

# Linguaggi di Programmazione (6 CFU)

Cognome e nome	
Email	

- 1.** Specificare la definizione regolare relativa ad una lista (anche vuota) di numeri complessi, come nel seguente esempio:

```
[ ( 1 , 24.66 ) , ( 0.12 , +3 ) , ( -1.845 , -1.23E20 ) , ( 33E4 , 26.80E-3 ) ]
```

sulla base dei seguenti vincoli lessicali:

- Un numero complesso è rappresentato da una coppia (parte reale, parte immaginaria);
- Ognuna delle due parti è rappresentata da un numero, eventualmente con segno, avente una parte intera, opzionalmente una parte decimale e, infine, opzionalmente una parte esponenziale (rappresentata da E seguita da un intero, eventualmente con segno);
- La parte intera non contiene zeri non significativi;
- Il linguaggio è libero dal formato: ogni elemento lessicale può essere separato dagli altri mediante spazi bianchi e/o tab e/o newline.

- 2.** Specificare la grammatica BNF del linguaggio specificato al punto **1**, assumendo che gli elementi lessicali siano: **0 1 2 3 4 5 6 7 8 9 [ ] ( ) , + - E .**

- 3.** Specificare la semantica operativa della seguente espressione relazionale:

```
join [ x = y and z > 10 ] R S
```

(in cui R ed S sono gli operandi dell'operazione di join) sulla base dei seguenti requisiti:

- L'operatore logico di congiunzione è valutato in corto circuito;
- Gli attributi delle tabelle operando sono interi o stringhe;
- Nei confronti all'interno del predicato, nel caso di identificatore, si assume che il primo operando appartenga alla tabella R ed il secondo alla tabella S;
- Non sono ammesse operazioni miste;
- Nel caso di errore semantico, si invoca la funzione **errore**, il cui argomento è un messaggio (pertinente) di errore, la quale termina l'esecuzione;
- Sono disponibili le seguenti funzioni ausiliarie:
  - **schema**(T): restituisce la lista delle coppie (attributo, tipo) della tabella T,
  - **attributes**(T): restituisce la lista degli attributi della tabella T,
  - **type**(attr, schema): restituisce il tipo dell'attributo attr nello schema;
- È possibile utilizzare gli operatori insiemistici di appartenenza, unione ed intersezione applicati a liste.

4. È dato il seguente frammento di grammatica BNF relativo alla specifica di una istruzione **case** in un linguaggio imperativo:

```
case-stat → case expr of case-list default end
case-list → case case-list | case
case → const : stat-list
default → otherwise stat-list |  $\epsilon$ 
```

Si chiede di specificare la semantica denotazionale del corrispondente frammento di linguaggio assumendo la disponibilità delle seguenti funzioni ausiliarie (di cui non è richiesta la specifica):

- $M_{\text{expr}}(\text{expr}, s)$ : restituisce il valore di *expr* (eventualmente ERROR) allo stato *s*;
- $M_{\text{stat-list}}(\text{stat-list}, s)$ : restituisce il nuovo stato (eventualmente ERROR) dopo l'esecuzione delle istruzioni in *stat-list* allo stato *s*;
- $M_{\text{const}}(\text{const})$ : restituisce il valore della costante **const**.

5. Specificare nel linguaggio *Scheme* la funzione **funpair**, avente in ingresso una funzione binaria *f* e due liste, *list1* e *list2*, la quale restituisce la lista dei valori risultanti dall'applicazione di *f* agli elementi di *list1* e *list2* che si trovano nella stessa posizione nella rispettiva lista, come nel seguente esempio:

```
(funpair + '(1 2 3) '(4 5 6 7 8)) = (5 7 9)
```

6. È data la seguente dichiarazione nel linguaggio *Haskell*, per la rappresentazione di alberi:

```
data Albero = Nodo Float [Albero]
```

Si chiede di codificare la funzione **media** (protocollo incluso), la quale riceve in ingresso un albero e restituisce la media dei numeri nell'albero.

7. Specificare nel linguaggio *Prolog* il predicato **posizioni** (*N1, N2, N3, N4, N5, P1, P2, P3, P4, P5*) i cui argomenti sono numeri interi (diversi fra loro), il quale risulta vero quando ogni *Pi*, *i* ∈ [1..5], rappresenta l'*i*-esimo numero in un ordinamento ascendente dei numeri *N1 ... N5*, come nel seguente esempio:

```
?- posizioni(34,12,1,78,0,P1,P2,P3,P4,P5).
P1 = 0,
P2 = 1,
P3 = 12,
P4 = 34,
P5 = 78
```

8. Illustrare le scelte progettuali nel passaggio di parametri che sono sottoprogrammi.