Azzolini Riccardo 2020-12-03

PDA — Confronto tra accettazione per stato finale e per stack vuoto

1 Nozioni di accettazione

Si è visto in precedenza che, dato un PDA $P = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle$, si definiscono le nozioni di accettazione

- per stato finale:

$$L(P) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \stackrel{*}{\vdash} (p, \epsilon, \alpha) \text{ con } p \in F\}$$

- per stack vuoto:

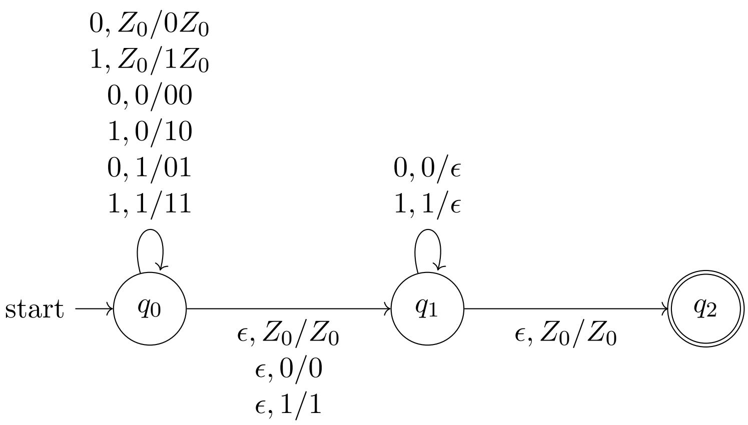
$$N(P) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \stackrel{*}{\vdash} (p, \epsilon, \epsilon)\}$$

e in generale $L(P) \neq N(P)$, ma

- per ogni PDA P, esiste un PDA P' tale che $N(P') = L(P)$;
- per ogni PDA P, esiste un PDA P' tale che $L(P') = N(P)$.

2 Esempio

Come esempio, si consideri l'automa P_{ww^R}:



Esso è stato costruito (intuitivamente) in modo da accettare per stato finale il linguaggio dei palindromi di lunghezza pari su $\{0, 1\}$:

$$L(P_{ww^R}) = \{ww^R \mid w \in \{0, 1\}^*\}$$

(ciò potrebbe essere dimostrato formalmente). Invece, il linguaggio riconosciuto per stack vuoto è $N(P_{ww^R}) = \varnothing$, perché nessuna transizione permette di eliminare Z_0 dallo stack, dunque nessuna computazione può condurre a una configurazione in cui lo stack sia vuoto. Allora, $L(P_{ww^R}) \neq N(P_{ww^R})$: questo è un esempio di PDA per cui le due nozioni di accettazione producono linguaggi diversi. Tuttavia, come già detto, esiste un altro PDA P'_{ww^R} che accetta per stack vuoto il linguaggio accettato per stato finale da P_{ww^R}, ovvero tale che

$$N(P'_{ww^R}) = L(P_{ww^R}) = \{ww^R \mid w \in \{0, 1\}^*\}$$

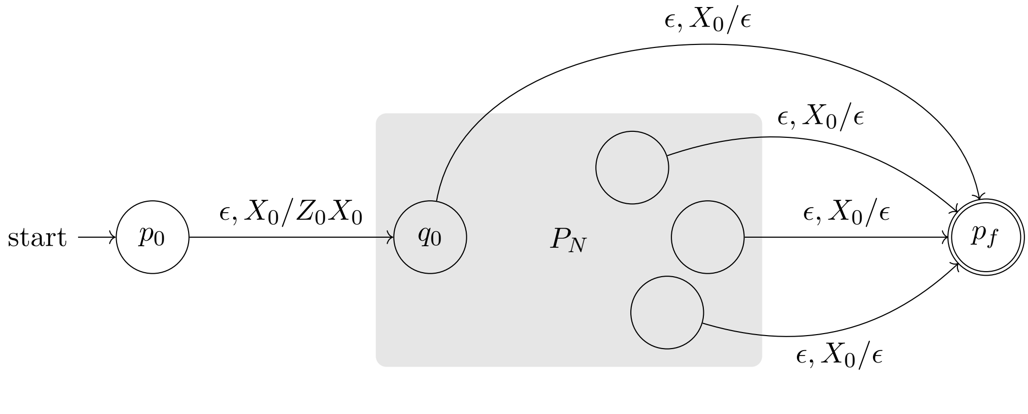
Per costruire l'automa P'_{ww^R} a partire da P_{ww^R}, è sufficiente modificare la funzione di transizione δ sostituendo $\delta(q_1, \epsilon, Z_0) = \{(q_2, Z_0)\}$ con $\delta(q_1, \epsilon, Z_0) = \{(q_2, \epsilon)\}$, in modo da eliminare l'unico simbolo (Z_0) presente sullo stack quando si passa da q_1 allo stato finale q_2. Si potrebbe dimostrare che, con questa costruzione:

- se $w \in L(P_{ww^R})$ allora $w \in N(P'_{ww^R})$;
- se $w \in N(P'_{ww^R})$ allora $w \in L(P_{ww^R})$.

3 Da stack vuoto a stato finale

Teorema: Dato un PDA $P_N = \langle Q, \Sigma, \Gamma, \delta_N, q_0, Z_0 \rangle$ che accetta per stack vuoto il linguaggio $N(P_N)$, esiste un PDA P_F che accetta lo stesso linguaggio per stato finale, cioè tale che $L(P_F) = N(P_N)$.

L'idea della costruzione di P_F è la seguente:



- Si aggiungono a P_N un nuovo stato iniziale p_0 e un nuovo simbolo iniziale di stack X_0. L'unica transizione uscente da p_0, che sarà per forza la prima mossa compiuta dall'automa, è fatta in modo tale da non consumare input (è un'ϵ-mossa), mettere Z_0 nello stack sopra X_0, e porta l'automa in q_0:

$$(p_0, w, X_0) \vdash (q_0, w, Z_0X_0)$$

- Si aggiunge uno nuovo stato finale p_f, e a ogni stato di P_N si aggiunge un'ϵ-mossa che porta in p_f se il simbolo in cima allo stack è X_0 (e in tal caso, essendo anche il simbolo in fondo, X_0 è sicuramente l'unico simbolo sullo stack).

Formalmente, l'automa P_F è definito come

$$P_F = \langle Q \cup \{p_0, p_f\}, \Sigma, \Gamma \cup \{X_0\}, \delta_F, p_0, X_0, \{p_f\}\rangle$$

dove δ_F è tale che:

$$\begin{aligned} & \delta_F(p_0, \epsilon, X_0) = \{(q_0, Z_0X_0)\} \\ \forall q \in Q,\ \forall a \in \Sigma_\epsilon,\ \forall Y \in \Gamma \qquad & \delta_F(q, a, Y) = \delta_N(q, a, Y) \\ \forall q \in Q \qquad & \delta_F(q, \epsilon, X_0) = \{(p_f, \epsilon)\} \end{aligned}$$

Una computazione accettante di P_N è del tipo

$$(q_0, w, Z_0) \overset{*}{\vdash} (p, \epsilon, \epsilon)$$

e per la proprietà (P2) delle computazioni essa può essere simulata nel nuovo automa P_F, aggiungendo all'inizio l'ϵ-mossa da p_0 a q_0,

$$(p_0, w, X_0) \vdash (q_0, w, Z_0X_0) \overset{*}{\vdash} (p, \epsilon, X_0)$$

e infine si sfrutta una delle ϵ-mosse verso p_f per arrivare a uno stato finale:

$$(p_0, w, X_0) \vdash (q_0, w, Z_0X_0) \overset{*}{\vdash} (p, \epsilon, X_0) \vdash (p_f, \epsilon, \epsilon)$$

Complessivamente, quella appena descritta

$$(p_0, w, X_0) \overset{*}{\vdash} (p_f, \epsilon, \epsilon)$$

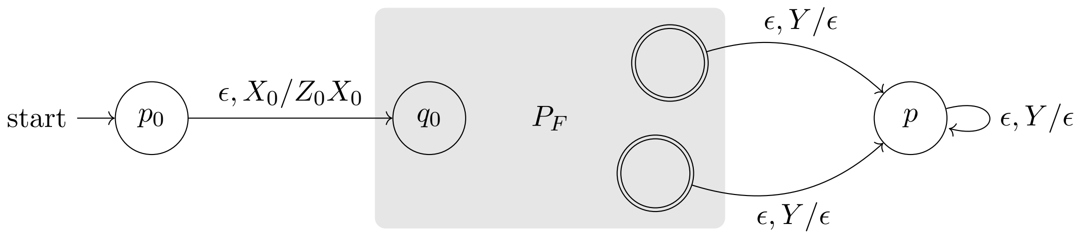
è una computazione dallo stato iniziale a uno stato finale di P_F che consuma tutto l'input, ovvero una computazione accettante.

Quella appena data è una dimostrazione informale di $w \in N(P_N) \implies w \in L(P_F)$. Per dimostrare formalmente il teorema, sarebbe necessario formalizzare questa dimostrazione, e mostrare che vale anche il viceversa, $w \in L(P_F) \implies w \in N(P_N)$.

4 Da stato finale a stack vuoto

Teorema: Dato un PDA $P_F = \langle Q, \Sigma, \Gamma, \delta_F, q_0, Z_0, F\rangle$ che accetta per stato finale il linguaggio $L(P_F)$, esiste un PDA P_N che accetta lo stesso linguaggio per stack vuoto, cioè tale che $N(P_N) = L(P_F)$.

Intuitivamente, l'idea della costruzione di P_N è la seguente:



- Si aggiunge un nuovo stato p, al quale si arriva tramite ϵ-mosse da tutti gli stati finali di P_F, in cui l'automa rimuove ripetutamente qualunque simbolo dallo stack, fino a svuotarlo. Così, P_N accetta ogni stringa che porta P_F in uno stato finale.
- Siccome L'automa P_F accetta per stato finale, esso potrebbe svuotare il proprio stack, eliminando il suo simbolo iniziale di stack Z_0, anche per stringhe che non accetta. In P_N, ciò porterebbe all'accettazione (indesiderata) di tali stringhe, dunque, per evitare che questo accada, si introduce un diverso simbolo iniziale di stack X_0 (che non può essere eliminato dalle transizioni di P_F), e si aggiunge Z_0 sopra di esso mediante un'ϵ-mossa da un nuovo stato iniziale p_0 a q_0,

  $$(p_0, w, X_0) \vdash (q_0, w, Z_0X_0)$$

  esattamente come nella costruzione vista prima per il verso opposto dell'equivalenza.

Formalmente, si definisce

$$P_N = \langle Q \cup \{p_0, p\}, \Sigma, \Gamma \cup \{X_0\}, \delta_N, p_0, X_0, F\rangle$$

dove δ_N è tale che:

$$\begin{aligned} & & \delta_N(p_0, \epsilon, X_0) &= \{(q_0, Z_0X_0)\} & (1)\\ \forall q \in Q,\ \forall a \in \Sigma_\epsilon,\ \forall Y \in \Gamma & & \delta_F(q, a, Y) &\subseteq \delta_N(q, a, Y) & (2)\\ \forall q \in F,\ \forall Y \in \Gamma \cup \{X_0\} & & (p, \epsilon) &\in \delta_N(q, \epsilon, Y) & (3)\\ \forall Y \in \Gamma \cup \{X_0\} & & \delta_N(p, \epsilon, Y) &= \{(p, \epsilon)\} & (4) \end{aligned}$$

Si noti in particolare che le coppie contenute in $\delta_N(q, \epsilon, Y)$ dipendono sia dal punto (2) che dal punto (3).

Per dimostrare il teorema, bisognerebbe verificare formalmente che, se esiste una computazione accettante in P_F,

$$(q_0, w, Z_0) \overset{*}{\vdash} (q, \epsilon, \alpha) \quad \text{con } q \in F$$

allora per le proprietà delle computazioni esiste una corrispondente computazione accettante in P_N,

$$(p_0, w, X_0) \vdash (q_0, w, Z_0X_0) \overset{*}{\vdash} (q, \epsilon, \alpha X_0) \overset{*}{\vdash} (p, \epsilon, \epsilon)$$

e viceversa.