

compilatori AA 2009-2010

1. Data la seguente grammatica aumentata con $V_T = \{ \{ , \} , ; , \text{cmd} \}$:

$S' \rightarrow S$

$S \rightarrow \{ L \}$

$S \rightarrow \text{cmd}$

$L \rightarrow S$

$S \rightarrow L ; S$

i) costruire l'automa LR(0);

ii) costruire la tabella di parsing SLR(0);

iii) usando le regole SLR(0), individuare eventuali conflitti **shift-reduce**;

iv) risolvendo in modo arbitrario tutti gli eventuali conflitti mostrare tutti i passi del parser SLR(0) assumendo di avere in ingresso la stringa $\{ \text{cmd} ; \{ \text{cmd} ; \text{cmd} \} \}$

2. Effettuare la traduzione in codice intermedio di

if($x < 3$) $z=1$; **else** $z=2$;

3) Data la grammatica G:

$S \rightarrow a$

$S \rightarrow DES$

$D \rightarrow E$

$D \rightarrow d$

$E \rightarrow \epsilon$

$E \rightarrow b$

i) calcolare gli insiemi **FIRST** e **FOLLOW** per ogni non-terminale;

ii) G è LL(1)?

1. Data la seguente grammatica aumentata con $V_T = \{ a, b, c, x \}$:

(0) $S' \rightarrow S$

(1) $S \rightarrow a S c$

(2) $S \rightarrow B x$

(3) $B \rightarrow B b$

(4) $B \rightarrow b$

i) costruire l'automa LR(0);

ii) costruire la tabella di parsing SLR(1);

iii) usando le regole SLR(1), individuare eventuali conflitti (**in caso di assenza di conflitti, scrivere esplicitamente: NON ci sono conflitti**).

iv) risolvendo tutti gli eventuali conflitti, mostrare tutti i passi del parser SLR(1) assumendo di avere in ingresso la stringa $\{ aabxxcc \}$

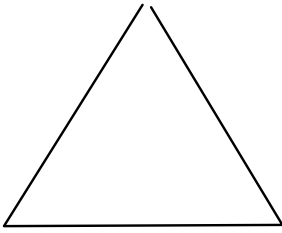
2. Effettuare la traduzione in codice intermedio di

x < 3 && y > 4 || x > 2

assumendo che gli attributi iniziali $B.true$ e $B.false$ per l'espressione booleana siano come in figura

compilatori AA 2009-2010

B.code=
B.true= LT
B.false= LF



N.B. nel costruire il parse tree si consideri che && ha precedenza su ||

3) Si consideri la grammatica G

$E \rightarrow E \mathbf{a} T | T$

$T \rightarrow T \mathbf{b} F | F$

$F \rightarrow z E \mathbf{w} | c$

i) calcolare gli insiemi **FIRST** e **FOLLOW** per ogni non-terminale;

ii) G è LL(1)? (si risponda usando la definizione di LL(1))

iii) si applichi l'algoritmo di eliminazione della ricorsione sinistra alla grammatica G.

1. Data la seguente grammatica aumentata con $V_T = \{\mathbf{x, i, u, p}\}$:

$S' \rightarrow S$

$S \rightarrow S \mathbf{x} A$

$S \rightarrow A$

$A \rightarrow E$

$A \rightarrow \mathbf{i u} E$

$E \rightarrow \mathbf{i}$

$E \rightarrow E \mathbf{p i}$

i) costruire l'automa LR(0);

ii) costruire la tabella di parsing SLR;

iii) individuare eventuali conflitti

2. Effettuare la traduzione in codice intermedio di

if(x < y) x=x+1; **else** y=x;

3) Data la seguente grammatica con $V_T = \{\mathbf{a, e, b, c, d}\}$:

$S \rightarrow aSe$

$S \rightarrow B$

$B \rightarrow bBe$

$B \rightarrow C$

$C \rightarrow cCe$

$C \rightarrow d$

i) calcolare gli insiemi **FIRST** e **FOLLOW**;

ii) G è LL(1)? rispondere applicando la definizione di LL(1).

compilatori AA 2009-2010

1. Data la seguente grammatica aumentata:

$S' \rightarrow S$

$S \rightarrow A$

$A \rightarrow B \mid Ba A$

$B \rightarrow b C$

$C \rightarrow Cb \mid Cc \mid \epsilon$

- costruire l'automa LR(0);
- costruire la tabella di parsing SLR;
- individuare eventuali conflitti

2. Effettuare la traduzione in codice intermedio di

```
while((x < 2 || x > -1) && y == 3) y = x;
```

3) Data la seguente grammatica :

$S \rightarrow A$

$A \rightarrow B \mid Ba A$

$B \rightarrow b C$

$C \rightarrow Cb \mid Cc \mid \epsilon$

calcolare gli insiemi **FIRST** e **FOLLOW**;

- G è LL(1)? rispondere applicando la definizione di LL(1);
- Eliminare ricorsione sinistra e fattorizzare a sinistra.

1. Data la seguente grammatica aumentata con $V_T = \{a, b, c, d\}$:

$S' \rightarrow S$

$S \rightarrow dca$

$S \rightarrow dAb$

$A \rightarrow c$

- costruire l'automa LR(0);
- costruire la tabella di parsing SLR;
- La grammatica è SLR? (usando le regole SLR, individuare eventuali conflitti nella tabella di parsing);
- mostrare l'evoluzione del parser nel riconoscimento della stringa **dcba**.

2. Effettuare la traduzione in codice intermedio di

```
y = 4; while(x < y) x = x + 1;
```

3) Data la grammatica G:

$S \rightarrow *$

$S \rightarrow MNS$

$M \rightarrow N \mid -$

$N \rightarrow \epsilon \mid +$

- calcolare gli insiemi **FIRST** e **FOLLOW** per ogni non-terminale;
- G è LL(1)? (è **obbligatorio** rispondere utilizzando la definizione di grammatica LL(1))