

Corso di Laurea in Informatica
Linguaggi e Compilatori: Linguaggi

Esame del 5 Febbraio 2014

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 intuitivamente e formalmente il concetto di specializzazione e specializzatore.
2. (4) Dimostrare per induzione che $\forall n \in \mathbb{N}. n > 0$ si ha che $n + n^2$ è un numero pari.
3. (4) Descrivere intuitivamente cosa è una dichiarazione. Definire formalmente le dichiarazioni di IMP e dare semantica operativa statica e dinamica ad una composizione di dichiarazioni a scelta.
4. (6) Si consideri lo schema del seguente codice, nel quale vi sono due buchi indicati rispettivamente con (*) e (**). Si dia il codice da inserire al posto di (*) e (**) in modo tale che:

- (a) Se il linguaggio usato adotta lo scope statico, le due chiamate alla procedura foo assegnino a x lo stesso valore;
- (b) Se il linguaggio usato adotta lo scope dinamico, le due chiamate alla procedura foo assegnino a x valori diversi.

```
{ int i
(*)
for (int i = 0; i <= 1; i++) {
    int x;
    (**);
    x=foo();
}
```

5. (6) Definire il concetto di ricorsione. Si consideri il seguente programma ricorsivo e si descriva che cosa calcola. Trasformare quindi tale programma in un programma tail recursive.

```
int function (lista list){
if (length(list) = 0) then return 0;
else return(car(list) + function(cdr(list)));
}
```

NOTA: Si supponga che data una lista $list$, $length(list)$ restituisca la lunghezza della lista, $car(list)$ restituisca il primo elemento della lista e $cdr(list)$ restituisca tutta la lista tranne il primo elemento. Mostrare poi il funzionamento dell'algoritmo tail recursive sulla lista (3, 6, 9).

6. (2) Descrivere almeno un aspetto caratterizzante per ognuno dei seguenti paradigmi: imperativo, logico e funzionale.
7. (6) Modificare la semantica *dinamica* di IMP in modo da poter passare le procedure come parametri di altre procedure implementando il deep binding in caso

di scoping dinamico. Cosa cambia in caso di shallow binding? Siano le seguenti le modifiche necessarie delle definizioni e della semantica statica:

Lista formali: $form ::= \dots | \mathbf{proc} \ p : aetproc, form$
 Valori attuali: $ak ::= \dots | \lambda \ form.c, ak$
 Tipi attuali: $aet ::= \dots | aetproc, aet$

Estrazione tipi formali: $\mathcal{T}(\mathbf{proc} \ p : aetproc, form) = aetproc, \mathcal{T}(form)$

Identificatori liberi: $FI(\mathbf{proc} \ p : aetproc, form) = \emptyset$
 Identificatori definiti: $DI(\mathbf{proc} \ p : aetproc, form) = \{p\} \cup DI(form)$

Semantica statica:
$$\frac{form : \Delta_0, \Delta_0 : I_0, p \notin I_0}{(\mathbf{proc} \ p : aetproc, form) : \Delta_0[p = aetproc]}$$