Prova Scritta di Linguaggi di Programmazione II

13/04/2007

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.

```
hanoi( z, _, _, _, M, M ).

hanoi( s(N), X, Y, Z, M, Q )
  :- hanoi( N, Z, Y, X, P, Q ), hanoi( N, X, Z, Y, M, [m(X,Y) | P] ).
```

hanoi/6:	try_me_else	$hanoi_1$		put_val	Y_1	A_1
	get_str	z/0	A_1	put_val	Y_2	A_2
	get_var	X_7	A_5	put_val	Y_4	A_3
	get_val	X_7	A_6	put_val	Y_3	A_4
	preceed			put_val	Y_5	A_5
$hanoi_1$:	trust_me			put_str	m/2	X_8
	alloc		6	set_val		Y_2
	get_str	s/1	A_1	set_val		Y_3
(N)	uni_var		Y_1	put_str	cons/2	A_6
(X)	get_var	Y_2	A_2	set_var		X_8
(Y)	get_var	Y_3	A_3	set_val		Y_6
(Z)	get_var	Y_4	A_4	call		hanoi/6
(M)	get_var	Y_5	A_5	dealloc		
(Q)	get_var	X_7	A_6			
	put_val	Y_1	A_1			
	put_val	Y_4	A_2			
	put_val	Y_3	A_3			
	put_val	Y_2	A_4			
(P)	put_var	Y_6	A_5			
	put_val	X_7	A_6			
	call		hanoi/6			

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

```
last( _ ++ [x] ) = x
[]      ++ ys = ys
(x:xs) ++ ys = x:(xs++ys)
```

	sx	dx	dup	el	col
R1	•	•		•	•
R2	•	•			•
R3	•	•			
TRS	sì	sì	no	sì	sì

Il TRS non è constructor based dato che ++ deve stare dentro $\mathcal{D}$ per R2 (ad esempio) ma è anche dentro il pattern di R1.

IL TRS non è ortogonale, visto che (ad esempio) R1 ed R2 si sovrappongono.

3. Sia dato il seguente programma Curry

```
data P = Z | S P

f v [] = v
f (Just Z) (x:xs) = Just x

g [] = Nothing
g (x:_) = Just x
```

Si stabilisca se è induttivamente sequenziale e si calcolino tutte le derivazioni di *needed* narrowing per il termine `f (g x) x where x free`.

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

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

```
sum 1 = 1
sum n = n + sum (n-1)
```

Il tipo di dato inferito nei due linguaggi è lo stesso? Ci si può aspettare di ottenere i medesimi risultati nei 2 linguaggi usando la funzione su dati di tipo `Int`?

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 scriva un predicato PROLOG `limitedVisit/4` che dato un Binary Search Tree e due valori x, y costruisce la lista (ordinata) degli elementi dell'albero compresi nell'intervallo di valori da x a y.

Si scriva il programma con variabili anonime ove possibile.

6. Si ha un grafo con archi orientati etichettati da un colore e i cui nodi sono identificati da un numero univoco e sono, a loro volta, etichettati con un colore (non necessariamente distinto). Si parte con 2 nodi "marcati" ed un nodo "goal". La marcatura può essere spostata lungo un arco se il nodo destinazione non è già marcato e se il colore dell'arco è lo stesso del nodo attualmente marcato. Bisogna trovare una sequenza di azioni che fanno arrivare una marcatura nel nodo goal.

Il grafo viene rappresentato mediante una lista di archi e una funzione che ad ogni identificativo di nodo ne associa il colore secondo le definizioni

```
data Graph = Graph [Arrow] Coloring
data Arrow = Arrow Node Node Color
type Node = Int
type Coloring = Node->Color
data Color = Red | Blue | Green | Magenta | ...
```

Si scriva una funzione Curry (non-deterministica) che preso un grafo e tre identificatori (distinti) di nodo fornisca una (ogni) rappresentazione di una (ogni) soluzione.

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

$\{hanoi(z, A, B, C, D, D),\ hanoi(s(z), A, B, C, [m(A, B)|D], D),$
$hanoi(s(s(z)), A, B, C, [m(A, C), m(A, B), m(C, B)|D], D),$
$hanoi(s(s(s(z))), A, B, C, [m(A, B), m(A, C), m(B, C), m(A, B), m(C, A), m(C, B), m(A, B)|D], D)\}$