

# Linguaggi di Programmazione

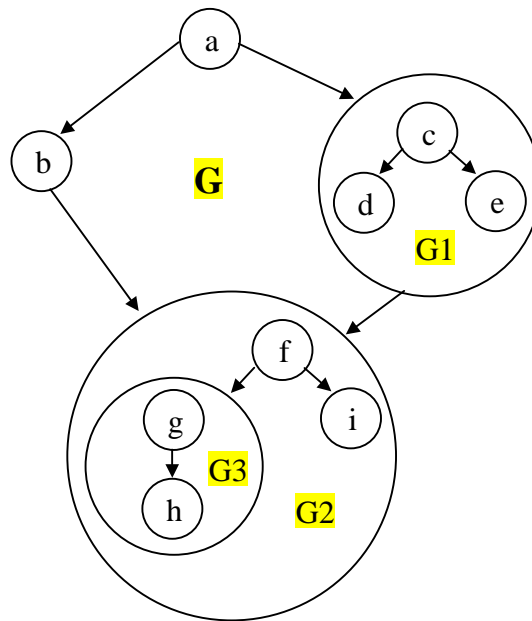
Nome e Cognome	
Corso di laurea	
Telefono	
Email	

1. Specificare la grammatica EBNF di un linguaggio in cui ogni frase specifica un grafo, come nel seguente esempio:

```

graph G is
  nodes a,
    b,
    graph G1 is
      nodes c, d, e
      arcs (c,d), (c,e)
    end G1,
    graph G2 is
      nodes f,
        i,
        graph G3 is
          nodes g, h
          arcs (g,h)
        end G3,
      arcs (f,G3), (f,i)
    end G2
  arcs (a,b), (a,G1), (b,G2), (G1,G2)
end G

```



Un grafo è definito da un nome, una lista di nodi ed una lista di archi. Un nodo può essere a sua volta un grafo, senza limiti di profondità. Ogni arco è definito dalla coppia di identificatori dei nodi congiunti dall'arco. (Le regole di punteggiatura si evincono dall'esempio.)

2. Specificare la semantica operativa della seguente istruzione *SQL*-like di aggiornamento di una tabella:

(esempio di frase)

```

update T
set a = E
where p

```

```

update Impiegati
set stipendio = stipendio + 0.2 * stipendio
where categoria <= 3

```

in cui  $T$  è il nome della tabella operando,  $a$  è il nome dell'attributo da aggiornare,  $E$  è l'espressione di aggiornamento e  $p$  il predicato di selezione delle righe da aggiornare. Si assume che il linguaggio di specifica della semantica sia imperativo.

3. Dopo aver specificato in *Scheme* la funzione di Fibonacci (**fib** *n*), specificare la funzione (**listafib** *n*) che, ricevendo in ingresso un intero  $n \geq 1$ , computa la lista dei numeri di Fibonacci, da 1 ad *n*, come nei seguenti esempi:

<b>n</b>	( <b>listafib</b> <i>n</i> )
1	(0)
2	(0 1)
3	(0 1 1)
4	(0 1 1 2)
5	(0 1 1 2 3)
10	(0 1 1 2 3 5 8 13 21 34)

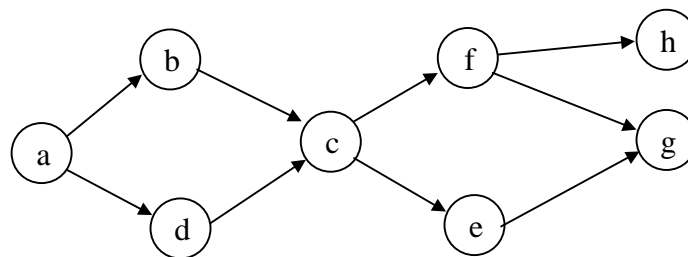
(Si ricorda che, per  $n \geq 3$ , il numero di Fibonacci relativo ad *n* è la somma dei numeri di Fibonacci relativi ad *n-1* ed *n-2*.)

4. Definire nel linguaggio *Haskell*, mediante la notazione di pattern-matching, la funzione **dispari** (protocollo incluso) che, avente in ingresso un intero  $n \geq 0$ , computa la lista dei primi *n* numeri dispari, come nei seguenti esempi:

<b>n</b>	<b>dispari</b> <i>n</i>
0	[ ]
1	[1]
2	[1,3]
3	[1,3,5]
10	[1,3,5,7,9,11,13,15,17,19]

5. E' data una base di fatti *Prolog* relativa ad un grafo aciclico diretto, come nel seguente esempio,

```
arco(a,b).
arco(a,d).
arco(b,c).
arco(d,c).
arco(c,e).
arco(c,f).
arco(e,g).
arco(f,g).
arco(f,h).
```



Si chiede di specificare il predicato **path**(*N1*,*N2*,*P*), che risulta vero qualora *P* sia la lista di nodi generata da un cammino dal nodo *N1* al nodo *N2*, ad esempio **path**(*a*,*f*,*[a,d,c,f]*). È possibile avere anche cammini di un solo nodo, ad esempio **path**(*a*,*a*,*[a]*).

6. Illustrare i principi della semantica denotazionale con l'ausilio di un esempio.