

Linguaggi Formali e Compilatori

Proff. Breveglieri, Crespi Reghizzi, Morzenti

Prova scritta¹: Domanda relativa alle esercitazioni

27/06/2009

COGNOME:

NOME: Matricola:

Iscritto a: ☐ Laurea Specialistica ☐ V. O. ☐ Laurea Triennale ☐ Altro:.....

Sezione: ☐ Prof. Breveglieri ☐ Prof. Crespi ☐ Prof. Morzenti

Per la risoluzione della domanda relativa alle esercitazioni si deve utilizzare l'implementazione del compilatore **Acse** che viene fornita insieme al compito.

Si richiede di modificare la specifica dell'analizzatore lessicale da fornire a **flex**, quella dell'analizzatore sintattico da fornire a **bison** ed i file sorgenti per cui si ritengono necessarie delle modifiche in modo da estendere il compilatore **Acse** con la possibilità di usare degli assegnamenti come parti di espressioni, similmente al linguaggio C, come nell'esempio:

```
int x, y, z, a[3];

x = y = z = a[0] = 1;
write( x = y + 3 );
z = (y = 2) * x;
x = z + y;
write( x );
```

Se tale programma venisse compilato e fatto girare, allora verrebbero stampati in uscita i valori "4" e "10".

La soluzione deve rispettare le seguenti specifiche:

- L'operatore di assegnamento ha priorità **inferiore** a qualunque operatore binario ed è associativo a **destra**.
- Un'espressione di assegnamento deve poter apparire nella grammatica ovunque possa apparire un'espressione (non terminale **exp**).
- L'operando sinistro in un'espressione di assegnamento può essere sia una variabile che un elemento di un array.
- L'operando destro può essere una qualunque espressione del linguaggio valida (incluso un assegnamento)

Si faccia attenzione che la grammatica risultante non sia ambigua.

¹Tempo 45'. Libri e appunti personali possono essere consultati.

È consentito scrivere a matita. Scrivere il proprio nome sugli eventuali fogli aggiuntivi.

Si espliciti ogni eventuale ulteriore assunzione che sia ritenuta necessaria a completare la specifica data.

1. Definire le regole sintattiche (e/o le modifiche a quelle esistenti) necessarie per ottenere le funzionalità richieste, incluse eventuali direttive per la precedenza e l'associatività. (7 punti)
2. Definire le azioni semantiche (e/o le modifiche a quelle esistenti) necessarie per ottenere le funzionalità richieste. (8 punti)

3. Dato il seguente codice:

```
int a = 2;
int b;

b = a*5 + (a = a - 1) + a*2;
write( b )
```

- (a) Scrivere l'albero sintattico relativo all'istruzione di assegnamento (terza riga) partendo dalla grammatica Bison definita in Acse.y e considerando le modifiche introdotte nei punti precedenti, *iniziando dal non-terminale **statements***. (4 punti)
- (b) Secondo la specifica Bison definita in Acse.y e modificata nei punti precedenti, dire in che ordine vengono valutati i tre addendi nell'istruzione sopra analizzata e calcolare di conseguenza il valore finale della variabile **b**. (5 punti)

4. Nel frammento d'esempio riportato nella domanda 3 il valore assunto dalla variabile `b` dipende dall'ordine in cui sono valutati gli operandi dell'addizione. Tipicamente la specifica di un linguaggio non impone quale sia l'ordine di valutazione degli operandi, e quindi espressioni come quella nell'esempio sarebbero da evitare. Si esponga in maniera sommaria quale controllo semantico può essere eseguito dal compilatore per avvisare di potenziali situazioni come quella nell'esempio. Non è necessario scrivere codice, ma andrebbe comunque indicato in quali azioni semantiche vengono eseguite le operazioni descritte, e quali strutture dati sono coinvolte. (9 punti)