

SOLUZIONI APPELLO 2019_PROTO

1.

NO.

2.

VERO.

3. (COSTRUZIONE IDENTICA AD ES. 3 DEL 2021-01_P)
1 STATO, 1 FINALE.

4. (COSTRUZIONE IDENTICA AD ES. 4 DEL 2021-01_P)
 $Q_{ab} = \{A, B, C, E\}$.

5. (COSTRUZIONE IDENTICA AD ES. 5 DEL 2021-01_P)
 $D_1 = D_m$.
 $P_{abab} = \{B\}$.

6. (COSTRUZIONE IDENTICA AD ES. 6 DEL 2021-01_P)

	a	b	c	\$
B	$B \rightarrow \epsilon$		$B \rightarrow \epsilon$	$B \rightarrow \epsilon$

7.

NO CONFLICT.

8.

$$\boxed{8.} \quad \gamma[Aa] = \left\{ [S \rightarrow Aa \cdot B, \{ \$ \}], [B \rightarrow \cdot, \{ \$ \}] \right\}$$

9.

Gli automi LR(1) ed LRm(1) COINCIDONO.
NO CONFLICT.

10.

Derivazione RIGT-MOST di w:

$S \Rightarrow SS^+ \Rightarrow SSS^{*+} \Rightarrow SSa^{*+} \Rightarrow Saa^{*+} \Rightarrow aaa^{*+}$.

11.

Essendo TUTTI e quattro i conflitti risolti a favore dello SHIFT, la valutazione di ogni operazione è ritardata, finché non si legge il carattere '\$' di terminazione dell'input. In altre parole, le operazioni vanno compute da Dx verso Sx (l'esatto contrario del caso in cui i conflitti sono risolti a favore del reduce).

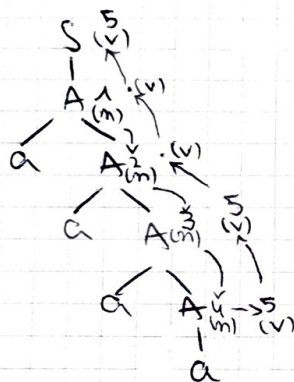
Siccome la parola 'nanbn' appartiene al linguaggio, segue che '2a3b4' produce

14.

12. (COSTRUZIONE IDENTICA AD ES. 12 DEL 2021-01_BIS)

12. $w = aaaaa$

Un possibile albero di derivazione per w è il seguente



$S.v = 5$

per $w = aaaaa$.

13.

È facile osservare come i body delle produzioni ' $A \rightarrow ab$ ' e ' $B \rightarrow abb$ ' differiscano per un solo terminale 'b' (in coda).

Supponendo di aver consumato dall'input la sequenza 'aab' e che il prossimo carattere in lettura sia 'b', il parser non riuscirà a decidere se ridurre sulla produzione ' $A \rightarrow ab$ ' o spostarsi (con shift) verso uno stato dove sarà invece possibile ridurre su ' $B \rightarrow abb$ '.

Per questo motivo, la grammatica certamente NON è LR(1).