

# Linguaggi Formali e Compilatori

Proff. Breveglieri, Morzenti

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

03/09/2014

COGNOME: .....

NOME: ..... Matricola: .....

Corso: ☐ Laurea Specialistica    ☐ V. O.    ☐ Laurea Triennale    ☐ Altro: ...

Sezione: ☐ Prof. Breveglieri    ☐ 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 il costrutto di `if` aritmetico. Si consideri il seguente snippet di codice come **esempio** dell'utilizzo del costrutto di `if` aritmetico:

```
int a,b;

over: a=b;
if A( a-b ) A sum,over,print;
b=b*2;
print: write(a);
sum: a=a+b;
```

Il costrutto di `if` aritmetico valuta l'espressione racchiusa tra le parentesi speciali `A ( e ) A` e salta a una delle tre etichette che sono fornite dopo l'espressione parentetizzata. In particolare, nel caso l'espressione sia negativa, salta alla prima; se è nulla, salta alla seconda; se è positiva, alla terza. La sintassi delle etichette e degli statement etichettati è la stessa del linguaggio C. E' possibile utilizzare al più un'etichetta per statement. Nel caso un costrutto di `if` aritmetico si riferisca a un'etichetta non presente nel programma, è necessario segnalare un errore a *compile-time*.

---

<sup>1</sup>Tempo 60'. Libri e appunti personali possono essere consultati.  
È consentito scrivere a matita. Scrivere il proprio nome sugli eventuali fogli aggiuntivi.

1. Definire i token (e le relative dichiarazioni in `Acse.lex` e `Acse.y`) necessari per ottenere la funzionalità richiesta. (2 punti)
2. Definire le regole sintattiche (o le modifiche a quelle esistenti) necessarie per ottenere la funzionalità richiesta. (3 punti)

3. Definire le azioni semantiche (o le modifiche a quelle esistenti) necessarie per implementare il costrutto di `if` aritmetico. (20 punti)



4. Data il seguente snippet di codice `Lance`:

```
; a = -a & b;
```

Scrivere l'albero sintattico relativo partendo dalla grammatica Bison definita in `Acse.y` *iniziando dal non-terminale* `statements`. (5 punti)

5. (**Bonus**) É possibile utilizzare (impropriamente) l' `if` aritmetico per costruire costrutti iterativi simili al più tipico `do-while`. Si descriva come individuare se è stato effettuato questo utilizzo improprio a compile-time ed emettere un avvertimento per il programmatore.