Corso di Linguaggi di Programmazione — Paradigmi di Programmazione
Prova scritta del 6 settembre 2023.

Tempo a disposizione: ore 2.
Svolgere gli esercizi 1-4, 5-6 e 7-8 su tre fogli separati.

1. Descrivere le regole di semantica operazionale strutturata per l'espressione booleana b_0 `xor` b_1, secondo la disciplina di valutazione interna-destra (ID). Ricordo che `xor` è l'operatore di or-esclusivo: l'espressione b_0 `xor` b_1 vale `tt` se e soltanto se il valore di verità di b_0 e di b_1 sono opposti. È possibile definire regole secondo la disciplina esterna-sinistra (ES)?

2. Sia data la grammatica G con simbolo iniziale S

$$\begin{array}{rcl} S & \rightarrow & AS \mid \mathrm{aa} \\ A & \rightarrow & \mathrm{a} \end{array}$$

   (*i*) Quale linguaggio genera G? (*ii*) È la grammatica G regolare? (*iii*) Determinare una grammatica G', equivalente a G, in forma normale di Greibach.

3. Mostrare che $L_1 = \{a^n b^n a^m \mid n, m \geq 1\}$ è libero deterministico, costruendo un opportuno DPDA. Sapendo che anche $L_2 = \{a^n b^m a^n \mid n, m \geq 1\}$ è libero deterministico, è vero che $L_1 \cap L_2$ è un linguaggio libero deterministico?

4. Si consideri la seguente grammatica G con simbolo iniziale S:

$$\begin{array}{rcl} S & \rightarrow & AB \mid \epsilon \\ A & \rightarrow & \epsilon \mid \mathrm{a}A\mathrm{b} \\ B & \rightarrow & \epsilon \mid \mathrm{b}B\mathrm{a} \end{array}$$

   (*i*) Si verifichi che G è ambigua. (*ii*) Si modifichi al minimo G in modo da ottenere una grammatica G' equivalente non ambigua. (*iii*) Si discuta se la grammatica risultante G' sia di classe LL(1).

1)

$$\frac{\langle b_1, \sigma \rangle \rightarrow \langle b_1', \sigma' \rangle}{\langle b_0 \text{ xor } b_1, \sigma \rangle \rightarrow \langle b_0 \text{ xor } b_1', \sigma' \rangle} \quad (ID)$$

$$\frac{\langle b_0, \sigma \rangle \rightarrow \langle b_0', \sigma' \rangle}{\langle b_0 \text{ xor } t_1, \sigma \rangle \rightarrow \langle b_0' \text{ xor } t_1, \sigma' \rangle}$$

$$\frac{}{\langle t_0 \text{ xor } t_1, \sigma \rangle \rightarrow \langle t, \sigma \rangle}$$

t_0	t_1	$t_0 \text{ xor } t_1 = t$
tt	tt	ff
tt	ff	tt
ff	tt	tt
ff	ff	ff

$$(ES) \quad \frac{\langle b_0, \sigma \rangle \rightarrow \langle b_0', \sigma' \rangle}{\langle b_0 \text{ xor } b_1, \sigma \rangle \rightarrow \langle b_0' \text{ xor } b_1, \sigma' \rangle}$$

$$\frac{}{\langle tt \text{ xor } b_1, \sigma \rangle \rightarrow \langle \sim b_1, \sigma \rangle}$$

$$\frac{}{\langle ff \text{ xor } b_1, \sigma \rangle \rightarrow \langle b_1, \sigma \rangle}$$

2)

$$\left.\begin{array}{l} S \to AS \mid aa \\ A \to a \end{array}\right] G$$

- $L(G) = \{ a^{n+2} \mid n \geq 0 \}$ $\quad a^* aa$

- G non è una grammatica regolare perché

$$S \to AS \quad e \quad S \to aa$$

non sono della forma corretta

- Nella forma normale di Greibach, le produzioni possono essere della forma

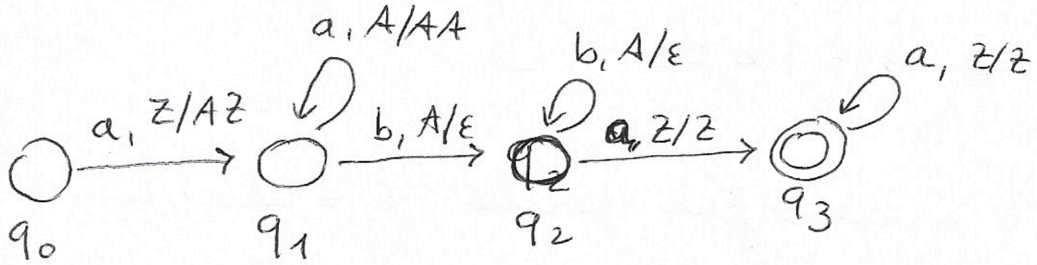
$$A \to aBC$$
$$A \to aB$$
$$A \to a$$

Una grammatica G', equivalente a G, potrebbe essere

$$S \to aS \mid aA$$
$$A \to a$$

che, oltre ad essere in forma normale di Greibach, è pure regolare.

3) $L_1 = \{ a^n b^n a^m \mid n, m \geq 1 \}$



DPDA che riconosce L_1 per stato finale (q_3).

$L_2 = \{ a^n b^m a^n \mid n, m \geq 1 \}$ è lib. det.

$L_1 \cap L_2 = \{ a^n b^n a^n \mid n \geq 1 \}$ che non è libero!

Dimostriamolo usando il pumping theorem a rovescio

- Fissiamo $N > 0$ generico $\quad (\forall N > 0)$
- Scegliamo $z = a^N b^N a^N \quad (\exists z \in L.\ |z| \geq N)$
- Per ogni $uvwxy$ tali che $\cdot\ z = uvwxy$
  $\cdot\ |vwx| \leq N$
  $\cdot\ |vx| \geq 1$

deve essere che vwx appartiene a

- a^*, oppure
- $a^* b^*$, oppure
- b^*, oppure
- $b^* a^*$, oppure
- a^*

In ogni caso, o contiene solo "b", oppure non contiene una "a" del primo blocco e una "a" del terzo blocco, per cui

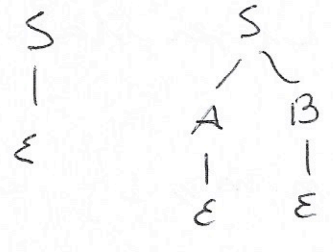
$\exists k = 2.\ uv^2wx^2y \notin L$

$\Rightarrow L$ non libero

4)

$$\left.\begin{array}{l} S \to AB \mid \varepsilon \\ A \to \varepsilon \mid aAb \\ B \to \varepsilon \mid bBa \end{array}\right] G$$

- G è ambigua perché $\varepsilon \in L(G)$ ammette 2 alberi di derivazione



- G' non ambigua, equivalente a G, si ottiene semplicemente eliminando la produzione $S \to \varepsilon$:

$$\left.\begin{array}{l} S \to AB \\ A \to \varepsilon \mid aAb \\ B \to \varepsilon \mid bBa \end{array}\right] G'$$

-

	Follow
S	\$
A	b, \$
B	a, \$

G' è LL(1) perché

- $First(\varepsilon) \cap First(aAb) = \emptyset$
  $Follow(A) \cap First(aAb) = \emptyset$
- $First(\varepsilon) \cap First(bBa) = \emptyset$
  $Follow(B) \cap First(bBa) = \emptyset$