

Linguaggi Formali e Compilatori
Proff. Breveglieri, Crespi Reghizzi, Sbattella
Prova scritta¹: Domanda relativa alle esercitazioni
07/07/2006

COGNOME:
NOME: Matricola:
Iscritto a: ☐ Laurea Specialistica ☐ V. O. ☐ Laurea Triennale ☐ Altro:.....
Sezione: ☐ Prof. Breveglieri ☐ Prof. Crespi ☐ Prof.ssa Sbattella

Per la risoluzione della domanda relativa alle esercitazioni si deve utilizzare l'implementazione del compilatore **Simple** 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 **Simple** con la possibilità di risolvere durante la compilazione le espressioni e sotto-espressioni costanti:

```
declarations
  integer a, b, c.
  const integer d=1, e=2.
  ...
begin
  ...
  a := d + 3;
  b := e * 7 + d;
  c := e + 5 + a;
  ...
end
```

che risulta equivalente al codice Simple:

```
declarations
  integer a, b, c.
  ...
begin
  ...
  a := 4;
  b := 15;
  c := 7 + a;
```

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

```
...  
end
```

dove le espressioni $d+3$, $e*7+d$ e $5+e$ sono costanti e sono sostituite dai relativi valori. Il compilatore deve generare il bytecode per SimpleVM che esegua direttamente le operazioni semplificate (*non* deve invece generare il codice sorgente modificato, che é proposto qui sopra solo per facilitare la comprensione del problema).

Le modifiche devono mettere il compilatore **Simple** in condizione di analizzare la correttezza sintattica dei costrutti sopra descritti e di generare una traduzione corretta nel linguaggio assembler della macchina **SimpleVM**.

1. Definire i token (e le relative dichiarazioni in Simple.lex e Simple.y) necessari per ottenere la funzionalità richiesta.
2. Definire le regole sintattiche necessarie per ottenere la funzionalità richiesta.
3. Definire le azioni semantiche necessarie per ottenere la funzionalità richiesta.