Corso di Laurea Magistrale in Informatica

Compito di Compilatori e Interpreti

17 Settembre 2021

Nota Bene. Alla fine del compito, fare una foto a tutto il compito col cellulare usando una applicazione che esegue scansioni, tipo CamScanner, e inviarla per email a `cosimo.laneve@unibo.it` .

Si consideri la seguente grammatica (scritta in ANTLR)

```
prg : stm ;
stm : (Id '=' exp ';' | '{' dec stm '}')+ ;
dec : (type Id ';')+ ;
type: 'int' | 'bool' ;
exp : Int | Bool | Id | exp '+' exp | exp '-' exp | exp '>' exp | exp '==' exp
    | exp '||' exp | exp '&&' exp ;
```

dove

- gli `Int` sono sequenze non vuote di cifre senza segno oppure prefissate dal segno `+` o `-`;
- i `Bool` sono i valori "`true`" e "`false`";
- gli `Id` sono gli identificatori (sequenze non vuote di caratteri);
- le operazioni di somma "`+`", sottrazione "`-`" e maggiore "`>`" si applicano a espressioni `Int`, le operazioni di or "`||`" e and "`&&`" si applicano a espressioni `Bool`, mentre l'operazione di uguaglianza si applica a espressioni dello stesso tipo.

Esercizi

1. dare tutte le regole di inferenza per la verifica dei tipi del linguaggio di sopra (attenzione che il linguaggio ammette annidamento di ambienti);
2. verificare, scrivendo l'albero di prova, che il programma seguente sia correttamente tipato:

   `{ int x; int y; x = 5; { bool z; z = (x > 5)||false; { int z; y = 6+x; z = 3 + y; }}}`
3. definire il codice intermedio per tutti i costrutti del linguaggio. In particolare, si ricordi che un booleano occupa 1 byte mentre un intero occupa 4 byte. Inoltre, per quanto riguarda le operazioni `||` e `&&`, si implementi la cosiddetta *lazy evaluation*: il secondo argomento non viene valutato se inutile (nell' `||`, il secondo argomento non viene valutato se il primo è `true`, nell' `&&` il secondo argomento non viene valutato se il primo è `false`).
4. scrivere il codice generato per l'esempio al punto 2.