CORSO DI LINGUAGGI DI PROGRAMMAZIONE
PROVA SCRITTA DEL 19 GIUGNO 2020.

Tempo a disposizione: ore 2.
Svolgere gli esercizi 1-4 e 5-8 su due fogli differenti.

1. Si consideri la grammatica $G = (\{S\}, \{+, a\}, S, \{S \to S{+}S, S \to a\})$. La grammatica G è ambigua? È possibile costruire un parser $LL(1)$ per G? E un parser $SLR(1)$ per G?

2. Fornire una definizione regolare per la categoria sintattica *IDE* (identificatori), che soddisfi questi requisiti: un identificatore è una qualunque sequenza di simboli alfanumerici (includendo anche il simbolo $*$) che cominci con una cifra, contenga al suo interno una sola occorrenza del simbolo $*$ e termini con la lettera z. Costruire un NFA per riconoscere gli identificatori così definiti.

3. Si consideri la grammatica G con simbolo iniziale S:

$$\begin{array}{rcl} S & \to & SA \mid \epsilon \mid cDB \\ A & \to & \epsilon \mid a \\ B & \to & b \mid A \\ C & \to & aSB \mid c \\ D & \to & cD \end{array}$$

   Semplificare la grammatica G, eseguendo, nell'ordine specificato, le seguenti trasformazioni: (i) rimuovere la ricorsione sinistra; (ii) rimuovere i simboli inutili; (iii) eliminare le produzioni epsilon; (iv) eliminare le produzioni unitarie. (v) La grammatica risultante è del tutto equivalente a G?

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

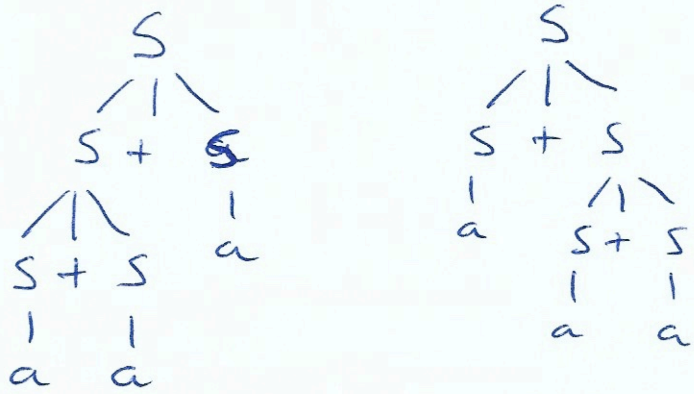
$$\begin{array}{rcl} S & \to & \mathtt{b}A\mathtt{d} \\ A & \to & \mathtt{a} \mid \mathtt{a}B\mathtt{a} \\ B & \to & \epsilon \end{array}$$

   (i) Quale linguaggio genera G? (ii) Costruire l'automa canonico LR(1) per G. (iii) Esiste una grammatica G' di classe LL(1) che genera il linguaggio $L(G)$?

1) $S \rightarrow S+S \mid a \;] \; G$

$a+a+a$

- G è ambigua



- Non è possibile costruire un parser LL(1) per G perché le grammatiche LL(1) sono non ambigue

- Per lo stesso motivo, non è possibile costruire un parser SLR(1) per G.

---

2)

IDE := cifra tutto* * tutto* z

cifra := [0 - 9]

letter := [a-z A - Z]

tutto := cifra | letter | *

3) $S \to SA \mid \varepsilon \mid cDB$

$A \to \varepsilon \mid a$

$B \to b \mid A$

$C \to aSB \mid c$

$D \to cD$

G

eliminare ric. sx $S \to SA$

$S \to S' \mid \cancel{cDBS'}$

$S' \to AS' \mid \varepsilon$

$A \to \varepsilon \mid a$

$\cancel{B \to b \mid A}$

$\cancel{C \to aSB \mid c}$

$\cancel{D \to cD}$

rimuovere i simboli inutili
- D non generatore
- B e C non raggiungibili

$S \to S'$

$S' \to AS' \mid \varepsilon$

$A \to \varepsilon \mid a$

eliminare le ε-prod

$S \to S'$

$S' \to AS' \mid A \mid S'$ inutile

$A \to a$

$N(G) = \{A, S', S\}$

eliminare le prod. unitarie

$S \to AS' \mid a$

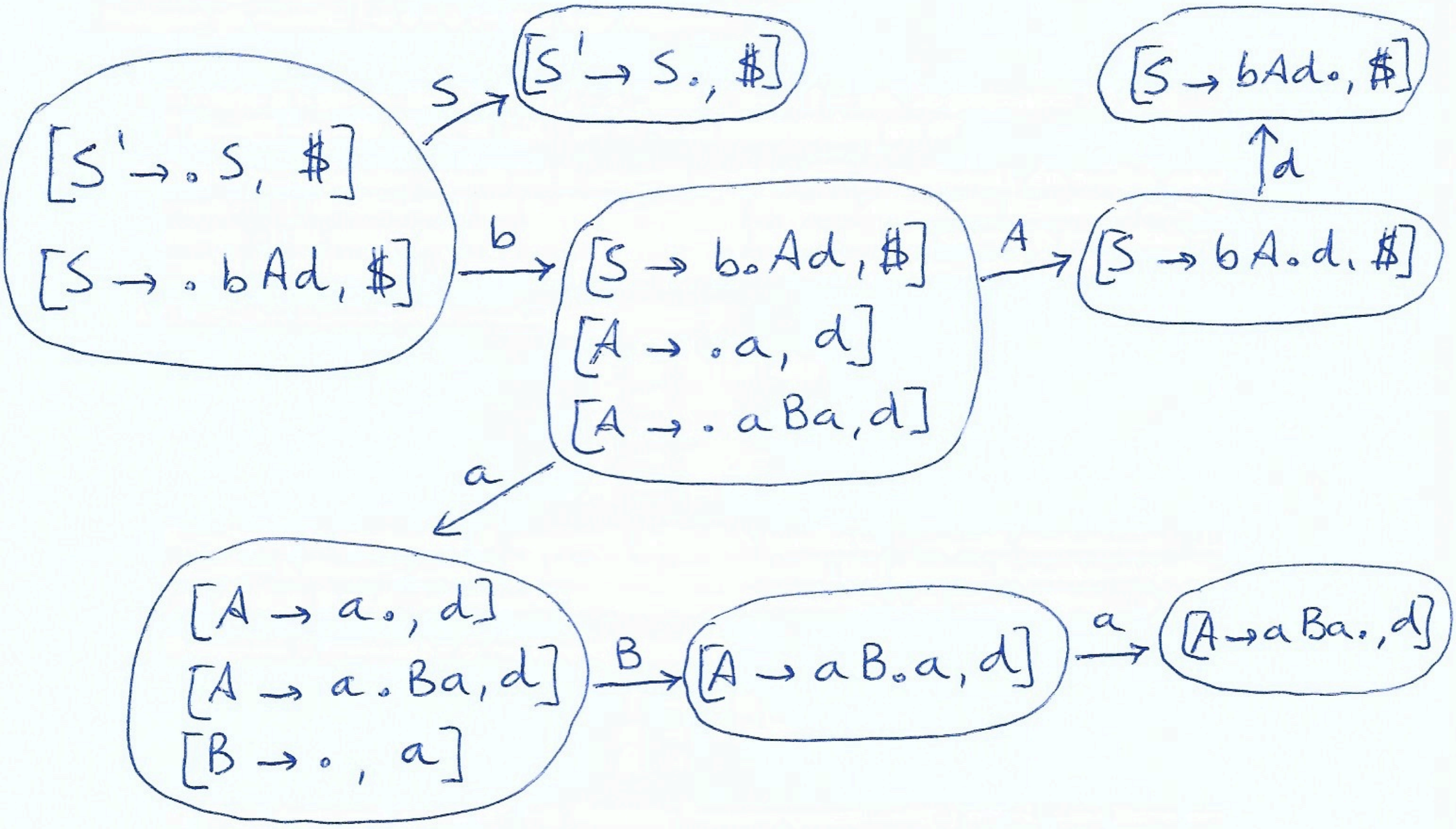
$S' \to AS' \mid a$

$A \to a$

La grammatica risultante non è del tutto equivalente a G, perché non genera ε.

4)

$$S \to bAd \qquad L(S) = \{bad, baad\}$$
$$A \to a \mid aBa \qquad L(A) = \{a, aa\}$$
$$B \to \varepsilon \qquad L(B) = \{\varepsilon\}$$



Poiché $L(G) = \{bad, baad\}$ è finito, è sicuramente regolare e, per un teorema visto a lezione, tutti i ling. regolari sono di classe LL(1).