Prova Scritta di Linguaggi di Programmazione II

07/09/2006

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

```
tls( 0, [1,0,0] ).
tls( N, [X,Y,Z] ) :-
  N > 0, K is N-1, tls( K, [T,U,V] ),
  sum( [T,T,U,U], X ), sum( [T,U,U,V], Y ), sum( [U,U,V,V], Z ).

sum( [], 0 ).
sum( [X|L], S ) :- sum( L, P ), S is X+P.
```

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

$$f(x, B(y), z) \to f(x, y, C(C(z)))$$
$$f(A(x), y, B(z)) \to f(z, y, C(y))$$
$$f(D(A(x), E), E, y) \to B(y)$$

3. Si calcoli il minimo punto fisso dell'operatore delle conseguenze immediate per le risposte calcolate (T_P^{ca}) del seguente programma:

```
q(h(X),X).
q(X,h(X)).
p(X,Y) :- p(h(k),h(Y)),q(X,Y).
p(X,Y) :- q(X,Y),q(Y,X).
```

$$\boxed{\{q(A, h(A)), q(h(A), A), p(A, h(A)), p(h(A), A), p(k, h(k)), p(h(h(k)), h(k))\}}$$

4. Sia dato il seguente programma Curry

```
data P = Z | S P
data T a = L a | N a (T a) (T a)

g Z     = L Z                    --R1
g (S m) = N (S m) (g m) (g m)--R2

f (L x)       = L x              --R3
f (N x l r) = N x (f r) (f l)--R4
```

Si calcolino tutte le derivazioni di *basic* narrowing di lunghezza 2 per il termine `f(g(x))`.

5. Si vuole risolvere il seguente problema: dati certi numeri positivi distinti ed un valore positivo v determinare se esiste un'espressione aritmetica che valuti a v in cui ogni numero compaia *esattamente* una volta. Ad esempio una soluzione per $[2, 3, 6, 8]$ e 24 è $(2 + 8) * 3 - 6$.

Codificando le espressioni aritmetiche mediante il tipo di dato seguente, si scriva una funzione Curry (non-deterministica) *solve* che data una lista di numeri e un valore restituisce una soluzione (se esiste).

```
data Exp = N Int | O Op Exp Exp
data Op = Add | Mul | Sub | Div
```

Si dia esplicitamente il tipo della funzione.

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

```
fact 0 = 1
fact n = n * fact (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 perchè ciò non sia possibile.

7. Si scriva un predicato PROLOG `diff2next/2` che, dato un albero binario di ricerca, costruisce un albero binario di ricerca (annotato) di coppie dove il primo elemento di ogni coppia è l'elemento dell'albero originale mentre il secondo elemento è la differenza rispetto al valore successivo (secondo l'ordinamento dei valori contenuti). Si utilizzi la costante `infty/0` per annotare il nodo di valore massimo. A titolo di esempio,

```
diff2next(tree(4,void,tree(7,tree(5,void,void),void)),T)
```

restituisce la soluzione

```
T = tree((4,1),void,tree((7,infty),tree((5,2),void,void),void)).
```

Si deve passare su ogni nodo dell'albero **una volta sola**.

Si scriva il programma con variabili anonime ove possibile.