

Corso di Laurea in Informatica

Linguaggi

Esame del 5 Luglio 2018

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. (3) Definire cosa è un interprete: Dare definizione semantica e definire la struttura.
2. (2) Dimostrare per induzione che $n + n^2$ sempre un numero pari.
3. (4) Descrivere intuitivamente cosa è un comando ed in particolare un comando condizionale. Definire formalmente i comandi di IMP e dare semantica operativa statica e dinamica al comando condizionale **if**.

4. (7) Si consideri il programma sulla destra. Si dica cosa viene calcolato ad ogni assegnamento in caso di scoping statico e in caso di scoping dinamico. In particolare, nello scoping statico si mostri l'evoluzione dello stack dei RdA mostrando come vengono calcolati i link statici degli RdA e come avviene la risoluzione dei riferimenti non locali. Nello scoping dinamico si mostri l'evoluzione della tabella centrale dei riferimenti, specificando sempre come vengono risolti i riferimenti non locali.

```
{int x := 1;
void A(){
    int x := x + 1;}
void B(){
    int x;
    void C(int y){
        int x := y + 2;
        A();}
    x:=0;
    A();
    C(3);}
x := 10; B(); }
```

5. (4) Definire cosa sono le politiche di binding. Spiegare in particolare la differenza tra shallow binding e deep binding, facendo attenzione a come si combinano con lo scoping statico e/o dinamico?
6. (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.

```
lista function (lista list){
    if (length(list) = 0) then return ();
    else return concat(function(cdr(list)), car(list) + 1);
}
```

NOTA: Si supponga che data una lista *list*, *length(list)* restituisca la lunghezza della lista, *car(list)* restituisca il primo elemento della lista, *cdr(list)* restituisca tutta la lista tranne il primo elemento, *concat* concateni due liste (se un elemento non è una lista, *concat* lo converte prima in lista). *()* è la lista vuota.

Mostrare sia il funzionamento dell'algoritmo ricorsivo dato, sia di quello tail recursive sulla lista (3, 6, 9).

7. (4) Sia AExp la categoria sintattica delle espressioni aritmetiche e Ids identifica una lista di identificatori. Descrivere la semantica statica e dinamica in \mathbb{K} della dichiarazione multipla di variabili, aggiungendo eventuali attributi di *strictness*.

```
syntax Ids ::= List {Id, ", " }
syntax Stmt ::= ... | int Ids = 0 ;
```

La configurazione la seguente:

```
<T>
    <k> $PGM: Stmt </k>
    <env> .Map </env>
    <store> .Map </store>
    <nextLoc> 0 </nextLoc>
</T>
```