Paradigmi di Programmazione - A.A. 2021-22

Esame Scritto del 05/07/2022

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:

$$((\lambda x.\lambda y.(xy))(\lambda z.(yz)))k$$

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

Esercizio 3 [Punti 7]

Assumendo il seguente tipo di dato che descrive alberi binari di interi:

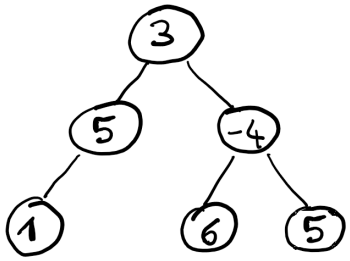
```
type btree =
| Void
| Node of int * btree * btree
```

si definisca, usando i costrutti di programmazione funzionale di OCaml, una funzione `distinct` con tipo

```
distinct : btree -> bool
```

tale che `distinct bt` restituisca `true` se i valori interi nell'albero rappresentato da `bt` sono tutti diversi tra loro, `false` se invece `bt` contiene almeno due numeri uguali.
Ad esempio dato il seguente albero binario (a destra in una rappresentazione visuale):

```
let bt =
  Node (3,
    Node (5,
      Node(1,Void,Void),
      Void
    ),
    Node (-4,
      Node(6,Void,Void),
      Node(5,Void,Void)
    )
  )
```



abbiamo che `distinct bt` restituisce `false` a causa delle due occorrenze del valore `5`.

Esercizio 4 [Punti 15]

Si estenda il linguaggio MiniCaml visto a lezione con un nuovo tipo di dato `IntSet` che permette di dichiarare insiemi di interi. Supponiamo che il linguaggio preveda operazioni primitive per costruire e operare su insiemi di interi, come descritto di seguito:

```
Empty           /* Insieme Vuoto */
Insert(n, set)  /* Operazione di inserimento di un valore intero n
                   in un insieme di interi set */
CheckAll(predicato, set) /* Operazione che restituisce il valore true
                            se tutti gli elementi dell'insieme soddisfano il
                            predicato (funzione da interi a booleani) */
```

Esempi (in sintassi concreta):

```
let set1 = Insert(3,Insert(2,Insert(1,Empty)));;  /* risultato {1,2,3} */
let set2 = Insert(2,Insert(2,Insert(1,Empty)));;  /* risultato {1,2} (2 gia' presente) */
CheckAll((fun x -> x>2), set1);;  /* risultato true */
CheckAll((fun x -> x>2), set2);;  /* risultato false */
```

Si modifichi l'implementazione dell'interprete del linguaggio con quanto serve per gestire i costrutti per operare su insiemi di interi descritti in precedenza.