

Paradigmi di Programmazione - A.A. 2021-22

Esempio di Testo d'Esame n. 4

CRITERI DI VALUTAZIONE:

La prova è superata se si ottengono almeno 12 punti negli esercizi 1,2,3 e almeno 18 punti complessivamente.

Esercizio 1 [Punti 4]

Applicare la β -riduzione alla seguente λ -espressione fino a raggiungere una espressione non ulteriormente riducibile o ad accorgersi che la derivazione è infinita:

$$(\lambda x. \lambda y. yx)(\lambda x. \lambda y. x(yy))(\lambda x. xz(\lambda y. yy))$$

Nella soluzione, mostrare tutti i passi di riduzione calcolati sottolineando ad ogni passo la porzione di espressione a cui si applica la β -riduzione (redex) ed evidenziando le eventuali α -conversioni.

Esercizio 2 [Punti 4]

Indicare il tipo della seguente funzione OCaml, mostrando i passi fatti per inferirlo:

```
let f x y z =  
  match (x y) with  
  | [] -> (z y)  
  | _::_ -> (z y) + 1;;
```

Esercizio 3 [Punti 7]

Definire, usando i costrutti di programmazione funzionale in OCaml, una funzione `g` con tipo

```
g : int list -> int * int
```

che, data una lista non vuota di interi, restituisce la coppia formata dal massimo elemento della lista e dal numero di volte che esso occorre nella lista stessa. Ad esempio: `g [1;-4;5;-1;5;-6;5] = (5,3)`. L'applicazione di `g` alla lista vuota causa invece un'eccezione.

Esercizio 4 [Punti 15]

Si consideri il linguaggio didattico funzionale MiniCaml, e se ne estenda la sintassi astratta e l'interprete del linguaggio in modo gestire il costrutto iterativo `for-each`. La sintassi concreta del costrutto è la seguente:

```
for-each (lista-interi; funzione)
```

dove `lista-interi` rappresenta una lista di interi non vuota e `funzione` è una funzione non ricorsiva che viene eseguita passandole sequenzialmente tutti i valori presenti nella lista. Il costrutto `for-each` restituisce come risultato la somma dei valori calcolati dall'invocazione della funzione.