

Corso di Laurea Magistrale in Informatica

Compito di Compilatori e Interpreti

19 Febbraio 2020

Esercizio 1 (7 punti). Data la grammatica (le lettere minuscole sono simboli terminali, A è il simbolo iniziale)

$$\begin{aligned} A &\rightarrow BC \\ B &\rightarrow aB \mid \varepsilon \\ C &\rightarrow CbB \mid c \end{aligned}$$

Riscrivere la grammatica rimuovendo la ricorsione sinistra e verificare se la grammatica è LL(1) costruendo l'opportuna tabella. Nel caso non lo sia, esiste un k per cui essa è LL(k)? Motivare la risposta.

Esercizio 2 (7 punti). I seguenti sono potenziali regole di tipo per il costruito `let` in un linguaggio con sottotipaggio ($<:$). Dire quali regole sono corrette e quali sbagliate. Per quelle sbagliate dare (a) un codice che dovrebbe essere tipabile e non lo è; (b) un codice che è tipabile e invece non dovrebbe essere.

1.
$$\frac{\Gamma \vdash e : T' \quad \Gamma \vdash e' : T'' \quad T' <: T}{\Gamma \vdash \text{let } T \ x = e \text{ in } e' : T''}$$
2.
$$\frac{\Gamma \vdash e : T' \quad \Gamma[x : T] \vdash e' : T'' \quad T <: T'}{\Gamma \vdash \text{let } T \ x = e \text{ in } e' : T''}$$
3.
$$\frac{\Gamma \vdash e : T' \quad \Gamma[x : T'] \vdash e' : T'' \quad T' <: T}{\Gamma \vdash \text{let } T \ x = e \text{ in } e' : T''}$$

Nel caso in cui nessuna regola sia corretta, (i) dare la regola giusta e (ii) controllare che i codici di prima siano correttamente tipabili/non tipabili.

Esercizio 3 (10 punti). Definire la funzione `code_gen` per

1. la dichiarazione di funzione void come: `void f(T1 x, T2 y){ S }` ;
2. l'invocazione di funzione `f(e, e')` (e, e' sono espressioni).

Quindi, assumendo che l'etichetta che corrisponde alla seguente funzione `fact` sia `fact_label`, scrivere il codice per

```
int x = 1 ;
void fact(int n, int z){
    if (n == 0) x = z ;
    else fact(n-1, z*n) ;
}
```