

Linguaggi di Programmazione

<i>Nome e Cognome</i>	
<i>Matricola</i>	
<i>Anno di corso</i>	
<i>Telefono</i>	

1. Specificare la grammatica EBNF di un linguaggio per la manipolazione di tabelle, in cui ogni frase comprende una o più sezioni. Esistono due tipi di sezioni: **data** e **query**. Nella sezione **data** vengono definiti gli schemi di una o più tabelle. Nella sezione **query** vengono formulate una o più interrogazioni sulle tabelle utilizzando (in modo ortogonale) le operazioni relazionali di proiezione (**project**) e selezione (**select**). Lo schema di una tabella è specificato dal nome della tabella e dalla lista di attributi definiti su corrispondenti domini (**int**, **string**, **bool**). Nella operazione **project** si specifica la lista degli attributi di proiezione. Nella operazione **select** si specifica un predicato di selezione composto dalla congiunzione di una o più confronti tra un attributo e una costante. Ecco una frase contenente due sezioni:

```

data
  R: (a: int, b: string, c: bool, d: int);
  S: (alfa: bool, beta24: string);
  T: (X: string);
end

query
  project [a, b] R;
  select [ alfa = false ] S;
  select [ X = "luna" ] T;
  project [ b ] select [ a = 5 ] R;
  project [a, d ]
    select [ a = 3 and b = "stella" and c = false ]
      select [ d = 125 ] R;
end

```

2. Assumendo la seguente tabella di operatori (in cui ogni operatore binario valuta prima l'operando di destra e poi quello di sinistra, e la valutazione delle espressioni logiche è in corto circuito),

<i>Operatori</i>	<i>Associatività</i>
^	destra
+, -, *, /	sinistra
<, >	nonassoc
and, or	sinistra

specificare la semantica operativa del seguente assegnamento della variabile logica *v*:

```
v := a + b * c > d ^ e ^ f + 1 or g < h/i
```

NB: Il linguaggio di specifica operativa contempla le seguenti limitazioni:

- Non contiene gli operatori logici **and**, **or**;
- Contiene gli operatori **^**, **+**, **-**, *****, **/**, **<**, **>**, che però non possono essere applicati al risultato di altre operazioni (quindi è necessario l'introduzione di opportuni temporanei per i risultati intermedi ...).

3. Definire nel linguaggio funzionale *Scheme* la funzione **reverse**, che computa l'inverso di una lista **A** di atomi in ingresso. Ecco alcuni esempi:

A	reverse
()	()
(a)	(a)
(a b c d)	(d c b a)

4. Definire nel linguaggio *Haskell* una struttura tabellare

```
Prodotti(Codice, Nome, Prezzo)
```

in cui ogni riga rappresenta un prodotto. Quindi, codificare la funzione `selezione` che, ricevendo in ingresso due interi, `min` e `max`, computa la lista dei nomi dei prodotti il cui prezzo è compreso tra `min` e `max`.

5. Discutere i fattori di costo di un linguaggio di programmazione, correlandoli alle caratteristiche dei linguaggi che possono influenzarli.
6. Discutere i principi della semantica assiomatica, con particolare riferimento ai costrutti di assegnamento e sequenza.