Elaborati

16 gennaio 2023

0.1 Modalità di consegna

L'esercizio risolto dovrà essere consegnato tramite email a marco.lucchese@univr.it e per conoscenza a massimo.merro@univr.it.

0.2 Descrizione

Scegliere e svolgere uno degli esercizi seguenti. L'estensione prevede l'aggiunta per: interprete e typechecker visto a laboratorio dei costrutti elencati di seguito. Il file da consegnare dovrà contenere il seguente linguaggio:

$$\textsc{Linguaggio while} \tag{1}$$

$$op :: piu \mid maggioreouguale \tag{2}$$

$$e \in Expr :: n \mid b \mid e\ op\ e \mid if\ e\ then\ e\ else\ e \mid skip \mid e; e \mid while\ e\ do\ e \mid !e \mid e_1 := e_2 \mid ...estensioni... \mid \tag{3}$$

0.3 Esercizi

Di seguito ogni riga sarà anticipata da [CBN] che indicherà Call by Name oppure da [CBV] che indicherà Call by Value. Il voto è indicato a lato. (laboratorio 2 CFU)

1. (a) [CBN] Estendere il linguaggio while con funzioni
   (b) [CBN] Implementare applicazione di funzioni.
   (c) [CBN] Implementare operatori di punto fisso.
   (d) [CBN] Mostrare funzionamento implementado tramite funzione successione Fibonacci (Prendere in input n e ritornare N termini della successione di Fibonacci). [Voto 25]
   (e) [CBN] Implementare RepPar.

$$\begin{aligned} RepPar(e) \quad &\stackrel{\text{def}}{=} \quad \mathsf{let}\ P : (\mathsf{unit} \to \mathsf{unit}) \to (\mathsf{unit} \to \mathsf{unit}) \\ &\qquad = \big(\mathsf{fn}\ f : \mathsf{unit} \to \mathsf{unit} \Rightarrow (\mathsf{fn}\ x : \mathsf{unit} \Rightarrow x \parallel (f\ x))\big) \\ &\qquad \mathsf{in}\ \mathsf{fix}.P\ e \end{aligned}$$

[Voto 27]

   (f) [CBN] Implementare AwtPar.

$$\begin{aligned} &AwtPar(e) \quad \stackrel{\text{def}}{=} \\ &\mathsf{let}\ A : (\mathsf{unit} \to \mathsf{unit}) \to (\mathsf{unit} \to \mathsf{unit}) \\ &\quad = \big(\mathsf{fn}\ f : \mathsf{unit} \to \mathsf{unit} \Rightarrow (\mathsf{fn}\ x : \mathsf{unit} \Rightarrow \mathsf{await}\ \mathit{true}\ \mathsf{protect}\ x\ \mathsf{end} \parallel (f\ x))\big) \\ &\mathsf{in}\ \mathsf{fix}.A\ e \end{aligned}$$

[Voto 30]

2. (a) [CBV] Estendere il linguaggio while con funzioni
   (b) [CBV] Implementare applicazione di funzioni.
   (c) [CBV] Implementare operatori di punto fisso.
   (d) [CBV] Mostrare funzionamento implementado tramite funzione successione Fibonacci (Prendere in input n e ritornare N termini della successione di Fibonacci). [Voto 26]
   (e) [CBV] Implementare RepPar.

$$\begin{aligned} \mathit{RepPar}(e) \quad &\stackrel{\text{def}}{=} \quad \mathsf{let}\ P : (\mathsf{unit} \to \mathsf{unit}) \to (\mathsf{unit} \to \mathsf{unit}) \\ &\qquad = \left(\mathsf{fn}\ f : \mathsf{unit} \to \mathsf{unit} \Rightarrow \left(\mathsf{fn}\ x : \mathsf{unit} \Rightarrow x \parallel (f\ x)\right)\right) \\ &\qquad \mathsf{in}\ \mathsf{fix}.P\ e \end{aligned}$$

[Voto 28]

   (f) [CBV] Implementare AwtPar.

$$\begin{aligned} &\mathit{AwtPar}(e) \quad \stackrel{\text{def}}{=} \\ &\mathsf{let}\ A : (\mathsf{unit} \to \mathsf{unit}) \to (\mathsf{unit} \to \mathsf{unit}) \\ &\quad = \left(\mathsf{fn}\ f : \mathsf{unit} \to \mathsf{unit} \Rightarrow \left(\mathsf{fn}\ x : \mathsf{unit} \Rightarrow \mathsf{await}\ \mathit{true}\ \mathsf{protect}\ x\ \mathsf{end} \parallel (f\ x)\right)\right) \\ &\mathsf{in}\ \mathsf{fix}.A\ e \end{aligned}$$

[Voto 30 e lode]