

Linguaggi e Compilatori

Modulo Linguaggi - AA 2011-2012

1. Dare la semantica operativa in stile SOS per il seguente programma

```
x := 2 while ¬(x == 0) do y := y + x; x := x - 1;
```

nello stato iniziale $\sigma = [x \mapsto 2, y \mapsto 3]$.

2. Si rappresenti in ML (usando `datatype`) il tipo `nums` dei numeri interi e reali. Si scriva una funzione ML che data una lista di tipo `nums` restituisca la somma dei numeri interi nella lista.
3. Verificare che la funzione ML

```
fun comp(f,g,h) = f(g(h)) + 5;
```

ha tipo `('a -> int) -> ('b -> 'a) -> 'b -> int` (si assuma che il tipo di `+` sia `int -> int -> int`).

4. Si assuma la valutazione con scoping statico.

```
{
  int x = 2;
  void A(){
    int x = 10;
    void B(){
      int x = 300;
      print(x);
      x++;
    }
    print(x);
    B();
    print(x);
  }
  void B(){
    int x = -5;
    A();
    print(x);
  }
}
```

```
}
B();
print(x); //ultimo comando
}
```

Cosa viene stampato? Si risponda mostrando l'evoluzione completa dello stack dei record di attivazione sino all'esecuzione di `print(x); //ultimo comando`.

5. Sia dato il seguente frammento di codice e si assumi la valutazione con scoping statico.

```
{
  int x = 1; int y = 2;
  void C(value int a){
    int x = 3;
    void D(ref int z){
      z = z + 1;
      print(z, x, a);
    }
    D(x);
  }
  void B(){
    int y = 4;
    C(y);
  }
  B();
}
```

Cosa viene stampato? Si mostri l'esecuzione fornendo l'evoluzione dello stack dei record di attivazione sino al momento in cui viene effettuato il pop del RA di `D()`.

6. Si descriva il metodo di passaggio dei parametri per riferimento. Il linguaggio ANSI C ha il passaggio per riferimento?

7. Si descriva sinteticamente l'implementazione del passaggio dei parametri di tipo funzione (riferendosi allo stack dei record di attivazione).
8. Si descriva il problema dell' *upward funarg* e la sua implementazione.
9. Si descriva il metodo di passaggio dei parametri per **value result** e si mostri che tale metodo non è equivalente al passaggio per **riferimento**.
10. Si descriva l'algoritmo di garbage collection *mark and sweep*.
11. Definire una funzione **sottoliste** che data una lista $[a_0; a_1; \dots; a_n]$ restituisca la lista di sottoliste


```
[[ ]; [a0]; [a0; a1]; [a0; a1; a2]; ...; [a0; a1; ...; an]].
```

Soluzione

```
fun inc(x, []) = [] |
  inc(x, a::r) = (x::a)::inc(x,r);

fun sottoliste([]) = [[]]
  | sottoliste (x::r) = []::(inc(x,sub(r)));
```

12. Si definisca un intero n come una lista contenente n volte il numero uno come elemento. Ad esempio 0 è rappresentato da $[\]$, 1 è rappresentato da $[1]$, 2 è rappresentato da $[1; 1]$ e così via. Si definiscano per tali numeri le operazioni somma **sm** e prodotto **pr** tramite opportune funzioni ML.

Soluzione

```
fun sm([],y) = y
  | sm(1::r, y) = 1:: sm(r,y);

fun prod([],y) = [] |
  prod(1::r,y) = sm(y,prod(r,y));
```

13. Si definisca una funzione **merge** che date due liste di interi ordinate restituisca la lista ordinata contenente l'unione delle liste date.

Soluzione

```
fun merge (xs,nil) = xs
  | merge (nil,ys) = ys
  | merge (x::xs,y::ys) = if x<y
  then x :: merge(xs,y::ys)
  else y :: merge(x::xs,ys);
```

14. Si definisca una funzione che data una lista di interi applichi la funzione 91 di McCarthy ad ogni elemento della lista.

Soluzione

```
fun map (f,nil) = nil
  | map (f,h::t) = f(h) :: map (f,t);

fun mc91 (n) = if n>100 then n-10
  else mc91 (mc91 (n+11));
```

15. Che tipo di passaggio degli argomenti hanno i linguaggi ML, C e Java? In Java, cosa accade quando viene passato un array ad un metodo?