$\delta(q_1, \varepsilon, X) = \{(q_1, \varepsilon)\}$$

$$\delta(q_1, \varepsilon, Z_0) = \{(q_1, \varepsilon)\}$$

$$S \longrightarrow [q_0 Z_0 q_0]$$

$$S \longrightarrow [q_0 Z_0 q_1]$$

Da $\delta(q_0, o, Z_0) = \{(q_0, XZ_0)\}$:

$$[q_0 Z_0 q_0] \longrightarrow o[q_0 X q_0][q_0 Z_0 q_0]$$

$$[q_0 Z_0 q_0] \longrightarrow o[q_0 X q_1][q_1 Z_0 q_0]$$

$$[q_0 Z_0 q_1] \longrightarrow o[q_0 X q_0][q_0 Z_0 q_1]$$

$$[q_0 Z_0 q_1] \longrightarrow o[q_0 X q_1][q_1 Z_0 q_1]$$

Da $\delta(q_0, o, X) = \{(q_0, XX)\}$:

$$[q_0 X q_0] \longrightarrow o[q_0 X q_0][q_0 X q_0]$$

$$[q_0 X q_0] \longrightarrow o[q_0 X q_1][q_1 X q_0]$$

$$[q_0 X q_1] \longrightarrow o[q_0 X q_0][q_0 X q_1]$$

$$[q_0 X q_1] \longrightarrow o[q_0 X q_1][q_1 X q_1]$$

$$[q_0 X q_1] \longrightarrow 1$$

$$[q_1 X q_1] \longrightarrow 1$$

$$[q_1 X q_1] \longrightarrow \varepsilon$$

$$[q_1 Z_0 q_1] \longrightarrow \varepsilon$$

Portata in forma ridotta, la grammatica risultante sarà

$$S \longrightarrow o[q_0 X q_1][q_1 Z_0 q_1]$$

$$[q_0 X q_1] \longrightarrow o[q_0 X q_1][q_1 X q_1] \mid 1$$

$$[q_1 X q_1] \longrightarrow 1 \mid \varepsilon$$

$$[q_1 Z_0 q_1] \longrightarrow \varepsilon.$$

L'equivalenza deriva dalla dimostrazione che

$$[q, A, q'] \xRightarrow{*} x \quad \text{se e solo se} \quad (q, x, A) \vdash^* (q', \varepsilon, \varepsilon)$$

da cui otteniamo come caso particolare, che $[q_0, Z_0, q'] \xRightarrow{*} x$ se e solo se $(q_0, x, Z_0) \vdash^* (q', \varepsilon, \varepsilon)$ per qualche q' , cioè se e solo se l'automa accetta la stringa x per pila vuota.

Equivalenza NPDA-CFG

Accettazione/produzione della stringa 00011

letti/prodotti	stato	da leggere	pila	frase
	q_0	0	Z_0	S
0	q_0	0	XZ_0	$0[q_0Xq_1][q_1Z_0q_1]$
00	q_0	0	XXZ_0	$00[q_0Xq_1][q_1Xq_1][q_1Z_0q_1]$
000	q_0	1	$XXXZ_0$	$000[q_0Xq_1][q_1Xq_1][q_1Xq_1][q_1Z_0q_1]$
0001	q_1	1	XXZ_0	$0001[q_1Xq_1][q_1Xq_1][q_1Z_0q_1]$
00011	q_1	ε	XZ_0	$00011[q_1Xq_1][q_1Z_0q_1]$
00011	q_1	ε	Z_0	$00011[q_1Z_0q_1]$
00011	q_1	ε	ε	00011

Esercizio

Definire un automa a pila non deterministico che riconosca il linguaggio $\{w\tilde{w} \mid w \in \{a, b\}^*\}$ e costruire la grammatica che lo genera in base al metodo indicato nel teorema precedente.

Automi a pila deterministici

Un automa a pila **deterministico** è un automa a pila $\mathcal{M} = \langle \Sigma, \Gamma, Z_0, Q, q_0, F, \delta \rangle$ tale che, $\forall a \in \Sigma, \forall Z \in \Gamma, \forall q \in Q$

$$| \delta(q, a, Z) | + | \delta(q, \varepsilon, Z) | \leq 1$$

Una stringa è accettata da un automa a pila deterministico se e solo se essa dà luogo ad una computazione che termina in una configurazione $\langle q, \varepsilon, w \rangle$, con $q \in F$ e $w \in \Gamma^*$

Automi a pila deterministici

1. La definizione di automa a pila deterministico è una specializzazione della definizione di automa a pila non deterministico, quindi la classe dei linguaggi accettati da automi a pila non deterministici include quella dei linguaggi accettati da automi a pila deterministici.

Automi a pila deterministici

1. La definizione di automa a pila deterministico è una specializzazione della definizione di automa a pila non deterministico, quindi la classe dei linguaggi accettati da automi a pila non deterministici include quella dei linguaggi accettati da automi a pila deterministici.
2. La classe dei linguaggi di tipo 2, vale a dire dei linguaggi accettati da automi a pila non deterministici, non è chiusa rispetto alla complementazione.

Automi a pila deterministici

1. La definizione di automa a pila deterministico è una specializzazione della definizione di automa a pila non deterministico, quindi la classe dei linguaggi accettati da automi a pila non deterministici include quella dei linguaggi accettati da automi a pila deterministici.
2. La classe dei linguaggi di tipo 2, vale a dire dei linguaggi accettati da automi a pila non deterministici, non è chiusa rispetto alla complementazione.
3. Per ogni automa a pila deterministico \mathcal{M} , è possibile costruirne uno che accetta il linguaggio $\Sigma^* - L(\mathcal{M})$: quindi classe dei linguaggi accettati da automi a pila deterministici è chiusa rispetto alla complementazione

Automi a pila deterministici

1. La definizione di automa a pila deterministico è una specializzazione della definizione di automa a pila non deterministico, quindi la classe dei linguaggi accettati da automi a pila non deterministici include quella dei linguaggi accettati da automi a pila deterministici.
2. La classe dei linguaggi di tipo 2, vale a dire dei linguaggi accettati da automi a pila non deterministici, non è chiusa rispetto alla complementazione.
3. Per ogni automa a pila deterministico \mathcal{M} , è possibile costruirne uno che accetta il linguaggio $\Sigma^* - L(\mathcal{M})$: quindi classe dei linguaggi accettati da automi a pila deterministici è chiusa rispetto alla complementazione
4. Quindi, la classe dei linguaggi accettati da automi a pila deterministici dunque non coincide con quella dei linguaggi di tipo 2.

Automi a pila deterministici

Intuitivamente, un linguaggio separatore è dato da
 $L = \{w\tilde{w} \mid w \in \{a, b\}^+\}.$

Il linguaggio, di tipo 2, non può essere accettato da alcun automa a pila deterministico. Infatti, intuitivamente, durante la scansione dell'input non è possibile individuare a priori, in maniera deterministica, dove termina la stringa w ed inizia \tilde{w} .