

Appello di  
Linguaggi di Programmazione e Compilatori  
(A.A. 2007/2008)  
3 ore

Martedì, 09 settembre 2008

**Esercizio 1 - (9 Punti)**

Si consideri il linguaggio  $\mathcal{L} = \{a^n w v b^n \mid n \geq 1 \wedge w, v \in \{a, b, c\}^* \wedge w = \bar{w} \wedge v = \bar{v}\}$  e se ne determini la classe di appartenenza in accordo alla classificazione di Chomsky.

- Si definisca un'automa capace di accettare il linguaggio fornendo la definizione di tutte le sue componenti, e commentando altresì le scelte effettuate.
- si derivi una grammatica capace di generare il linguaggio.

**Esercizio 2 - (15 Punti)**

Si consideri la seguente grammatica G:

$$S \longrightarrow ABC \mid CBA \quad A \longrightarrow aA \mid \varepsilon \quad B \longrightarrow bB \mid b \quad C \longrightarrow aC \mid c \quad (1)$$

commentando adeguatamente:

1. si derivino gli insiemi FIRST, FOLLOW e *nullable* per G. Nella derivazione degli insiemi si annotino i vari simboli con l'indice dell'iterazione e il riferimento alla produzione che hanno richiesto l'aggiunta del simbolo all'insieme;
2. si discuta l'applicabilità del parsing LL(1).
3. si derivi l'automa LR(0) e le tabelle di parsing LR(0) ed SLR discutendo altresì l'applicabilità dei due differenti tipi di parsing in caso di non applicabilità si faccia attenzione ad evidenziare tutti i possibili conflitti.

### Esercizio 3 - (9 punti)

Dopo aver definito la tipologia di parser generato (LL,LR,SLR,LALR) da JavaCC si discutano le tecniche messe a disposizione da questo strumento al fine di aumentare i simboli di lookahead da prendere in considerazione prima di decidere la produzione da applicare. (**ATTENZIONE! Questa parte dell'esercizio deve essere svolta ANCHE da chi ha presentato il progetto**)

Con riferimento al seguente frammento di codice si discutano possibili difficoltà nella generazione del parser e si identifichi una possibile soluzione.

```
options {
    JDK_VERSION = "1.5";
    STATIC = false;
}

PARSER_BEGIN(pp)
public class pp {
    public static void main(String args[]) throws ParseException {
        pp parser = new pp(System.in);
        parser.Start();
    }
}
PARSER_END(pp)

void Start() : {}
{
    "a" "b" AB() | "b" Start() "b"
}

void AB(): {}
{
    "a" "b" AB() | A()
}

void A() : {}
{
    "a"
}
```