

Corso di Laurea in Informatica
Linguaggi e Compilatori: Linguaggi

Esame del 18 Giugno 2015

Si ricorda che ogni risposta va giustificata ed ogni concetto spiegato nel corso che viene citato va definito. Per ogni esercizio si indica tra parentesi il valore in 30-esimi dell'esercizio.

1. (4) Definire cosa è un interprete: Dare definizione semantica e descrivere la struttura.
2. (4) Dimostrare per induzione che $\forall n \in \mathbb{N}. n > 1$ si ha che $n^3 + 5n$ è un multiplo di 6, ricordando che $n + n^2$ è pari.
3. (6) Descrivere intuitivamente cosa è una dichiarazione e descrivere i vari tipi di composizione di dichiarazioni in IMP. Definire formalmente le dichiarazioni di IMP e dare semantica operativa statica e dinamica di una composizione di dichiarazioni a scelta.
4. (5) Si dica per ognuno dei seguenti metodi di passaggio di parametri (definiti opportunamente), quali sono i valori di *value* e *list* dopo le tre chiamate di *scambia*?

- Passaggio per valore;
- Passaggio per riferimento;
- Passaggio per valore-risultato.

```
void scambia (int a, int b){  
    int temp = a;  
    a = b;  
    b = temp; }  
void main(){  
    int value = 2, list[5] = {1, 3, 5, 7, 9};  
    scambia(value, list[0]);  
    scambia(list[0], list[1]);  
    scambia(value, list[value]); }
```

5. (4) Si dica cosa viene stampato dal seguente frammento di pseudo-codice in caso di: (1) scoping statico (deep-binding); (2) scoping dinamico con deep-binding; (3) scoping dinamico con shallow-binding.

```
int x = 3; int y = 2;  
int fun1 (int in){  
    return x + y + in;  
}  
void fun2 (int h(int b)){  
    int x = 4; int y = 4;  
    return h(5) + x;  
}  
...  
{int x = 6; y = 3;  
 int z = fun2(fun1); }
```

6. (3) Descrivere il paradigma funzionale, descrivendo le caratteristiche principali che lo distinguono da altri paradigmi di programmazione.

7. (6) Dire quale errore determina la semantica statica del codice a destra fornendo le derivazioni. Ignorando l'errore rilevato staticamente, quale modifica nella strategia di valutazione delle regole di semantica dinamica permette l'esecuzione senza errori del programma a tempo di esecuzione?

```
var  $x$  : bool = tt  
const  $y$  : bool = tt  
procedure  $p(\bullet)$   $x := z$  and  $y$   
const  $y$  : bool = ff  
 $p(\bullet)$ 
```