

# Linguaggi Formali e Compilatori

## Proff. Breveglieri, Crespi Reghizzi, Morzenti

### Prova scritta<sup>1</sup>: Domanda relativa alle esercitazioni

### 28/06/2008

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 gestire un costrutto di tipo "computed goto", del quale viene qui fornito un esempio.

```
int a;  
...  
lab3: ...  
...  
goto a in lab1,mialab,lab3;  
...  
lab1: ...  
...  
mialab: ...  
...
```

La semantica è la seguente: se a run-time la variabile **a** vale 1 viene effettuato un salto a **lab1**, se vale 2 a **mialab**, se vale 3 a **lab3**. In generale, la lista di etichette può essere lunga a piacere, e il valore della variabile indica il posizione nella lista dell'etichetta a cui saltare. Se il valore della variabile è minore di 1 oppure maggiore del numero di etichette in lista, non viene eseguito alcun salto.

Allo scopo di risolvere correttamente l'esercizio, ricordiamo l'esistenza della funzione **getNewRegister**, contenuta in **axe\_engine.[h,c]**, che è in grado di riservare un registro libero per eventuali elaborazioni. Il prototipo della funzione è il seguente:

---

<sup>1</sup>Tempo 45'. Libri e appunti personali possono essere consultati.  
È consentito scrivere a matita. Scrivere il proprio nome sugli eventuali fogli aggiuntivi.

```
/* get a register still not used. This function returns  
 * the ID of the register found */  
int getNewRegister(t_program_infos *program);
```

L'intero ritornato dalla funzione rappresenta, in maniera non ambigua, il registro allocato.

2. Definire le regole sintattiche necessarie per ottenere la funzionalità richiesta.  
(8 punti)

3. Definire le azioni semantiche necessarie per ottenere la funzionalità richiesta.  
(17 punti)

4. Definire le azioni semantiche necessarie per segnalare la presenza di salti a etichette non dichiarate. (5 punti)