

# Linguaggi Formali e Compilatori

## Proff. Breveglieri, Crespi Reghizzi, Morzenti

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

### 06/09/2013

COGNOME: .....

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

Corso: ☐ 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 il costrutto *either-or-on*.

```
int a, z, b, k;

// ...

either {
    read(a);
    a = a - 99;
} or {
    z = 27;
    a = z - b;
    a = a * a;
} on a < 27 * k;
```

Figura 1: Esempio

Il costrutto *either-or-on* è simile al costrutto *if-then-else*: quando la condizione (specificata dopo la keyword **on**) è verificata sarà eseguito il blocco *either*, altrimenti sarà eseguito il blocco *or*.

---

<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. (3 punti)
2. Definire le regole sintattiche (o le modifiche a quelle esistenti) necessarie per ottenere la funzionalità richiesta. (4 punti)
3. Definire le azioni semantiche (o le modifiche a quelle esistenti) necessarie per ottenere la funzionalità richiesta. (18 punti)

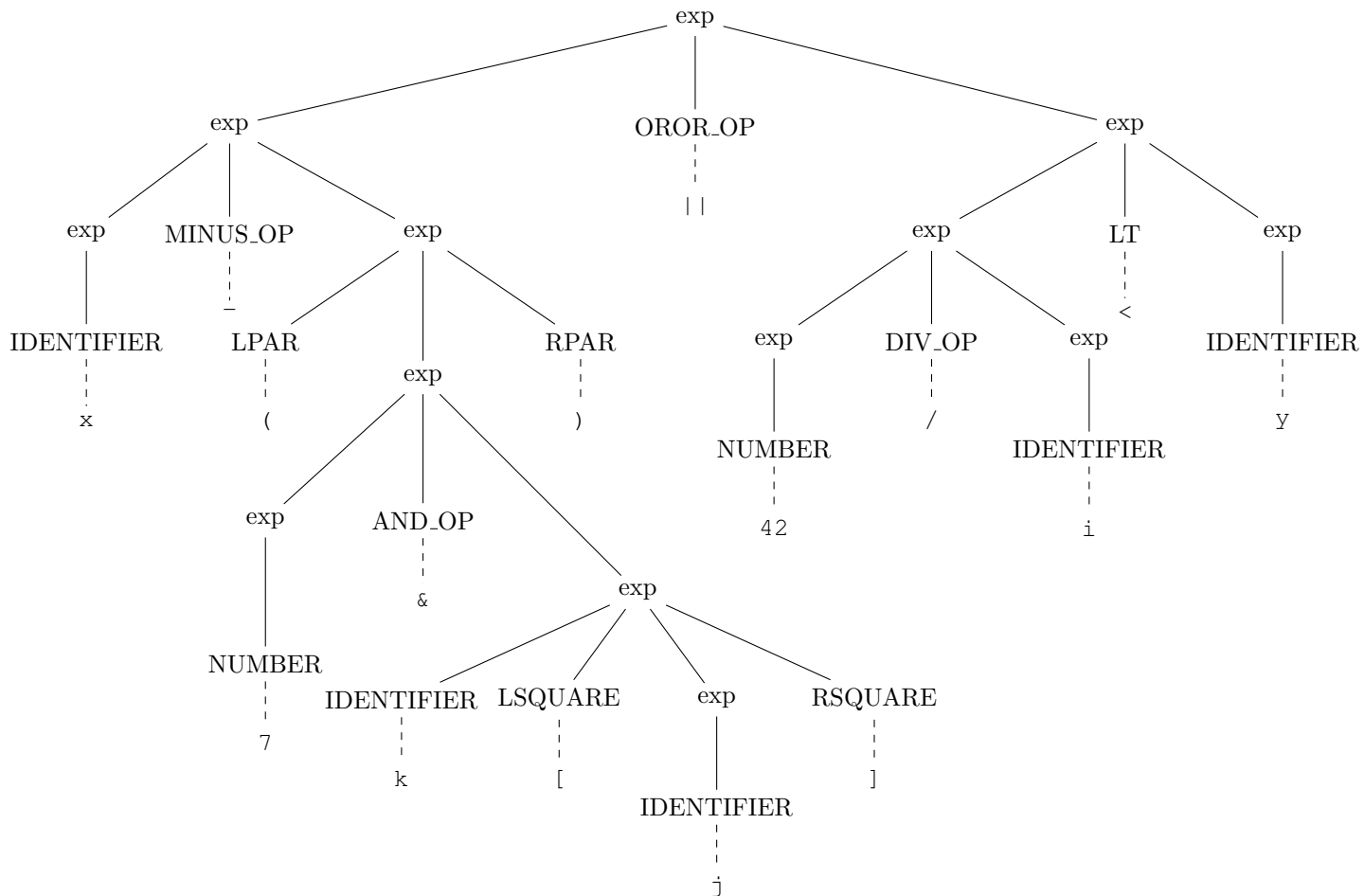
La soluzione è riportata nella patch allegata.



4. Data il seguente snippet di codice Lance:

`x - (7 & k[j]) || 42 / i < y`

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



5. (**Bonus**) Si indichi come estendere il costrutto *either-or-on* in modo da ripetere l'esecuzione dello stesso fino a quando non è verificata una condizione data:

```
either {  
    a[j] = h[j];  
    j = j + 1  
} or {  
    b[j] = h[j];  
    j = j + 2;  
} on h[j] - h[j]/2  
until j >= 2*z;
```