

# Prova Scritta di Linguaggi di Programmazione II

03/09/2007

Si noti che quanto messo nei riquadri è una *bozza* fornita solo a *titolo indicativo*.  
Quindi **non** è un modello di soluzione completa che ci si aspetta ad un esame.

1. Si rappresenti il codice della macchina di Warren relativo al seguente programma in Prolog.

$gf([X], [], X).$

$gf([], [Y | Q], R) :- rv([Y | Q], P), gf(P, [], R).$

$gf([X], [Y | Q], R) :- rv([Y | Q], P), gf([X | P], [], R).$

$gf([X, _ | P], Q, R) :- gf(P, [X | Q], R).$

2. Si calcoli  $T_P^{ca} \uparrow 4$  per il programma  $P$  ottenuto aggiungendo al programma dell'Esercizio 1 il programma

$rv([], []).$

$rv([a(X)], [c(X)]).$

$rv([a(X), b(Y)], [a(Y), b(X)]).$

$$\begin{aligned} T_P \uparrow 1 &= \{rv([], []), rv([a(A)], [c(A)]), rv([a(A), b(B)], [a(B), b(A)]), gf([A], [], A)\} \\ T_P \uparrow 2 &= T_P \uparrow 1 \cup \{gf([], [a(A)], c(A))\} \\ T_P \uparrow 3 &= T_P \uparrow 2 \cup \{gf([a(A), B], [], c(A))\} \\ T_P \uparrow 4 &= T_P \uparrow 3 \cup \{gf([], [a(A), b(B)], c(B)), gf([a(A)], [a(B)], c(A))\} \end{aligned}$$

3. Sia dato il seguente programma Curry

$f \ x \ (\mathbf{Right} \ (\mathbf{Just} \ y)) = y : x$

$f \ [x] \ (\mathbf{Left} \ y) = x : y$

Si stabilisca se è induttivamente sequenziale e si calcolino tutte le derivazioni di *needed narrowing* per il termine  $f \ (f \ [y] \ x) \ x$  where  $x, y$  free.

Si dica inoltre se le posizioni selezionate nei vari passi delle derivazioni calcolate sono *Basic*.

Le risposte (delle implementazioni di Curry) sono

$\{x = \mathbf{Right} \ (\mathbf{Just} \ v)\} \ [v, v, y]$

$\{x = \mathbf{Left} \ []\} \ [y]$

Le derivazioni *non sono* tutte basic.

4. Si consideri il seguente frammento di codice sintatticamente ammissibile sia in Haskell che in Curry.

```
data Table a b = Void |
  Node a b (Table a b) (Table a b)
```

```
lookup _ Void = Nothing
```

```
lookup x (Node y v l r)
```

```
  | x == y      = Just v
```

```
  | x < y       = lookup x l
```

```
  | otherwise   = lookup x r
```

# Prova Scritta di Linguaggi di Programmazione II

03/09/2007

Si indichi il tipo di dato inferito nei due linguaggi.

Ci si può aspettare di ottenere i medesimi risultati nei 2 linguaggi usando la funzione su dati di tipo `Table Int a?`

**Se sì** sebbene si abbiano gli stessi risultati, potrebbe avere un qualche effetto pratico una eventuale riscrittura del codice?

**Se no** si mostri come andrebbe riscritta la versione Curry per ottenere gli stessi risultati o si spieghi perché ciò non sia possibile.

5. Si stabilisca (argomentando opportunamente) se il seguente programma è lineare destro/sinistro, duplicante, eliminante, collassante; se è constructor-based e se è ortogonale.

```
perm xs = fst $ extract xs : snd $ extract xs
extract (x:xs) = (x, xs)
extract (y:ys) = (fst $ extract ys, y : snd $ extract ys)
```

6. Si scriva un predicato PROLOG `treepath/2` che determina percorsi all'interno di un albero binario. A titolo di esempio,

```
treepath(tree(1,tree(2,tree(3,void,void),void),tree(4,void,void)),Xs)
```

restituisce 2 soluzioni per `Xs`: `[1,2,3]` e `[1,4]`. Si faccia attenzione a *non generare la stessa soluzione più volte*.

Si scriva il programma con variabili anonime ove possibile.

7. Ci sono quattro numeri (1, 3, 7, 11) che devono attraversare un ponte stretto di notte. Sul ponte non riescono a passare più di 2 numeri alla volta. I numeri hanno a disposizione una sola torcia per vedere al buio, quindi i numeri all'andata passano in coppia e poi uno di quelli che si trovano sull'altra sponda torna indietro con la torcia per permettere di passare anche agli altri con la luce.

Il costo del viaggio di andata è la somma dei due numeri. Il costo del viaggio di ritorno è il valore del numero che torna indietro.

Il costo totale di un attraversamento di tutti i numeri all'altro lato del ponte è semplicemente dato dalla somma dei costi dei singoli viaggi.

Si scriva una funzione Curry (non-deterministica) che dato un costo  $c$  restituisca una soluzione (tutte) a costo totale minore di  $c$ . Con soluzione si intende una lista che codifichi (in qualche modo) le configurazioni dei numeri rispetto al ponte.

**Si dia esplicitamente il tipo di ogni funzione definita.**