

Esercitazione di Linguaggi e Compilatori

Modulo Linguaggi - - AA 2011-2012

25 Gennaio 2012

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

```
while (x > 0) do y := 2 * y; 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 f(g, h) = g(h) + 2;
```

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

4. Si consideri il seguente frammento di codice e si assuma la valutazione con scoping statico.

```
{ int x = 5;
  int y = 4;
  void C(){
    int x = 3;
    void D(){int x = 2; }
    D();
  }
  void B(){
    int x = 4;
    int z = 3;
    C();
  }
  B();
}
```

Si rappresenti graficamente lo stack dei record di attivazione dopo che il controllo è entrato nella funzione `D()`.

5. Si consideri il seguente frammento di codice:

```
int x = 1;
function g(z) = x+z;
function f(y) = {
  int x = y+1;
  return g(y*x)
};
f(3);
```

Si rappresenti graficamente lo stack dei record di attivazione e se ne descriva l'evoluzione in caso di scoping statico e in caso di scoping dinamico.

6. Si descriva il metodo di passaggio dei parametri per **value result** e si mostri che tale metodo non è equivalente al passaggio per **riferimento**.
7. Si descriva l'algoritmo di garbage collection *mark and sweep*.