

**Prova Scritta di Linguaggi di programmazione**  
**Corso di Linguaggi di programmazione I**  
**23/06/2004**

### **Prolog**

Scrivere un predicato `sumEvenOfTree/2` in SICSTUS PROLOG per calcolare, a partire da un albero binario di numeri interi, la somma dei valori **pari** presenti nell'albero.

Ad esempio `sumEvenOfTree(tree(4,void,tree(3,void,void)),4)` è vero.

Si assuma di chiamare `sumEvenOfTree/2` con il primo argomento sempre istanziato ad una albero binario di interi. Si scriva il programma con variabili anonime ove possibile.

### **Unificazione**

Si calcoli un unificatore più generale di  $f(g(x), x, y)$  e  $f(t, g(z), h(t))$

### **Semantica**

Si consideri il seguente frammento di programma espresso in un linguaggio imperativo C-like con scoping statico:

```
{ int x:=5;
  proc Q(ref int y; value int w)
  {
    x := 4+y;
    y := w+y;
  }
  Q(x, x);
}
```

si fornisca la semantica denotazionale del frammento, mostrando l'ambiente e la memoria nel momento in cui termina la valutazione della chiamata di procedura Q.

**Implementazione** Con riferimento alla tecnica di implementazione con "catena statica", per l'esercizio precedente si mostri (schematicamente) con dei diagrammi la situazione sullo stack di sistema (per i valori) degli identificatori al momento in cui inizia l'esecuzione di Q.

NOME E COGNOME:	Numero di Matricola:

## Prolog

```
sumEvenOfTree(void,0)
sumEvenOfTree(tree(X,L,R),S):-
    val(X,Y),
    sumEvenOfTree(L,LS),
    sumEvenOfTree(R,RS),
    S is Y+LS+RS.
```

```
val(N,N) :- 0 is N mod 2.
val(N,0) :- 1 is N mod 2.
```

## Unificazione

```
T = g(g(Z)),
X = g(Z),
Y = h(g(g(Z)))
```

# Linguaggi di Programmazione I

Esame del 23 Giugno 2004

cognome e nome

## Macchina astratta Pascal

Con riferimento al seguente programma in Pascal, si rappresenti in questo foglio il P-code relativo al corpo della procedura *ack*, cioè il P-code corrispondente alla sezione evidenziata dal riquadro.

```
program esercizio;    ...    ...    ...
```

```
  procedure ack( x, y: integer; var z: integer );
```

```
    var t: integer;
```

```
  begin
```

```
    if x = 0 then
```

```
      z := y + 1
```

```
    else if y = 0 then
```

```
      ack( x-1, 1, z )
```

```
    else begin
```

```
      ack( x, y-1, t );  ack( x-1, t, z )
```

```
    end
```

```
  end;
```

```
begin    ...    ...    ...  end.
```