

Esame LFC appello Febbraio 2023

Esercizio 1

Data la seguente tabella per il parsing bottom-up.

	a	b	\$	S	A
0	s4	s3		1	2
1	s4	s3	acc	6	5
2	s4	s3		7	2
3	r2	r2	r2		
4	r3	r3			
5	r4	r4		7	2
6	s4	s3		6	5
7	s4	s3	r1	6	5

generata dalla seguente grammatica:

$S \rightarrow AS$

$S \rightarrow b$

$A \rightarrow a$

$A \rightarrow SA$

Quali sono le prime 5 mosse di shift/reduce nell'analisi della stringa: abab

Esercizio 2

Data la seguente grammatica G, se G è ambigua rispondere "AMBIGUA" e fornire due derivazioni che permettono di dedurre l'ambiguità di G. Se invece G non è ambigua rispondere "NON AMBIGUA".

G:

$S \rightarrow AB \mid C$

$A \rightarrow aAb \mid ab$

$B \rightarrow cBd \mid cd$

$C \rightarrow aCd \mid aDd$

$D \rightarrow bDc \mid bc$

Esercizio 3

$r = \epsilon \mid b \mid (\epsilon \mid b) (\epsilon \mid b)^* (\epsilon \mid b)$ dire quanti stati ha il DFA minimo e quanti sono finali.

Esercizio 4

Sia N_{232} l'NFA con stato iniziale A, insieme di stati finali $\{A, C, D\}$ e con la seguente funzione di transizione. Chiamiamo D il DFA ottenuto da N_{232} per subset construction, dire quanti stati ha D e quanti di questi sono finali.

	a
A	$\{B, E\}$
B	$\{C, F\}$
C	\emptyset
D	$\{B, E\}$
E	$\{C, F\}$
F	\emptyset

Esercizio 5

G_{232} :

$S \rightarrow A \mid AB$

$A \rightarrow baB \mid B$

$B \rightarrow aB \mid \epsilon$

Se la tabella T di parsing LL(1) per G_{232} non ha entry multiply defined, scrivere "NO MULTIPLI". Altrimenti indicare tutte le coppie (X,Y) tali che $T[X,Y]$ è multiply defined.

Esercizio 6

G_{232} :

$S \rightarrow A \mid AB$

$A \rightarrow baB \mid B$

$B \rightarrow aB \mid \epsilon$

Chiamiamo A l'automa LR(1) di G_{232} I stato iniziale di A, T la tabella LR(1) per G_{232} se T non contiene alcun conflitto nello stato $I[\llbracket baaa \rrbracket]$, rispondere "NO CONFLICT". Altrimenti, per ciascuna X tale che $T[I[\llbracket baaa \rrbracket], X]$ specificare il conflitto e le riduzioni coinvolte.

Esercizio 7

G_{232} :

$S \rightarrow A \mid AB$

$A \rightarrow baB \mid B$

$B \rightarrow aB \mid \epsilon$

Sia A l'automa LALR(1) di G_{232} , H lo stato iniziale di A , T la tabella di parsing LALR(1) per G_{232} . Se non ci sono riduzioni in $H[\llbracket ba \rrbracket]$ di T rispondere "NO RIDUZIONI" altrimenti per ciascuna X tale che $T[H[\llbracket ba \rrbracket], X]$ specificare X .

Esercizio 8

Sia D la seguente porzione incompleta di SDD

```
P->S          S.next = newlabel()
               P.code = S.code D label(S.next)
```

$S \rightarrow \text{loop } S1 \text{ break on } B \text{ else } S2 \text{ endloop}$

Assumendo che:

La condizione B è gestita con gli usuali attributi $B.code$, $B.true$ e $B.false$.

La semantica del comando $\text{loop } S1 \text{ break on } B \text{ else } S2 \text{ endloop}$ è la seguente: si esegue $S1$ se al termine di tale computazione B è vera, allora l'intero comando termina.

Altrimenti si esegue $S2$ e poi l'intero comando loop è eseguito nuovamente.

Dire quali regole semantiche vanno associate all'ultima riduzione per ottenere la corretta traduzione del loop-statement .

Esercizio 9

a è $\text{array}(3, \text{array}(5, \text{integer}))$ il tipo b è $\text{array}(5, \text{array}(2, \text{integer}))$ la base di a è zero, la base di b è zero, d i j k sono interi; la dimensione di un intero è 4. Dire quale codice viene generato, usando la syntax-directed translation vista in classe nell'analisi bottom-up di $c = d + a[i][j] + b[h][k]$

Esercizio 10

Sia G la grammatica ambigua con insieme di terminali $\{a, b, \text{if}, \text{then}, \text{else}\}$ e con le seguenti produzioni:

$S \rightarrow \text{if } b \text{ then } S \text{ else } S \mid \text{if } b \text{ then } S \mid a$

Fornire una grammatica SLR(1) per la generazione di $L(G)$ in cui l'ambiguità di G è risolta rispettando la convenzione dello innermost binding secondo cui ogni else deve essere accoppiato al più vicino then non ancora accoppiato.