

# Linguaggi Formali e Traduttori

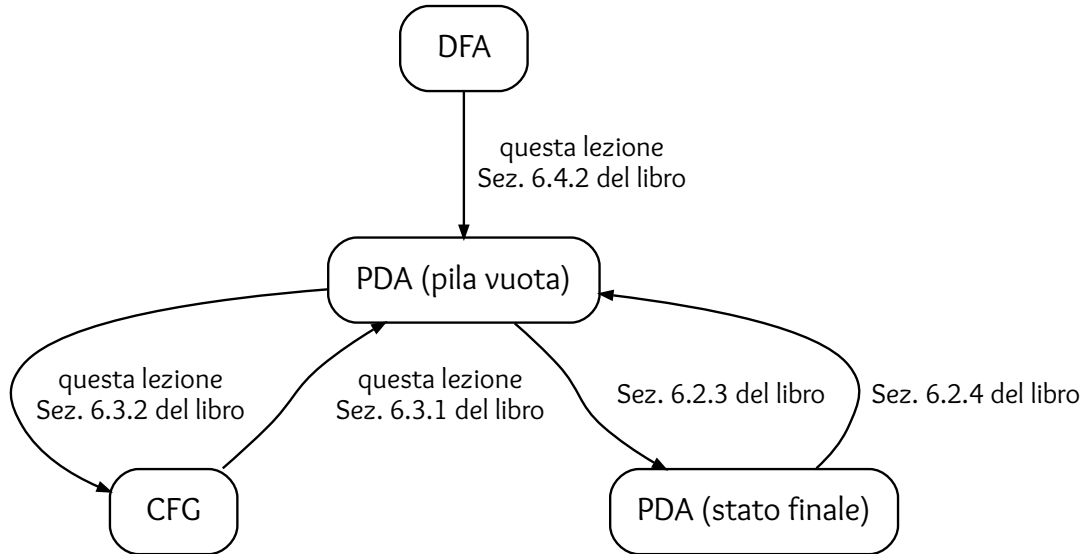
## 3.5 Relazione tra automi a pila e grammatiche libere

- Sommario
- Relazione tra CFG e PDA
- Osservazione
- Automi a pila deterministici
- Esempio: riconoscitore di stringhe  $wc w^R$
- DPDA e linguaggi regolari
- DPDA e grammatiche ambigue

È proibito condividere e divulgare in qualsiasi forma i materiali didattici caricati sulla piattaforma e le lezioni svolte in videoconferenza: ogni azione che viola questa norma sarà denunciata agli organi di Ateneo e perseguita a termini di legge.

# Sommario

- Studiamo la relazione tra CFG e PDA.
- Definiamo una variante **deterministica** dei PDA.
- Mostriamo che i PDA deterministici sono in grado di riconoscere tutti i linguaggi regolari e un sottoinsieme dei linguaggi liberi non inerentemente ambigui.



# Relazione tra CFG e PDA

## Teorema

1. Per ogni CFG  $G$ , esiste un PDA  $P$  tale che  $N(P) = L(G)$ .
2. Per ogni PDA  $P$ , esiste una CFG  $G$  tale che  $L(G) = N(P)$ .

## Intuizione per 1

Data  $G = (V, T, Q, S)$ , definiamo un PDA che simuli le derivazioni a sinistra di  $G$ . Sia  $P$  il PDA  $(\{q\}, T, V \cup T, \delta, q, S, \emptyset)$  dove  $\delta$  è definita come segue:

$$\begin{aligned}\delta(q, \epsilon, A) &= \{(q, \beta) \mid A \rightarrow \beta \in Q\} && \text{per ogni } A \in V \\ \delta(q, a, a) &= \{(q, \epsilon)\} && \text{per ogni } a \in T\end{aligned}$$

Per concludere la dimostrazione è sufficiente mostrare che

$$\alpha \Rightarrow_{lm}^* w \Leftrightarrow (q, w, \alpha) \vdash^* (q, \epsilon, \epsilon)$$

in quanto, come caso particolare, avremo

$$\underline{G \text{ genera } w} \Leftrightarrow \underline{S \Rightarrow^* w} \Leftrightarrow \underline{(q, w, S) \vdash^* (q, \epsilon, \epsilon)} \Leftrightarrow \underline{P \text{ accetta } w \text{ per pila vuota}}$$

I dettagli della dimostrazione si trovano nel libro di testo.

# Osservazione

Richiamando la funzione di transizione definita nella slide precedente:

$$\begin{aligned}\delta(q, \varepsilon, A) &= \{(q, \beta) \mid A \rightarrow \beta \in Q\} && \text{per ogni } A \in V \\ \delta(q, a, a) &= \{(q, \varepsilon)\} && \text{per ogni } a \in T\end{aligned}$$

- Il PDA che riconosce il linguaggio generato da una CFG fa uso chiave del **non determinismo** per “indovinare” la produzione giusta da usare per riscrivere una variabile  $A$
- I risultati di equivalenza tra PDA e CFG hanno una valenza principalmente teorica, ma **non** aiutano molto a ottenere **riconoscitori efficienti per linguaggi liberi**
- Consideriamo PDA **deterministici**

# Automi a pila deterministici

## Intuizione

Non devono esserci “scelte” possibili a partire dalla stessa D.I.

## Definizione

Un **automa a pila deterministico** (DPDA, da **D**eterministic **P**ush**D**own **A**utomaton) è un PDA  $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$  in cui, per ogni  $q \in Q$ ,  $a \in \Sigma$  e  $X \in \Gamma$ , l'insieme  $\delta(q, a, X) \cup \delta(q, \epsilon, X)$  contiene al massimo un elemento.

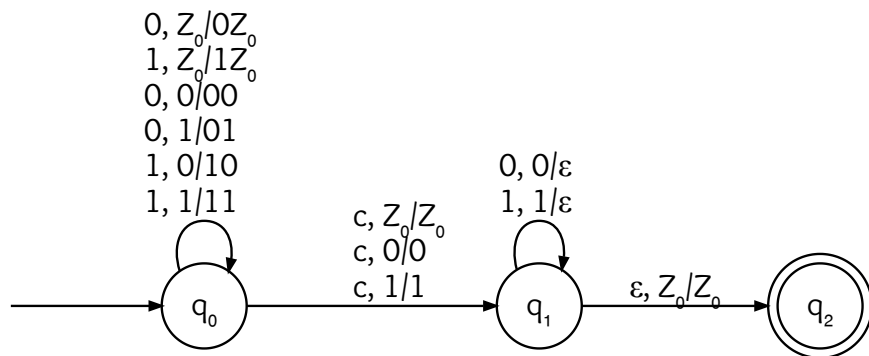
## Note

- A parità di stato, simbolo letto e simbolo sulla pila, un DPDA può fare al massimo una mossa.
- L'insieme  $\delta(q, a, X) \cup \delta(q, \epsilon, X)$  può essere vuoto, nel qual caso il DPDA rifiuta definitivamente la stringa, senza che vi siano altre possibilità di riconoscerla.

# Esempio: riconoscitore di stringhe $wcw^R$

- È possibile dimostrare che non esiste un DPDA in grado di riconoscere il linguaggio delle stringhe della forma  $ww^R$ . Intuizione: il PDA deve “scommettere” sul punto in cui finisce il prefisso  $w$  e inizia il suffisso  $w^R$ .
- Separando con una “sentinella”  $c$  il prefisso  $w$  dal suffisso  $w^R$ , il linguaggio delle stringhe della forma  $wcw^R$  diventa riconoscibile dal DPDA seguente

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



- Notare che il comportamento è deterministico anche nello stato  $q_1$ , in quanto la transizione  $\epsilon$  è possibile solo se in cima alla pila c'è  $Z_0$ .

# DPDA e linguaggi regolari

## Teorema

Ogni linguaggio regolare è riconosciuto da un DPDA.

## Dimostrazione

Sia  $A = (Q, \Sigma, \delta_A, q_0, F)$  un DFA che riconosce un certo linguaggio regolare  $L$ . Definiamo un PDA strutturalmente identico ad  $A$  che non usa la pila. Sia

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

dove

$$\begin{aligned}\delta_P(q, a, Z_0) &= \{(\delta_A(q, a), Z_0)\} && \text{per ogni } q \in Q \text{ e } a \in \Sigma \\ \delta_P(q, \epsilon, Z_0) &= \emptyset && \text{per ogni } q \in Q\end{aligned}$$

È facile mostrare che  $(q_0, w, Z_0) \vdash^* (\hat{\delta}_A(q_0, w), \epsilon, Z_0)$  da cui segue il risultato.

# DPDA e grammatiche ambigue

## Teorema

Per ogni DPDA  $P$ , esiste una grammatica libera non ambigua  $G$  tale che  $L(G) = N(P)$ .

## Osservazione

Il viceversa del risultato qui sopra non vale. In particolare, esistono grammatiche non ambigue il cui linguaggio non è riconosciuto da alcun DPDA. Ad esempio

$$S \rightarrow \epsilon \mid 0S0 \mid 1S1$$

è non ambigua e genera il linguaggio  $\{ww^R \mid w \in \{0,1\}^*\}$  che non è riconoscibile da un DPDA.

## Conclusioni

La famiglia dei linguaggi riconoscibili da DPDA è inclusa in – ma non coincide con – quella dei linguaggi generabili da grammatiche libere non ambigue.

## Potere espressivo dei formalismi visti fino ad ora

- $DFA = NFA = \epsilon\text{-NFA} = RE$
- $DFA \subsetneq DPDA \subsetneq CFG \text{ non ambigua} \subsetneq CFG = PDA$
- Non abbiamo ancora un algoritmo per ottenere il DPDA quando esiste