Automi e Linguaggi Formali

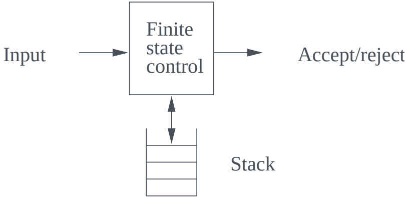
Automi a stack
(Pushdown automata)

A.A. 2014-2015
Enrico Mezzetti
emezzett@math.unipd.it

Università degli Studi di Padova

Automi a stack

- Un **Pushdown Automata** (PDA) o *Automa a stack*
- Essenzialmente un ϵ-NFA con uno stack
  - Per $\rightarrow$ memorizzare stringa di simboli
  - Stack e' **LIFO** (*last-in-first-out*)



- Per ogni transizione un PDA:
  1. Consuma un simbolo di input (eccetto ϵ)
  2. Esegue (eventualmente) una transizione
  3. Modifica il top dello stack
     - Elimina, sostituisce, inserisce

Esempio PDA

- Consideriamo il linguaggio $L_{ww^r} = \{ww^R : w \in \{0,1\}^*\}$
- CFG(ww^r) ha regole di produzione

$$P \rightarrow 0P0|1P1|\epsilon$$

- Un PDA per L_{wwr} ha tre stati
  1. Iniziamo da q_0, che rappresenta la lettura della prima parte dell'input w. Rimaniamo in q_0 e accumuliamo (push) i simboli in input sullo stack
  2. Andiamo spontaneamente in q_1 quando assumiamo di aver letto la prima parte dell'input. Siamo nel mezzo di ww^R.
  3. Confrontiamo il prossimo simblo in input (in w^R) e lo paragoniamo al simbolo al top dello stack. Se sono uguali esguiamo una pop e rimaniamo in q_1.
     Se non sono uguali, ci fermiamo (*stuck*).
  4. Quando lo stack e' vuoto, andiamo in q_2 e accettiamo la stringa

Definizione formale di PDA

- Un PDA e' una tupla di 7 elementi

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

dove

  - Q e' un insieme finito di stati
  - Σ e' un *alfabeto finito di input* $(a, b, c, \dots)$
  - Γ e' un *alfabeto finito di stack* $(\alpha, \beta, \gamma, \dots)$
  - $\delta : Q \times \Sigma \cup \{\epsilon\} \times \Gamma \to 2^{Q \times \Gamma^*}$ e' la *funzione di transizione,*
  - q_0 e' lo *stato iniziale,*
  - $Z_0 \in \Gamma$ e' il *simbolo iniziale* per la pila, e
  - $F \subseteq Q$ e' l'insieme di *stati di accettazione.*

- Funzione di transizione
  - $\delta(q, a, X) = \{(p_1, \gamma_1), \dots (p_k, \gamma_k)\}$

    nuovo stato ↑ ↑ *stringa che rimpiazza X*

  $\gamma = \epsilon,\ X$ oppure YZ

Esempio PDA

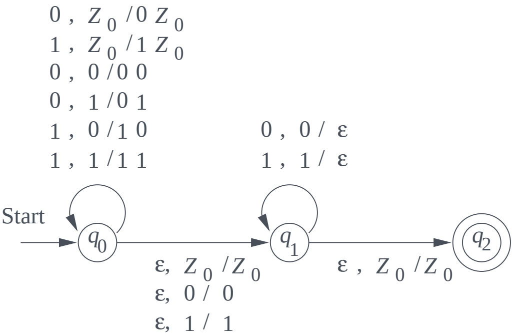
- Il PDA per L_{ww^r} formalmente definito come

$$P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, \delta, q_0, Z_0, \{q_2\})$$

	$0, Z_0$	$1, Z_0$	0,0	0,1	1,0	1,1	ϵ, Z_0	$\epsilon, 0$	$\epsilon, 1$
$\rightarrow q_0$	$q_0, 0Z_0$	$q_0, 1Z_0$	$q_0, 00$	$q_0, 01$	$q_0, 10$	$q_0, 11$	q_1, Z_0	$q_1, 0$	$q_1, 1$
q_1			q_1, ϵ			q_1, ϵ	q_2, Z_0		
$\star q_2$									

Notazione grafica di un PDA

- Il PDA per L_{ww^r} come *diagramma di transizione*
  - Archi corrispondono alle transizioni del PDA



  - $q \xrightarrow{a,X/\alpha} p \Rightarrow \delta(q, a, X)$ *contiene la coppia* (p, α)

Descrizioni istantanee

- **Descrizioni istantanee** (ID) di un PDA
- Formalismo per ragionare sul comportamento di un PDA

$$\underset{\textit{input residuo}\uparrow\quad\uparrow\textit{contenuto dello stack (sx}\rightarrow\textit{dx)}}{\overset{\textit{stato}\downarrow}{(q, w, \gamma)}}$$

- Sia $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ un PDA
  - Allora $\forall\ w \in \Sigma^*, \beta \in \Gamma^*$

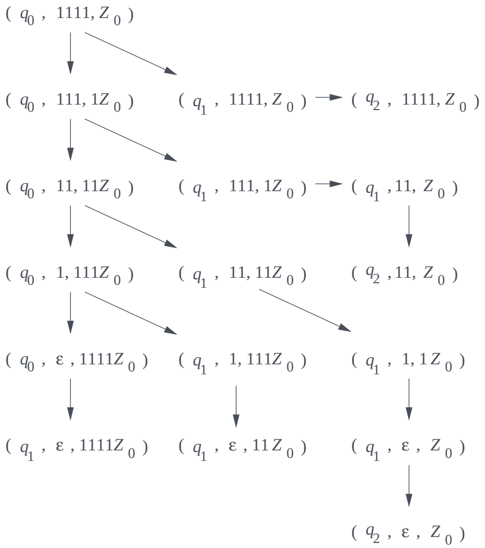
$$(p, \alpha) \in \delta(q, a, X) \Rightarrow (q, aw, X\beta) \vdash (p, w, \alpha\beta)$$

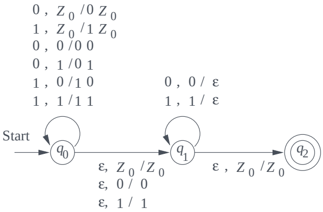
Consumando a dall'input e sostituendo X con α si puo' transitare da q a p

- Definiamo $\overset{*}{\vdash}$ la chiusura riflessiva e transitiva di $\vdash$

Esempio ID

- Il PDA per L_{ww^r} su input 1111





ID e transizioni

- Principio fondamentale per lo studio dei PDA
  - I dati che P non esamina <u>non ne influenzano</u> la computazione

    (a) Se una sequenza di ID (*computazione*) e' legale per un PDA, allora lo e' anche la sequenza ottenuta aggiungendo una stringa alla fine della seconda componente (input).

    (b) Se una computazione e' legale per un PDA, allora lo e' anche la computazione formata aggiungendo una stringa alla fine della terza componente (fondo dello stack)

    (c) Se una sequenza di ID e' una computazione legale per un PDA, e la coda di un input non e' vuota, allora rimuovendola da tutte le ID si ottiene una computazione lecita.

Th. 6.5 Se $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ e' un PDA, allora (a),(b)
$\forall\ w \in \Sigma^*, \beta \in \Gamma^*$

$$(q, x, \alpha) \vdash^* (p, y, \beta) \Rightarrow (q, xw, \alpha\gamma) \vdash^* (p, yw, \beta\gamma)$$

Prova: Per induzione sulla sequenza di computazione di destra che utilizza in effetti tutte le transizioni della sequenza di sinistra.

- (a) $\rightarrow\ \gamma = \epsilon$
  (b) $\rightarrow w = \epsilon$
- L'inverso del teorema e' falso

Th. 6.6 Se $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ e' un PDA, allora (c)

$$(q, xw, \alpha) \vdash^* (p, yw, \beta) \Rightarrow (q, x, \alpha) \vdash^* (p, y, \beta)$$

Accettazione di un linguaggio

- Esistono due modalita' di accettazione di un PDA
  - Accettazione per **stato finale**
  - Accettazione per **stack vuoto**

- Modalita' sono equivalenti
  - $\exists\ PDA_{StatoFinale}(L) \iff \exists PDA_{StackVuoto}(L)$

- Dato un $PDA\ P$: $\quad L_{StatoFinale}(P) \neq L_{StackVuoto}(P)$
  - Si puo' passare da uno all'altro

Accettazione per stato finale

- Sia $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ un PDA
- Il linguaggio accettato da P **per stato finale** e'

$$L(P) = \{w : (q_0, w, Z_0) \overset{*}{\vdash} (q, \epsilon, \alpha),\ q \in F\}$$

A partire da un'istantanea iniziale su input w,
P consuma w ed entra in uno stato accettante
Lo stack residuo (α) e' irrilevante

Esempio accettazione per stato finale

- PDA delle palindrome pari $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$
- Accetta esattamente $L_{ww^r} = \{ww^R : w \in \{0,1\}^*\}$
  - P accetta $w \iff w \in L_{ww^r}$

Se Sia $x \in L_{ww^r}$ allora $x = ww^r$ ed esiste una computazione legale in P

$$(q_0, ww^R, Z_0) \overset{*}{\vdash} (q_0, w^R, w^R Z_0) \vdash (q_1, w^R, w^R Z_0) \overset{*}{\vdash}$$
$$\overset{*}{\vdash} (q_1, \epsilon, Z_0) \vdash (q_2, \epsilon, Z_0)$$

Solo se P accetta w solo attraverso q_1 con lo stack vuoto (Z_0)
E' sufficiente mostrare che se $(q_0, x, Z_0) \overset{*}{\vdash} (q_1, \epsilon, Z_0)$ allora $x = ww^R$, per una qualche stringa w.
Dimostriamo per induzione su $|x|$ che

$$(q_0, x, \alpha) \overset{*}{\vdash} (q_1, \epsilon, \alpha) \quad \Rightarrow \quad x = ww^R.$$

Esempio accettazione per stato finale

Base: Se $x = \epsilon$ allora x e' nella forma ww^r con $w = \epsilon$

Induz: Supponiamo che $x = a_1 a_2 \cdots a_n$, dove $n > 0$,
Ip. induttiva vale per stringhe piu' corte
Il PDA da ID (q_0, x, α) puo' eseguire due transizioni

Mossa 1 La mossa spontanea $(q_0, x, \alpha) \vdash (q_1, x, \alpha)$
Allora $(q_1, x, \alpha) \overset{*}{\vdash} (q_1, \epsilon, \beta)$ implica che $|\beta| < |\alpha|$, che implica $\beta \neq \alpha$ (premessa falsa)

Mossa 2 $(q_0, a_1 a_2 \cdots a_n, \alpha) \vdash (q_0, a_2 \cdots a_n, a_1\alpha)$
Sequenza puo' terminare in (q_1, ϵ, α) solo se l'ultimo step e' un'eliminazione: $(q_1, a_n, a_1\alpha) \vdash (q_1, \epsilon, \alpha) \Rightarrow a_1 = a_n$

Sappiamo anche che $(q_0, a_2 \cdots a_n, a_1\alpha) \overset{*}{\vdash} (q_1, a_n, a_1\alpha)$
Per il teorema 6.6 possiamo rimuovere a_n (ancora legale) ottenendo $(q_0, a_2 \cdots a_{n-1}, a_1\alpha) \overset{*}{\vdash} (q_1, \epsilon, a_1\alpha)$

Per ipotesi induttiva $a_2 \cdots a_{n-1} = yy^R$. Poich $a_1 = a_n$, $x = a_1 yy^R a_n$ e' una palindrome.

Accettazione per stack vuoto

- Sia $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ un PDA
- Il linguaggio accettato da P **per stack vuoto** e'

$$N(P) = \{w : (q_0, w, Z_0) \overset{*}{\vdash} (q, \epsilon, \epsilon)\}$$

N(P) e' l'insieme degli input w che P puo' consumare svuotando nel contempo lo stack.
Lo stato raggiunto (q) e' irrilevante

Esempi

✎

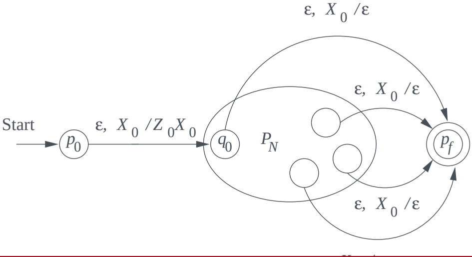
Da stack vuoto a stato finale

- La classe dei linguaggi accettati per stato finale $\mathcal{L}(P)$ coincide con la classe di linguaggi accettati per stack vuoto $\mathcal{N}(P)$

Th 6.9: Se $L = N(P_N)$ per un PDA $P_N = (Q, \Sigma, \Gamma, \delta_N, q_0, Z_0)$ allora $\exists$ PDA P_F, tale che $L = L(P_F)$.

Prova: Usiamo $X_0 \notin \Gamma$ come *simbolo iniziale e sentinella* per P_F
Sia $P_F = (Q \cup \{p_0, p_f\}, \Sigma, \Gamma \cup \{X_0\}, \delta_F, p_0, X_0, \{p_f\})$ dove

- $\delta_F(p_0, \epsilon, X_0) = \{(q_0, Z_0X_0)\}$
- $q \in Q, a \in \Sigma \cup \{\epsilon\}, Y \in \Gamma : \delta_F(q, a, Y) = \delta_N(q, a, Y)$
- $(p_f, \epsilon) \in \delta_F(q, \epsilon, X_0)$



Da stack vuoto a stato finale - 2

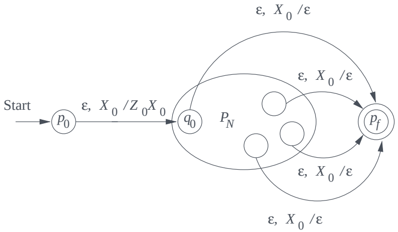
- Dobbiamo dimostrare che $w \in L(P_F) \iff w \in N(P_N)$

Se Sia $w \in N(P_N)$, allora $(q_0, w, Z_0) \overset{*}{\underset{N}{\vdash}} (q, \epsilon, \epsilon)$ per un qualche q

- Per Th. 6.5 $\rightarrow (q_0, w, Z_0X_0) \overset{*}{\underset{N}{\vdash}} (q, \epsilon, X_0)$
- Dato che $\delta_N \subset \delta_F \rightarrow (q_0, w, Z_0X_0) \overset{*}{\underset{F}{\vdash}} (q, \epsilon, X_0)$
- Quindi $(p_0, w, X_0) \underset{F}{\vdash} (q_0, w, Z_0X_0) \overset{*}{\underset{F}{\vdash}} (q, \epsilon, X_0) \underset{F}{\vdash} (p_f, \epsilon, \epsilon)$
- P_F accetta w per stato finale

Solo se Sia $w \in L(P_F)$ allora $w \in L(P_N)$

- Per le regole di costruzione del diagramm

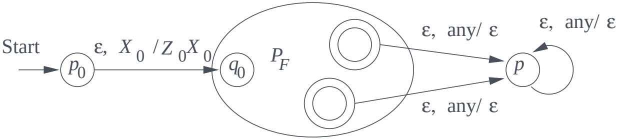


Da stato finale a stack vuoto

Th. 6.11 Sia $L = L(P_F)$, per un PDA $P_F = (Q, \Sigma, \Gamma, \delta_F, q_0, Z_0, F)$ allora $\exists$ PDA P_n, tale che $L = N(P_N)$.

Prova: Usiamo $X_0 \notin \Gamma$ come *simbolo iniziale e sentinella* per P_N
Sia $P_N = (Q \cup \{p_0, p\}, \Sigma, \Gamma \cup \{X_0\}, \delta_N, p_0, X_0)$ dove

- $\delta_N(p_0, \epsilon, X_0) = \{(q_0, Z_0 X_0)\}$
- $\forall\ q\ \in\ Q, a \in \Sigma \cup \{\epsilon\}, Y \in \Gamma : \delta_N(q, a, Y) = \delta_F(q, a, Y)$
- $\forall q \in F$, and $Y \in \Gamma \cup \{X_0\} : (p, \epsilon) \in \delta_N(q, \epsilon, Y)$
- $\forall\ Y \in \Gamma \cup \{X_0\},\ \delta_N(p, \epsilon, Y) = \{(p, \epsilon)\}$,



Da stato finale a stack vuoto - 2

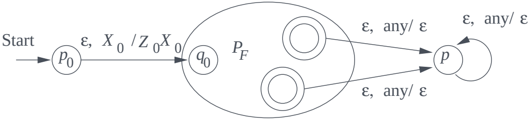
- Dobbiamo dimostrare che $w \in N(P_N) \iff w \in L(P_F)$

Se Sia $w \in L(P_F)$, allora $(q_0, w, Z_0) \overset{*}{\underset{F}{\vdash}} (q, \epsilon, \alpha)$ per qualche q, α

- Per Th. 6.5 e $\delta_F \subseteq \delta_N \rightarrow (q_0, w, Z_0X_0) \overset{*}{\underset{N}{\vdash}} (q, \epsilon, \alpha X_0)$
- Quindi $(p_0, w, X_0) \underset{N}{\vdash} (q_0, w, Z_0X_0) \overset{*}{\underset{N}{\vdash}} (q, \epsilon, \alpha X_0) \overset{*}{\underset{N}{\vdash}} (p, \epsilon, \epsilon)$
- P_N accetta w per stack vuoto

Solo se Sia $w \in L(P_N)$ allora $w \in L(P_F)$

- Per le regole di costruzione del diagramm



Equivalenza di PDA e CFG

- Per linguaggio L valgono le seguenti caratterizzazioni

generato da una CFG

$\Longleftrightarrow$

accettato da un PDA per pila vuota

$\Longleftrightarrow$

accettato da un PDA per stato finale

- Vedremo solo passaggio da grammatica ad automa a stack

Da CFG a PDA

- Data CFG G costruiamo un PDA che simuli $\overset{*}{\underset{lm}{\Rightarrow}}$
- Ogni forma sentenziale sinistra e' nella forma

terminali ↓ ↓ *terminali e variabili* (**coda**)

$$xA\alpha$$

leftmost variable↑

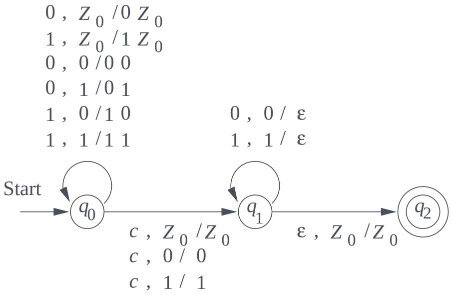
- In un PDA corrisponde a
  - Avere "consumato" x dall'input (resta y se $w = xy$)
  - A compare in cima allo stack
  - ID e' $(q, y, A\alpha)$
- Ad una derivazione $xA\alpha \underset{lm}{\Rightarrow} x\beta\alpha$ corrisponde
  - $(q, y, A\alpha) \vdash (q, y, \beta\alpha)$ (leggendo ε)

Da CFG a PDA - 2

- L'ID $(q, y, \beta\alpha)$ potrebbe non rappresentare la successiva forma sentenziale sinistra
  - β puo' avere un prefisso di terminali
  - β puo' essere composta solo da terminali
  - anche α potrebbe consistere di terminali
- Dobbiamo eliminare eventuali terminali nel prefisso di $\beta\alpha$
  - PDA li elimina se li legge nell'input
  - A seconda delle produzioni scelte potremmo finire in una diramazione morta
- Al termine il PDA consuma tutto l'input e resta con lo stack vuoto
- Formalmente, per una CFG $G = (V, T, P, S)$
  - $P_G = (\{q\}, T, V \cup T, \delta, q, S)$
  - $\delta(q, \epsilon, A) = \{(q, \beta) : A \rightarrow \beta \in P\}$ per $A \in V$
  - $\delta(q, a, a) = \{(q, \epsilon)\}$ per $a \in T$

PDA deterministici

- PDA generalmente non-deterministici
- PDA deterministici sono sotto-classe (*parser*)
- PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ e' *deterministico* se e solo se
  1. $\delta(q, a, X)$ e' sempre o vuoto o con un solo elemento.
  2. Se $\delta(q, a, X)$ non e' vuoto, allora $\delta(q, \epsilon, X)$ deve essere vuoto.
- Esempio $L_{wcw^R} = \{wcw^R : w \in \{0, 1\}^*\}$



Linguaggi regolari e PDA deterministici

- Relazione tra classi di linguaggi

$$LR \subset L(DPDA) \subset CFL$$

Th. 6.17: Se L e' regolare, allora $L = L(P)$ per qualche DPDA P.

Prova: Dato che L e' regolare, esiste un DFA A tale che $L = L(A)$
Sia $A = (Q, \Sigma, \delta_A, q_0, F)$ allora definiamo il DPDA

$$P = (Q, \Sigma, \{Z_0\}, \delta_P, q_0, Z_0, F)$$

dove

$$\delta_P(q, a, Z_0) = \{(\delta_A(q, a), Z_0)\},$$

per tutti i $p, q \in Q$ e $a \in \Sigma$.

- Ignoriamo semplicemente lo stack
  - Per induzione su $|w|$ possiamo oseervare che

$$(q_0, w, Z_0) \overset{*}{\vdash} (p, \epsilon, Z_0) \Leftrightarrow \hat{\delta}_A(q_0, w) = p$$

Linguaggi regolari e PDA deterministici - 2

- DPDA che accettano *per stack vuoto* sono meno espressivi
- Possono riconoscere solo CFL con la *proprieta' del prefisso*
  - L ha la *proprieta' del prefisso* se non esistono due stringhe distinte in L, tali che una e' un prefisso dell'altra
  - E.g., L_{wcwr} ha la proprieta' del prefisso.
  - E.g., $\{0\}^*$ non ha la proprieta' del prefisso.

Th. 6.19: L e' $N(P)$ per qualche DPDA P se e solo se L ha la proprieta' del prefisso e L e' $L(P')$ per qualche DPDA P'.

Linguaggi regolari e PDA deterministici - 3

- Quindi abbiamo
  - $LR \subseteq L(\mathrm{DPDA})$
  - $L_{wcw^r} \in L(\mathrm{DPDA}) \setminus LR$
- Esistono linguaggi $\in CFL \setminus L(\mathrm{DPDA})$
  - E.g., L_{ww^r}
- Relazione tra DPDA e grammatiche ambigue?
  - L_{ww^r} ha una grammatica non ambigua

    $$S \to 0S0|1S1|\epsilon$$

    ma non e' $L(\mathrm{DPDA})$
  - Vale pero' il contrario

Th. 6.20 Se $L = N(P)$ per qualche DPDA P, allora L ha una CFG non ambigua.

Th. 6.21 Se $L = L(P)$ per qualche DPDA P, allora L ha una CFG non ambigua.