UNIVERSITÀ
DEGLI STUDI
DI TORINO

Automi a pila

a.a. 2018-2019

Riferimenti bibliografici

Automi, Linguaggi e Calcolabilità, J. E. Hopcroft, R. Motwani, J. D. Ullman

Automi a pila [cap. 6]

- 6.1 Definizione di automa a pila.
  - 6.1.1 Introduzione informale.
  - 6.1.2 Definizione formale di automa a pila.
  - 6.1.3 Una notazione grafica per i PDA.
  - 6.1.4 Descrizioni istantanee di un PDA (fino all'Esempio 6.4 compreso).
- 6.2 I linguaggi di un PDA.
  - 6.2.1 Accettazione per stato finale (solo la definizione).
  - 6.2.2 Accettazione per stack vuoto.
  - 6.2.3 Da stack vuoto a stato finale (solo enunciato del Teorema 6.9).
  - 6.2.4 Da stato finale a stack vuoto (solo enunciato del Teorema 6.11).
- 6.3 Equivalenza di PDA e CFG.
  - 6.3.1 Dalle grammatiche agli automi a pila (Teorema 6.13 senza dimostrazione).
  - 6.3.2 Dai PDA alle grammatiche (solo enunciato del Teorema 6.14).
- 6.4 Automi a pila deterministici
  - 6.4.1 Definizione di PDA deterministico

Automi a pila

Un automa a pila (PDA) differisce da un automa finito, in particolare da un ε-NFA, per la presenza di una memoria a pila. In una transizione un PDA:

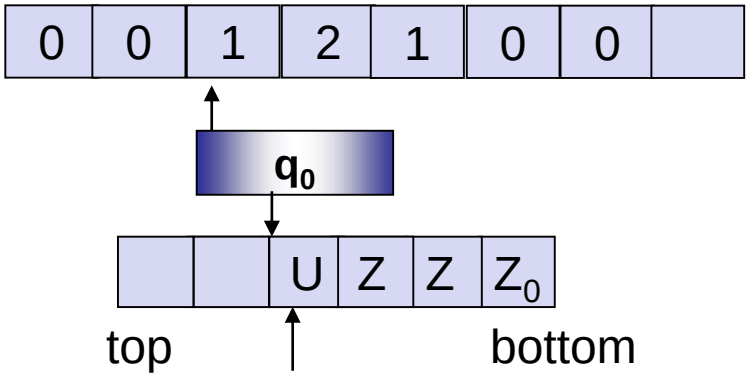
1. Consuma un simbolo di input o nessuno (mossa ε).
2. Il controllo transisce in un nuovo stato (o resta nello stato in cui si trova).
3. Sostituisce il simbolo sul top della pila con una stringa.

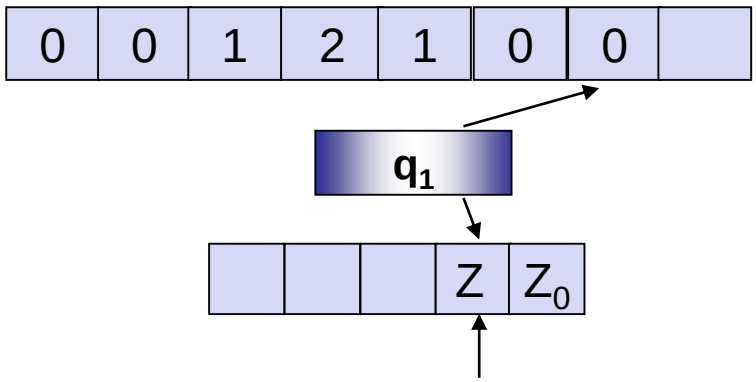
Automi a pila

Il linguaggio $\{w2w^R / w \in \{0,1\}^+\}$ non è regolare, non esiste un automa finito che lo riconosca: per controllare che la stringa di simboli che segue 2 sia la speculare di quella che precede 2 abbiamo bisogno di una memoria: lo stack serve a memorizzare la prima parte della stringa; per poter verificare che la seconda parte sia la stessa rovesciata si possono confrontare i simboli in input con i simboli sul top, che si possono via via eliminare.
È naturale pertanto pensare che una stringa appartenente al linguaggio faccia riaffiorare la base dello stack, di solito rappresentata dal simbolo Z_0.

Non dobbiamo però permettere che le due fasi di riempimento e di svuotamento dello stack si mescolino, cioè abbiamo bisogno di ricordare se si sta esaminando w o w^R: per questo non possiamo usare lo stack, ma basta usare uno stato diverso.

Tabella di transizione





La funzione di transizione del PDA per L_{w2w^R} descritta da una tabella:

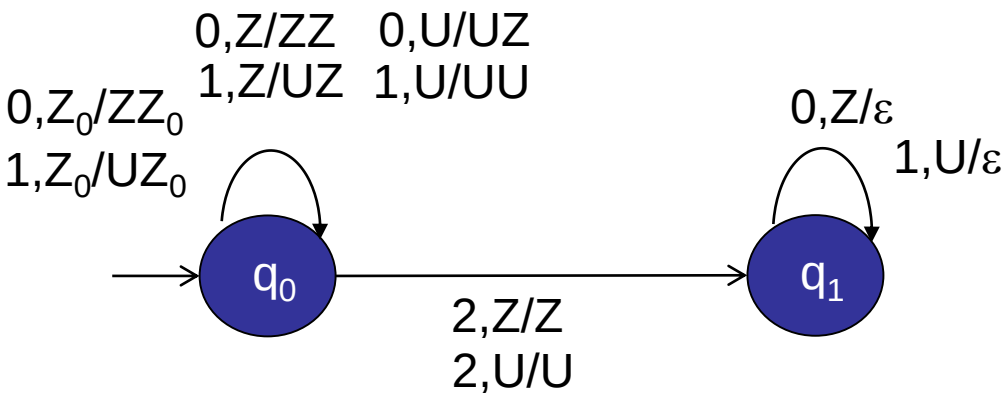
	$0,Z_0$	$1,Z_0$	$0,Z$	$0,U$	$1,Z$	$1,U$	$2,Z$	$2,U$
→ q_0	q_0,ZZ_0	q_0,UZ_0	q_0,ZZ	q_0,ZU	q_0,UZ	q_0,UU	q_1,Z	q_1,U
q_1			q_1,ε			q_1,ε		

Nota: Come per gli automi a stati finiti, → indica lo stato iniziale.

Diagramma di transizione

Un PDA può essere anche rappresentato come diagramma di transizione generalizzato.
Scriviamo "*a*,X/γ" per indicare che, leggendo il carattere *a* sulla stringa in input ed il carattere X in cima alla pila, si sostituisce la stringa γ a X in cima alla pila. All'inizio la pila contiene Z_0.

Il PDA per L_{w2w^R} descritto da un diagramma di transizione:



Automi a pila: definizione

Un *automa a pila* o *automa push-down* M è una 7-tupla

$$M = < Q, \Sigma, \Gamma, \delta, q_0, Z_0, F >$$

dove:

- Q è l'*insieme degli stati dell'unità di controllo* (finito e non vuoto)
- Σ è l'*alfabeto di ingresso*
- Γ è l'*alfabeto della pila*
- δ è la *funzione di transizione*
- $q_0 \in Q$ è lo *stato iniziale*
- $Z_0 \in \Gamma$ è il *simbolo iniziale della pila*
- $F \subseteq Q$ è l'*insieme degli stati finali*

Automi a pila

La funzione di transizione, in base allo stato del controllo, al simbolo sul nastro di input (a volte senza tenerne conto) e al simbolo sul top dello stack, specifica un insieme di coppie il cui primo elemento definisce lo stato del controllo dopo l'avanzamento della testina sul nastro di input (solo se il carattere è stato preso in considerazione) e il cui secondo elemento fornisce una stringa di caratteri dell'alfabeto dello stack che viene sostituita al carattere sul top. L'automa può quindi fare *due tipi di transizioni* che si possono chiamare *mossa con lettura* e *mossa spontanea*.

$$\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$$

$$\delta(q, a, Z) = \{(p_1, \gamma_1), (p_2, \gamma_2), \ldots, (p_k, \gamma_k)\} \qquad a \in \Sigma \cup \{\varepsilon\}$$

nello stato *q*, leggendo *a* (o nulla) sulla stringa e Z sulla pila, l'automa entra *non deterministicamente* nello stato p_i, avanza la testina sul nastro di input (non avanza nel caso di mossa spontanea) ed esegue pop, push(γ_i) sulla pila.

Nota: L'automa a pila (o push-down) definito è non deterministico.

Automi a pila

Modifichiamo ora il linguaggio L_{w2w^R} eliminando il carattere 2:

$$L_{ww^R} = \{ww^R \mid w \in \{0,1\}^+\}$$

Cosa differenzia un PDA per questo linguaggio da quello per il linguaggio precedente?

Un PDA per L_{ww^R} deve fare i conti con il fatto che non è in grado di riconoscere il punto in cui inizia la seconda metà della stringa: una soluzione è quella di «provare» due alternative tutte le volte che potrebbe essere stata esaminata la prima metà (la stringa w).

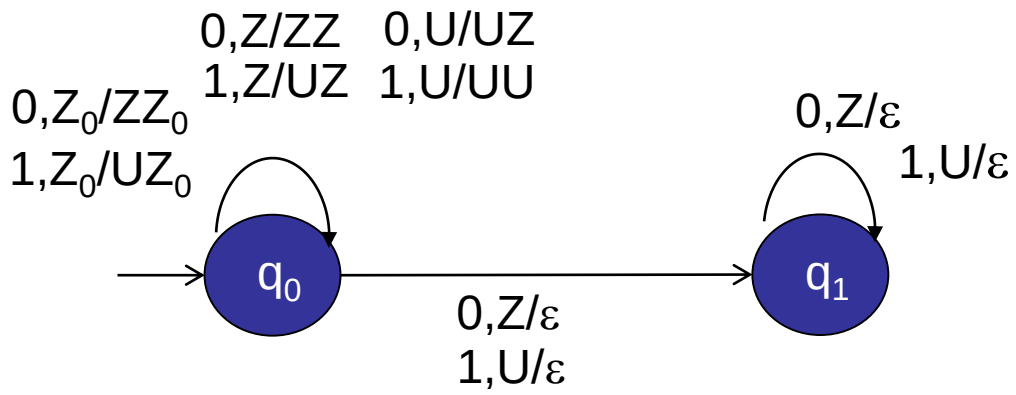
Tutte le volte che in input si presenta uno 0 preceduto da uno 0 o un 1 preceduto da un 1 potrebbe iniziare w^R e l'automa deve avere sia la possibilità di continuare a riempire la pila, sia quella di passare alla fase di svuotamento.

Automa a pila riconoscitore del linguaggio L_{ww^R}

La tabella di transizione :

	$0,Z_0$	$1,Z_0$	$0,Z$	$0,U$	$1,Z$	$1,U$
→ q_0	q_0,ZZ_0	q_0,UZ_0	q_0,ZZ q_1,ε	q_0,ZU	q_0,UZ	q_0,UU q_1,ε
q_1			q_1,ε			q_1,ε

Il diagramma di transizione:



Automi a pila: configurazione istantanea

Configurazione istantanea: (q, y, η)

q è lo stato del controllo
y è la parte della stringa ancora da esaminare
η è il contenuto della pila

Configurazione iniziale: (q_0, w, Z_0)

Transizioni da una configurazione ad un'altra:

$(q, ay, Z\eta) \vdash (p, y, \gamma\eta)$ se $(p, \gamma) \in \delta(q, a, Z)$

$(q, ay, Z\eta) \vdash (p, ay, \gamma\eta)$ se $(p, \gamma) \in \delta(q, \varepsilon, Z)$

Indichiamo con $\vdash^*$ la relazione ottenuta per chiusura riflessiva e transitiva della relazione di transizione $\vdash$ da una configurazione ad un'altra.

Automi a pila: linguaggio riconosciuto

Negli automi a pila si hanno due definizioni di *stringa accettata* o *riconosciuta*: per «stato finale» e per «stack vuoto».

Una stringa w è accettata da un automa a pila:

- per *stack vuoto* se $(q_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon)$
- per *stato finale* se $(q_0, w, Z_0) \vdash^* (q, \varepsilon, \eta)$ & $q \in F$

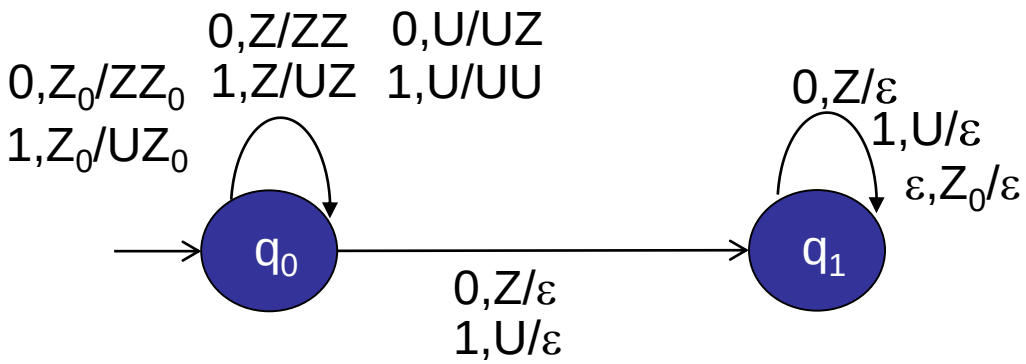
Il *linguaggio riconosciuto* dall'automa $M = < Q, \Sigma, \Gamma, \delta, q_0, Z_0, \Phi >$ *per stack vuoto* è l'insieme delle stringhe w il cui esame, a partire dalla configurazione iniziale (q_0, w, Z_0), può portare l'automa M in una configurazione in cui la memoria è vuota:

$$N(M) = \{ w \mid (q_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon)\}$$

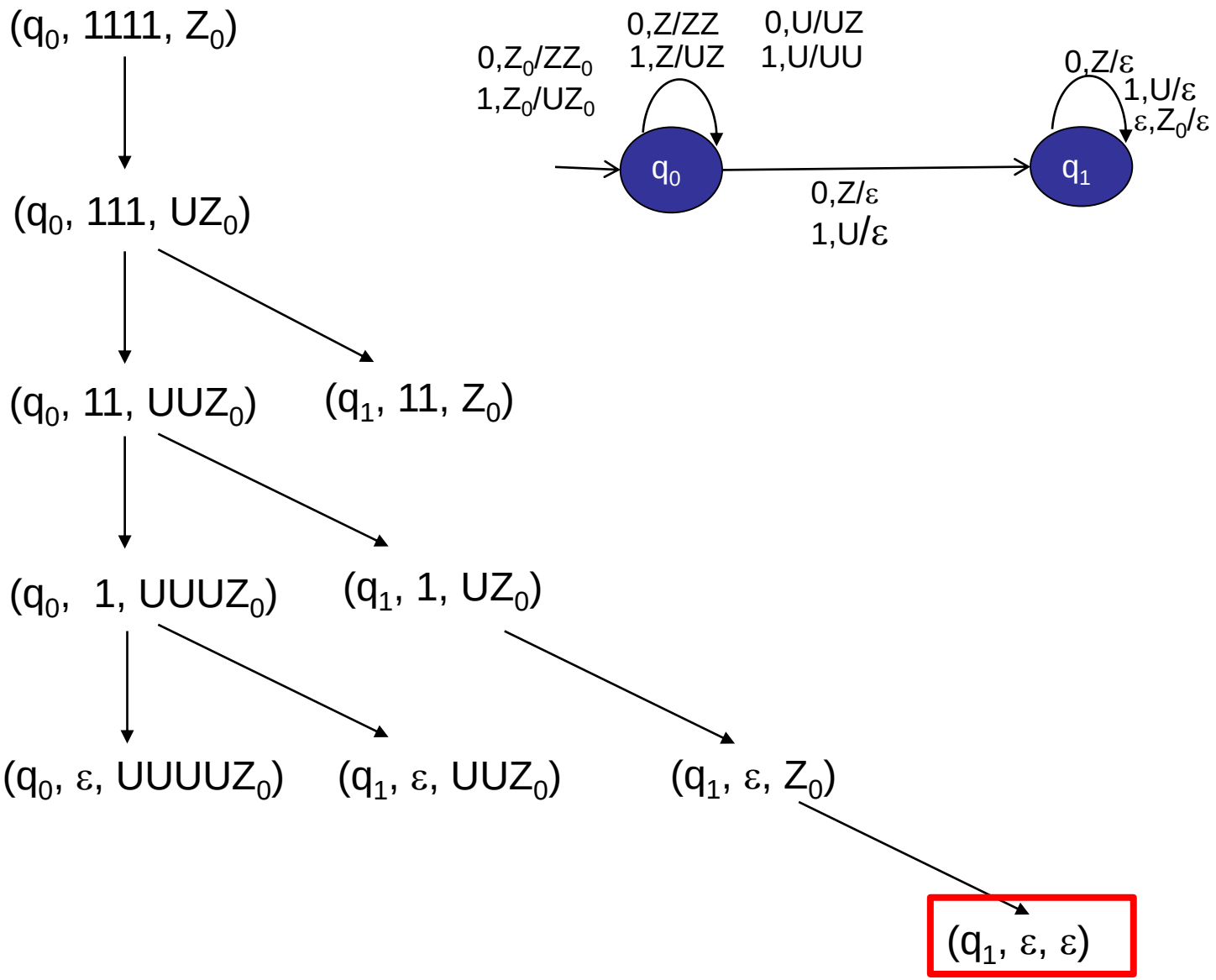
Nota: Se si considera il linguaggio accettato per stack vuoto, l'insieme degli stati finali è inessenziale e quindi può essere definito come l'insieme vuoto ($F = \Phi$).

Esempio

Data la definizione di linguaggio riconosciuto o accettato per stack vuoto da un automa a pila, dobbiamo modificare gli automi precedenti in modo da svuotare la pila in presenza di una stringa del linguaggio che si vuole riconoscere.
Per il linguaggio L_{ww^R} basta aggiungere nello stato q_1 una ε-mossa che cancella Z_0.



Esempio


$(q_0, 1111, Z_0)$
$(q_0, 111, UZ_0)$
$(q_0, 11, UUZ_0)$
$(q_1, 11, Z_0)$
$(q_0, 1, UUUZ_0)$
$(q_1, 1, UZ_0)$
$(q_0, \varepsilon, UUUUZ_0)$
$(q_1, \varepsilon, UUZ_0)$
(q_1, ε, Z_0)
$(q_1, \varepsilon, \varepsilon)$
$0,Z_0/ZZ_0$
$1,Z_0/UZ_0$
0,Z/ZZ
1,Z/UZ
0,U/UZ
1,U/UU
q_0
q_1
0,Z/ε
1,U/ε
ε,Z_0/ε
0,Z/ε
1,U/ε

Relazione tra grammatiche libere e automi a pila

Un linguaggio è generato da una CFG se e solo se è accettato da un PDA per pila vuota.

Vediamo la costruzione di un automa a pila dalla grammatica, mentre omettiamo la costruzione di una grammatica a partire dall'automa.

Data una grammatica G, costruiamo un PDA che simula le derivazioni left-most di G.

Ad esempio, consideriamo la grammatica:

$$G = (\{S,A\}, \{0,1\}, \{S \to 0S1 \mid 0A1, A \to 2A \mid \varepsilon\}, S)$$

e la derivazione:

$$S \Rightarrow 0S1 \Rightarrow 00A11 \Rightarrow 002A11 \Rightarrow 00211$$

Esempio

IDEA: usare lo stack per simulare una derivazione leftmost

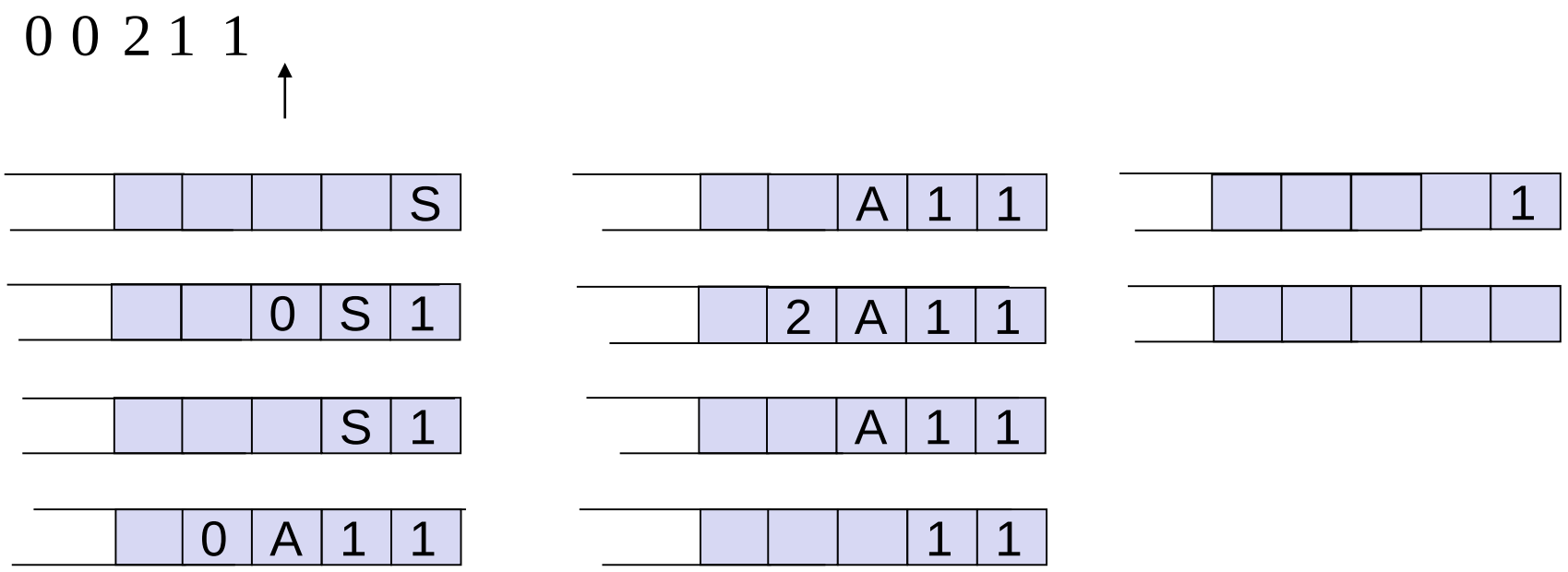
- sul nastro di input si scrive la stringa da esaminare: 00211 e sulla pila il simbolo iniziale della grammatica, S;
- se S viene sostituito con il corpo della produzione $S \rightarrow 0S1$, il simbolo sul top dello stack risulta uguale al simbolo in input;
- tale simbolo può allora essere "consumato" avanzando sul nastro ed eliminando il top dello stack;
- continuando in questo modo si arriva a rimuovere A dallo stack, usando la riscrittura $A \rightarrow \varepsilon$;
- sulla pila restano due 1, come sul nastro di input;
- gli 1 in input trovano corrispondenza con il contenuto della pila; il controllo avanza sul nastro svuotando contemporaneamente la pila.

Relazione tra grammatiche libere e automi a pila

Dalla grammatica all'automa push-down

Esempio: $G = \langle \{S,A\}, \{0,1\}, \{S \rightarrow 0S1 \mid 0A1,\ A \rightarrow 2A \mid \varepsilon\}, S \rangle$

$S \Rightarrow 0S1 \Rightarrow 00A11 \Rightarrow 002A11 \Rightarrow 00211$



Dalla grammatica all'automa

Costruzione dell'automa dalla grammatica:

1. Definire lo start symbol come simbolo iniziale dello stack.

2. Per ogni produzione del tipo $A \rightarrow YX_1 \ldots X_n$ ($A \rightarrow \varepsilon$), se A è il simbolo sul top dello stack, inserire sullo stack la stringa $YX_1 \ldots X_n$ (ε) al posto della variabile A, senza avanzare sulla stringa in input:

   $(q, YX_1 \ldots X_n) \in \delta(q, \varepsilon, A)$ ($(q, \varepsilon) \in \delta(q, \varepsilon, A)$).

3. Per ogni simbolo terminale *a*, se *a* è il simbolo in input e *a* è il simbolo sul top dello stack, eliminare il simbolo sul top dello stack ed avanzare la testina di lettura:
   $\delta(q, a, a) = \{(q, \varepsilon)\}$

Nota: L'automa costruito dall'algoritmo ha un solo stato

Esempio

Applichiamo l'algoritmo all'esempio precedente:

$S \rightarrow 0S1 \mid 0A1$

$A \rightarrow 2A \mid \varepsilon$

$S \rightarrow 0S1$
$S \rightarrow 0A1$

$\delta(q, \varepsilon, S) = \{(q, 0S1), (q, 0A1)\}$

$A \rightarrow 2A$
$A \rightarrow \varepsilon$

$\delta(q, \varepsilon, A) = \{(q, 2A), (q, \varepsilon)\}$

$\delta(q, 0, 0) = \delta(q, 1, 1) = \delta(q, 2, 2) = \{(q, \varepsilon)\}$

Automi a pila deterministici

Un automa a pila deterministico **M** è una 7-tupla

$$<Q, \Sigma, \Gamma, \delta, q_0, Z_0, F>$$

dove la funzione di transizione: $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$

definisce per ogni terna: stato, simbolo in input, simbolo sullo stack, al massimo una transizione e non offre mai la scelta tra una mossa spontanea e una mossa con lettura, cioè:

- $|\ \delta(q, a, Z)\ | \leq 1$
- $|\ \delta(q, \varepsilon, Z)\ | \leq 1$
- se per una coppia (q, Z) è definita una mossa spontanea, non è definita nessuna mossa con lettura.

Automi a pila deterministici

Esempio: $N(M) = \{xbx^R \mid x \in \{r, v\}^*\}$

$Q = \{q_0, q_1\}$ $\quad \Sigma = \{r, b, v\}$ $\quad \Gamma = \{Z_0, R, V\}$ $\quad F = \Phi$

stato iniziale q_0, simbolo iniziale dello stack Z_0

$\delta(q_0, b, Z_0) = \{(q_1, Z_0)\}$
$\delta(q_0, v, Z_0) = \{(q_0, VZ_0)\}$
$\delta(q_0, r, Z_0) = \{(q_0, RZ_0)\}$
$\delta(q_0, v, R) = \{(q_0, VR)\}$
$\delta(q_0, r, V) = \{(q_0, RV)\}$
$\delta(q_0, r, R) = \{(q_0, RR)\}$

$\delta(q_0, v, V) = \{(q_0, VV)\}$
$\delta(q_0, b, V) = \{(q_1, V)\}$
$\delta(q_0, b, R) = \{(q_1, R)\}$
$\delta(q_1, r, R) = \{(q_1, \varepsilon)\}$
$\delta(q_1, v, V) = \{(q_1, \varepsilon)\}$
$\delta(q_1, \varepsilon, Z_0) = \{(q_1, \varepsilon)\}$

$(q_0, rvbvr, Z_0) \vdash (q_0, vbvr, RZ_0) \vdash (q_0, bvr, VRZ_0) \vdash$
$\vdash (q_1, vr, VRZ_0) \vdash (q_1, r, RZ_0) \vdash (q_1, \varepsilon, Z_0) \vdash (q_1, \varepsilon, \varepsilon)$

Automi a pila deterministici

Proprietà

Se un linguaggio è riconosciuto da un automa push-down deterministico, esiste una grammatica non ambigua che lo genera.

Non è vero il viceversa:
Un linguaggio non inerentemente ambiguo (generato cioè da almeno una grammatica non ambigua) non è detto che abbia un riconoscitore deterministico.

Ad esempio il linguaggio $\{xx^R / x \in \{r,v\}^*\}$ è generato dalla grammatica non ambigua con le seguenti produzioni:

$$S \rightarrow rSr \mid vSv \mid \varepsilon$$

ma non ha nessun riconoscitore deterministico.

Teorema
La classe dei linguaggi riconosciuti da automi a pila deterministici è un *sottinsieme proprio* dei linguaggi riconosciuti dagli automi a pila.

Esercizi

Costruire un automa a pila per le grammatiche libere dal contesto che generano i seguenti linguaggi:

- $\{a^n b^m c^p \mid n = m + p \ \& \ m, p \geq 0\}$
- $\{ a^n c b^n \mid n > 0 \}$
- $\{(ab)^n (cd)^n \mid n > 0 \}$
- $\{ a^n b^m c^m d^n \mid n, m \geq 0 \}$
- $\{ a^i b^j \mid 0 \leq i < j \}$
- $\{ a^{2i} b^j c^i \mid i, j \geq 0 \}$
- $\{ a^n b^m c^p d^q \mid n + m = p + q \quad e \quad n, m, p, q > 0 \}$

Esercizi

- Costruire un automa a pila che riconosca il linguaggio generato dalla seguente grammatica:

  $S \rightarrow BC \mid AD$, $A \rightarrow aA \mid \varepsilon$, $B \rightarrow aBb \mid \varepsilon$, $C \rightarrow cC \mid \varepsilon$,

  $D \rightarrow bDc \mid \varepsilon$

- Dire se le stringhe ac, aabbc, bbbccc appartengono al linguaggio generato dalla grammatica precedente. In caso positivo costruire l'albero di derivazione, in caso negativo mostrare come si comporta l'automa costruito.

- Definire un automa a pila per il riconoscimento del linguaggio $L = \{a^i b^{2j} \mid i > 0, j > 0\}$.
  Scegliere, per la definizione dell'automa, se basarsi o meno sulla grammatica.
  Esiste un automa deterministico che riconosce il linguaggio? Argomentare la risposta.

Esercizi

- Costruire due automi che riconoscano rispettivamente i linguaggi:
  $\{wbw^R \mid w \in \{r, v\}\}$ e $\{(rv)^n b (vr)^m \mid n, m \geq 0\}$

- Definire automi a pila che riconoscano i linguaggi:
  $\{a^n b^n \mid n \geq 0\} \cup \{a^n b^{2n} \mid n \geq 0\}$
  $\{0^n 1^m 2^h \mid n, m, h > 0 \text{ e } h = n + m + 2\}$.
  Scegliere, per la definizione degli automi, se basarsi o meno sulla grammatica.

- Dire qual è il linguaggio generato dalla grammatica con le seguenti produzioni:
  $S \rightarrow aSb \mid bSc \mid b$
  e costruire un automa che lo riconosca.