

Corso di Laurea Magistrale in Informatica

Compito di Compilatori e Interpreti

19 Dicembre 2019

Esercizio 1 (6 punti) Scrivere le definizioni formali di `nullable`, `first`, e `follow` per grammatiche LL(1).

Esercizio 2 (10 punti) In un linguaggio di programmazione i programmi `Prg` sono definiti da questa sintassi

$$\begin{aligned} \text{Prg} &::= \text{Fun}^* \text{Stm} \\ \text{Fun} &::= \text{Type Id } \text{"(" FPar ")} = \text{Stm} \end{aligned}$$

dove `Type` possono essere solamente `int` e `bool`, `FPar` sono i parametri formali, cioè sequenze anche vuote del tipo `Type1 Id1, ..., Typen Idn`, e `Stm` è la categoria sintattica dei comandi (lo `Stm` in `Prg` è il *main*). Definire

1. le regole di inferenza per analizzare programmi con mutua ricorsione [Suggerimento: servono due regole, una per costruire l'ambiente iniziale con tutti i tipi delle funzioni, l'altra per analizzare il programma];
2. definire lo pseudocodice per `CheckProg` che implementa le regole di sopra;
3. fornire l'albero di prova per il programma

```
int f(int x) = return (g(x,x) + 1) ;
int g(int u, int v) = return(f(u+v)) ;
print(f(1)+g(2,3)) ;
```

assumendo i vincoli di tipo standard per i comandi e le espressioni (quelli visti a lezione).

Esercizio 3 (8 punti)

1. Definire la funzione `code_gen` per il termine `do S while E` che esegue `S`, quindi controlla `E` e se essa è vera riesegue `S`, altrimenti l'esecuzione termina.
2. Come verifica, si generi il codice di

```
do do ( x:= x+1 ; y:= y+x ) while (x>y) while (y<x+z)
```

dove le variabili `x`, `y` e `z` si trovano ad offset `+4` e `+8` e `+12` del frame pointer `FP`.