Indice

1 Automi a pila **2**
1.1 Definizioni . . . 3
1.2 Equivalenza tra le due nozioni di accettazione nel modello nondeterministico . . . 8
1.2.1 Da stati finali a pila vuota . . . 8
1.2.2 Da pila vuota a stati finali . . . 9

1 Automi a pila

Gli automi a pila sono automi che oltre ad avere un controllo a stati finiti hanno una memoria arbitrariamente grande, organizzata come una pila, quindi non si può accedere liberamente alle informazioni in questa. Questo modello è one-way sul nastro di input (la versione two-way è più potente).

Inizieremo subito con vedere il modello non deterministico.

Un automa a pila è una tupla

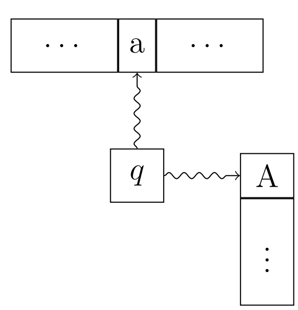
$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

- Γ l'alfabeto della pila o alfabeto di lavoro
- δ funzione di transizione
- $q_0 \in Q$, lo stato iniziale dell'automa
- $Z_0 \in \Gamma$, lo stato iniziale della pila
- F un insieme di stati finali

La funzione di transizione dipenderà quindi da tre cose: dallo stato corrente, dal simbolo dell'input corrente e dallo stato corrente dello stato

$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to \mathrm{PF}(Q \times \Gamma^*)$$

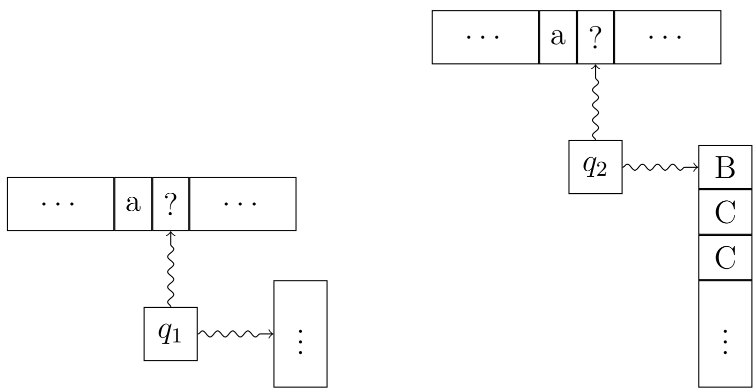
Quindi la funzione di transizione cambia lo stato dell'automa e rimpiazza il simbolo in cima alla pila con una stringa di stati della pila. $\mathrm{PF}(-)$ sta per le parti finite, infatti se utilizzassimo $2^{Q \times \Gamma^*}$ potremmo avere programmi infiniti, visto che Γ^* è un insieme infinito. $\Sigma \cup \epsilon$ perché sono contemplate mosse in base allo stato dell'automa che modificano la pila senza leggere un simbolo in input.



Nota 1. *Per convenzione la stringa di stati viene messa sulla pila di destra a sinistra, quindi il simbolo più a sinistra sarà in cima alla pila.*

Visto che ammettiamo ϵ-mosse il modello di sopra potrebbe non esaurire tutte le possibilità. Supponiamo che la funzione δ sia definita

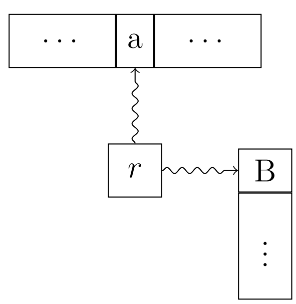
$$\delta(q, a, A) = \{(q_1, \epsilon), (q_2, BCC)\}$$



(a) Lo stato per (q_1, ϵ) (b) Lo stato per (q_2, BCC)

Inoltre potremmo anche avere ϵ-mosse, ad esempio

$$\delta(q, \epsilon, A) = \{(r, B)\}$$



1.1 Definizioni

Ci riferiremo agli automi a pila come PDA (Push Down Automaton) e assumeremo che siano sempre nondeterministici, a meno che diversamente specificato.

Chiameremo lo stato complessivo dell'automa a pila la sua **configurazione**, questa verrà rappresentata compattamente come una tupla dello stato, la porzione di input ancora da leggere da quello corrente, e il contenuto della pila. Quindi è rappresentato dalla configurazione

$$(q, ay, A\alpha)$$

con $q \in Q, a \in \Sigma \cup \{\epsilon\}, y \in \Sigma^*, A \in \Gamma, \alpha \in \Gamma^*$.

Una mossa, indica che da una configurazione q posso passare ad un'altra p, scritto

$$q \vdash p$$

Ad esempio sopra abbiamo che

$$\begin{aligned}(q, ay, A\alpha) &\vdash (q_1, y, \alpha)\\ (q, ay, A\alpha) &\vdash (q_2, y, BCC\alpha)\\ (q, ay, A\alpha) &\vdash (r, ay, B\alpha)\end{aligned}$$

Più rigorosamente, sia $(q, ay, Z\alpha)$ la configurazione corrente, con $Z \in \Gamma$ e M l'automa a pila, diciamo che

$$(q, ay, Z\alpha) \underset{M}{\vdash} (p, y, \beta\alpha)$$

sse $(p, \beta) \in \delta(q, a, Z)$ dove $q, p \in Q$, $y \in \Sigma^*$, $a \in \Sigma \cup \{\epsilon\}$, $Z \in \Gamma$ e $\alpha, \beta \in \Gamma^*$. Se l'automa è ovvio dal contensto possiamo ometterlo da $\underset{M}{\vdash}$ e scrivere solo $\vdash$.

Da una configurazione C' arrivo ad una configurazione C'' in un certo numero di mosse – scritto

$$C' \overset{*}{\underset{M}{\vdash}} C''$$

sse esistono $C_0, \ldots, C_k$ con $C_0 = C'$ e $C_k = C''$ e $\forall i \in 1, \ldots, k\ C_{i-1} \underset{M}{\vdash} C_i$.

La configurazione iniziale di un automa su input $w \in \Sigma^*$ è

$$(q_0, w, Z_0)$$

Per accettare possiamo dare alcune diverse definizioni di configurazione accettante:

- una volta finito l'input mi trovo in uno stato $q \in F$ e la pila può essere una stringa qualunque, questa è detta *accettazione per stati finali*, ed indichiamo il linguaggio accettato per stati finali dall'automa a pila M come

$$L(M) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \overset{*}{\vdash} (q, \epsilon, \gamma), q \in F, \gamma \in \Gamma^*\}$$

Nota 2. *Siccome sono accettate le ϵ mosse può esserci il caso in arriviamo alla fine dell'input con uno stato non finale, e si può fare una ϵ-mossa ed arrivare ad uno stato finale.*

- è ragionevole pensare che tutto quello che viene messo sulla pila debba anche essere tolto, questa è detta *accettazione per pila vuota* per cui si deve arrivare alla fine dell'input ed aver svuotato l'intera pila, ingorando lo stato. Il linguaggio accettato per pila vuota dall'automa M lo indichiamo come

  $$N(M) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \overset{*}{\vdash} (q, \epsilon, \epsilon), q \in Q\}$$

  In questo caso ovviamente si può omettere F dalla definizione dell'automa.

- si può pensare di richiedere entrambe le precedenti, come vedremo più avanti queste tre nozioni sono equivalenti (nel caso nondeterministico)

Nota 3. *Questa cosa la vedremo meglio, ma visto che la pila è la struttura fondamentale per la ricorsione, i linguaggi CF sono i linguaggi regolari a cui è stata aggiunta la ricorsione.*

Definiamo il linguaggio

$$\mathcal{L} = \{a^n b^n \mid n \geq 1\}$$

possiamo usare la pila per contare il numero di a.

$$\delta(q_0, a, Z_0) = \{(q_0, A)\}$$
$$\delta(q_0, a, A) = \{(q_0, AA)\}$$
$$\delta(q_0, b, A) = \{(q_1, \epsilon)\}$$
$$\delta(q_1, b, A) = \{(q_1, \epsilon)\}$$

E vale che data questa δ

$$\mathcal{L} = N(M)$$

Questo è un caso particolare di automa a pila in cui utilizziamo in simbolo solo (cioè A, oltre a Z_0) è detto *automa a contatore*.
Definiamo alternativamente

$$\delta(q_0, a, Z_0) = \{(q_0, AZ_0)\}$$
$$\delta(q_0, a, A) = \{(q_0, AA)\}$$
$$\delta(q_0, b, A) = \{(q_1, \epsilon)\}$$
$$\delta(q_1, b, A) = \{(q_1, \epsilon)\}$$
$$\delta(q_1, \epsilon, Z_0) = \{(q_F, \epsilon)\}$$

con $F = \{q_F\}$, e vale che con questa δ

$$\mathcal{L} = L(M)$$

Vediamo ora il caso di sopra, ma in cui

$$\mathcal{L} = \{a^n b^n \mid n \geq 0\}$$

possiamo usare la pila per contare il numero di a.

$$\begin{gathered}\delta(q_0, \epsilon, Z_0) = \{(q_0, \epsilon)\} \\ \delta(q_0, a, Z_0) = \{(q_0, A)\} \\ \delta(q_0, a, A) = \{(q_0, AA)\} \\ \delta(q_0, b, A) = \{(q_1, \epsilon)\} \\ \delta(q_1, b, A) = \{(q_1, \epsilon)\}\end{gathered}$$

E vale che data questa δ in cui si può direttamente accettare dallo stato q_0

$$\mathcal{L} = N(M)$$

L'introduzione della prima regola è problematica, perché con input non vuoto permette di svuotare la pila da Z_0, bloccando la continuazione dell'automa, quindi abbiamo introdotto il nondeterminismo tra le due regole $\delta(q_0, \epsilon, Z_0)$ e $\delta(q_0, a, Z_0)$.
Definiamo similmente a sopra

$$\begin{gathered}\delta(q_0, \epsilon, Z_0) = \{(q_F, \epsilon)\} \\ \delta(q_0, a, Z_0) = \{(q_0, AZ_0)\} \\ \delta(q_0, a, A) = \{(q_0, AA)\} \\ \delta(q_0, b, A) = \{(q_1, \epsilon)\} \\ \delta(q_1, b, A) = \{(q_1, \epsilon)\} \\ \delta(q_1, \epsilon, Z_0) = \{(q_F, \epsilon)\}\end{gathered}$$

con $F = \{q_F\}$, e vale che con questa δ

$$\mathcal{L} = L(M)$$

sempre introducendo non determinismo.
Questa versione, a differenza di quello di sopra per pila vuota, può anche essere fatta senza non determinismo infatti definiamo q_I come nuovo stato iniziale che è anche finale, se la stringa è vuota possono direttamente accettare, mentre

$$\delta(q_I, a, Z_0) = \{(q_0, AZ_0)\}$$

ci riconduce all'automa di sopra.

Supponiamo invece di voler riconoscere

$$L = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$$

bisogna contare sia le a che le b: finché si incontrano a si aggiungono alla pila, e appena si incontrano b le si tolgono. Se durante questa operazione si svuota la pila iniziamo a contare le b e facciamo la operazione inversa. Queste due operazioni continuano ad alternarsi finché la stringa non è svuotato.

Definiamo ora l'automa a pila deterministico. Questo in ogni configurazione permette una singola scelta:

- sono vietate configurazioni che ammettono una mossa e una ϵ-mossa, quindi $\forall q \in Q, z \in \Gamma$ se $\delta(q, \epsilon, z) \neq \varnothing$ allora $\forall a \in \Sigma \delta(q, a, Z) = \varnothing$
- $\forall q \in Q, z \in \Gamma, a \in \Sigma \cup \{\epsilon\} \mid |\delta(q, a, Z)| \leq 1$, quindi per ogni tripletta q, a, Z è ammessa al massimo una mossa.

A questo punto abbiamo definito quattro modelli: deterministico e nondeterministico che possono accettare per pila vuota o per stato finale. Vedremo che il caso nondeterminismo in questo caso è più potente del caso deterministico, e che nel modello nondeterministico automi che possono accettare per pila vuota o per stato finale sono equivalenti.

1.2 Equivalenza tra le due nozioni di accettazione nel modello nondeterministico

1.2.1 Da stati finali a pila vuota

Dato un automa $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ e supponiamo che $L = L(M)$ sia il linguaggio accettato per stati finali. Definiamo l'automa

$$M' = (Q \cup \{q_0', q_e\}, \Sigma, \Gamma \cup \{X\}, \delta', q_0', X, \varnothing)$$

con $q_0', q_e \notin Q$ e $X \notin \Gamma$, vogliamo che $L = N(M')$.

Ad alto livello quando M arriva in uno stato finale, M' si sposta nello stato q_e in cui inizia a svuotare la pila. Infatti la e di q_e sta per "empty".

Definiamo ora δ':

1. prima di tutto
$$\delta'(q_0', \epsilon, X) = \{(q_0, Z_0X)\}$$
questo serve solo ad infilare X in fondo alla pila. La X è necessaria per evitare che se l'automa iniziale M svuota la pila si accetti la stringa.

2. per ogni altra cosa M' si può comportare come M:
$$\forall q \in Q, a \in \Sigma \cup \{\epsilon\}, z \in \Gamma \mid \delta(q, a, Z) \subseteq \delta'(q, a, Z)$$

3. ogni qualvolta M entra in uno stato finale M' può – enfasi su può – iniziare a svuotare l'intera pila:
$$\forall q \in F, z \in \Gamma \cup \{X\} \mid (q_e, \epsilon) \in \delta'(q, \epsilon, Z)$$

4. una volta entrato nello stato di svuotamento, continua a svuotare:
$$\forall z \in \Gamma \cup \{X\} \mid \delta'(q_e, \epsilon, Z) = \{(q_e, \epsilon)\}$$

Questo necessariamente introduce nondeterminismo, infatti l'automa M potrebbe entrare in uno stato finale prima di essere arrivato alla fine della stringa. Ed anche se l'automa di partenza è deterministico il punto 3 potrebbe in ogni caso introdurre nondeterminismo.

Supponiamo di avere un automa deterministico che accetta la stringa w a pila vuota, allora ogni stringa che ha w come prefisso non può essere accettata, perché il prefisso w svuoterebbe la pila e un automa con pila vuota non può andare a avanti. Quindi il nondeterminismo è in un certo senso necessario per automi a pila che accettano con pila vuota.

1.2.2 Da pila vuota a stati finali

Dato un automa $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \varnothing)$ che accetta per pila vuota il linguaggio $L = N(M)$, vogliamo creare un automa che accetti per stati finali. Sia questo
$$M' = (Q \cup \{q_0', q_F\}, \Sigma, \Gamma \cup \{X\}, \delta', q_0', X, F = \{q_F\})$$
con $q_0', q_F \notin Q, X \notin \Gamma$.

Definiamo ora δ':

- come prima inizialmente infiliamo X in fondo alla pila:

$$\delta'(q_0', \epsilon, X) = \{(q_0, Z_0 X)\}$$

  X serve a riconoscere quando la pila è vuota.

- a questo punto copiamo tutte le mosse di M, per cui

$$\forall q \in Q, a \in \Sigma \cup \{\epsilon\}, Z \in \Gamma \mid \delta'(q, a, Z) = \delta(q, a, Z)$$

- nel momento in cui M svuota la prima, M' si trova X sulla pila, a questo punto può entrare in uno stato finale

$$\forall q \in Q \mid \delta'(q, \epsilon, X) = \{(q_F, \epsilon)\}$$

Supponendo che M sia deterministico, M' rimane deterministico – la trasformazione preserva il determinismo.