Linguaggi Formali e Compilatori
Proff. Breveglieri, Crespi Reghizzi, Morzenti
Prova scritta [1]: Domanda relativa alle esercitazioni
28/06/2013

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 *array comprehension* per assegnare valore ad un vettore di interi.

```
int i, x[5], y[7];

x[0] = 1; x[1] = 2; x[2] = 3;
x[3] = 4; x[4] = 5;

// y = {-2, 1, 6, 13, 22, undef, undef}
y = [i * i - 3 for i in x];

write(y[3]);
```

Figura 1: Esempio

Ad ogni posizione del vettore destinazione (`y`) è assegnato il valore di una espressione che dipende dal valore dell'elemento (`i`) nella medesima posizione del vettore sorgente (`x`). **Si noti che:**

- le variabili coinvolte devono essere dichiarate
- il costrutto non richiede che le dimensioni dei vettori siano uguali.

[1] 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)

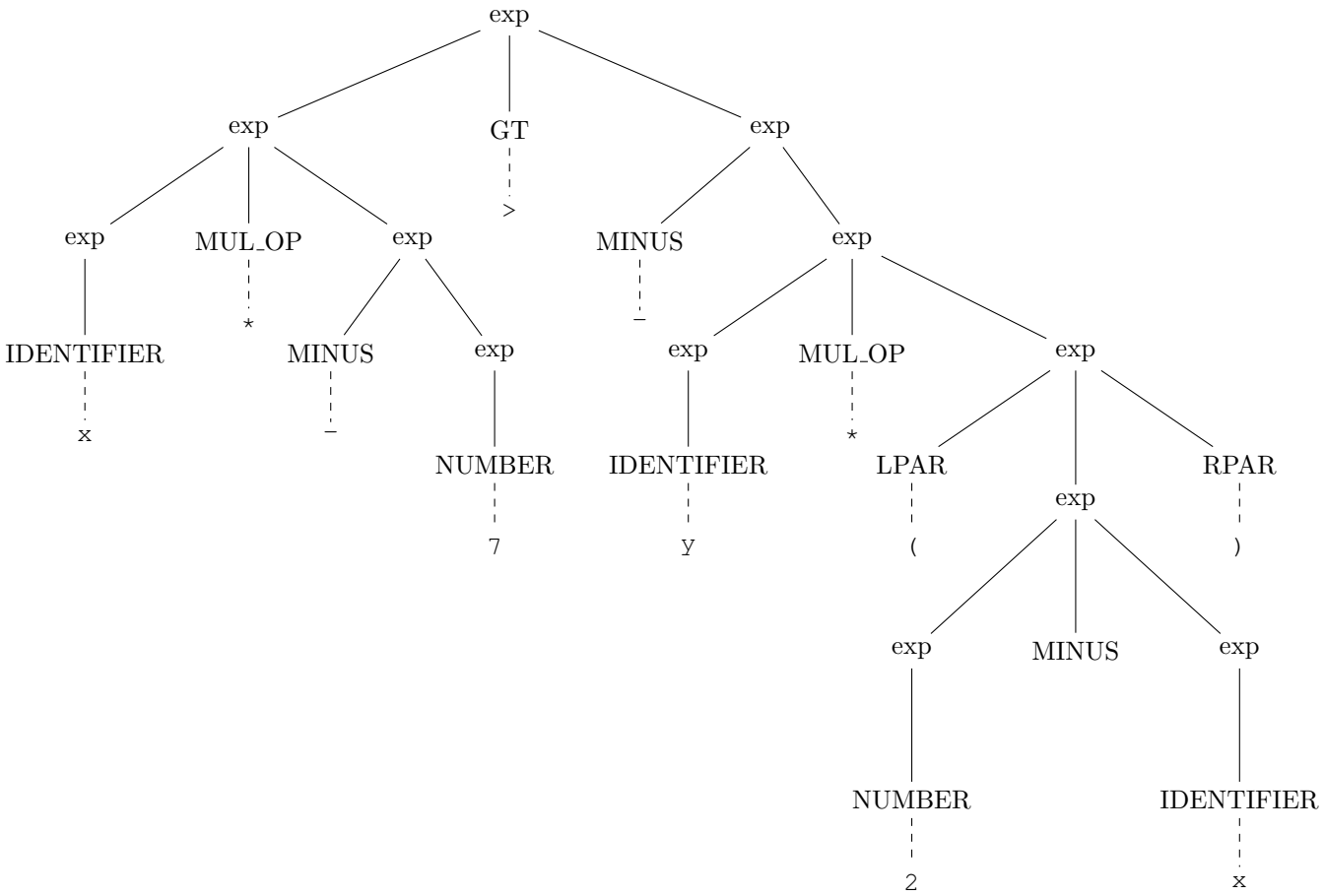
3. Definire le azioni semantiche (o le modifiche a quelle esistenti) necessarie per ottenere la funzionalità richiesta. (18 punti)

   La soluzione è riportata nella patch allegata.

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

```
x * -7 > -y * (2 - x)
```

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



5. (**Bonus**) Si mostri un possibile modo di estendere il supporto al costrutto *array comprehension* in modo da poter generare i valori da assegnare anche dalla specifica di un intervallo di valori discreti $e \in (a, b)$.

```
int y[7], b, e;
read(b);
y = [2 * e - 7 for e in range(3, b)];
```