

COMPILATORI - ESERCIZI D'ESAME

Per ognuno dei seguenti assegnamenti, generare il codice a tre indirizzi assumendo che tutti gli elementi degli array siano interi di 4 byte ciascuno;

- 1) $x = a/(b+c) - d*(e+f)$;
 - 2) $x = a[i] + b[j]$;
 - 3) $a[i][j] = b[i][k] + c[k][j]$;
 - 4) $x = a[b[i][j]][c[k]]$;
 - 5) $*p++ = a[*q++]$;
-

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

```
S' → S
S → S x A
S → A
A → E
A → i u E
E → i
E → E 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;
  z=1;
```

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

```
S → aSe
S → B
B → bBe
B → C
C → cCe
C → d
```

- i) calcolare gli insiemi **FIRST** e **FOLLOW**;
- ii) G è LL(1)? (Motivare la risposta).

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

$S' \rightarrow S$
 $S \rightarrow sLd$
 $S \rightarrow c$
 $L \rightarrow S$
 $S \rightarrow L?S$

- 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) dopo aver risolto tutti gli eventuali conflitti **shift-reduce** scegliendo **shift**, mostrare tutti i passi del parser SLR(0) assumendo di avere in ingresso la stringa **sc?sc?cdd**

2. Effettuare la traduzione in codice intermedio di

$x=2; \mathbf{while}(x < 3 \ \&\& \ 1 < 2) \ x=x+4;$

(la traduzione deve essere effettuata usando le regole di traduzione illustrate a lezione).

Dopo aver prodotto l'albero annotato, si scriva il codice completo corrispondente alla traduzione dello statement.

3. Data la grammatica G:

$S \rightarrow ?$
 $S \rightarrow ZFS$
 $Z \rightarrow F$
 $Z \rightarrow >$
 $F \rightarrow \varepsilon$
 $F \rightarrow \#$

- i) calcolare gli insiemi **FIRST** e **FOLLOW**;
 - ii) G è LL(1)? Si risponda costruendo la tabella LL(1).
-

COMPILATORI - ESERCIZI D'ESAME

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

S'	\rightarrow	S
S	\rightarrow	$AaBB$
A	\rightarrow	Cd
A	\rightarrow	Ab
C	\rightarrow	e
B	\rightarrow	aA
B	\rightarrow	b

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

2. Effettuare la traduzione in codice intermedio di

`while((z>x||z==x)&&(z<3||x>5))z=0;`

Esibire l'albero di parsing annotato con gli attributi e la traduzione finale del codice.

3. Data la grammatica G con $V_T = \{a, d, x, z, s\}$

$A \rightarrow CxA$
 $A \rightarrow \epsilon$
 $B \rightarrow zCu$
 $B \rightarrow sC$
 $C \rightarrow aBa$
 $C \rightarrow d$

Scrivere la tabella di parsing LL(1) per G dopo aver calcolato gli insiemi FIRST e FOLLOW.

La grammatica è LL(1)? (**rispondere utilizzando la definizione di grammatica LL(1)**).

COMPILATORI - ESERCIZI D'ESAME

Data la grammatica G con $V_T = \{a, b, c, d, f\}$

$E \rightarrow TZ$

$Z \rightarrow fTZ$

$Z \rightarrow \varepsilon$

$T \rightarrow FU$

$U \rightarrow bFU$

$U \rightarrow \varepsilon$

$F \rightarrow d$

$F \rightarrow aEc$

i. Scrivere la tabella di parsing LL(1) per G dopo aver calcolato gli insiemi FIRST e FOLLOW.

ii.

La grammatica è LL(1)? (rispondere utilizzando la tabella di parsing).

iii.

Costruire per la stessa grammatica l'automa LR(0) e LR(1).
