

Linguaggi di Programmazione

Nome e Cognome	
Matricola	
Anno di corso	
Telefono	

1. Specificare la grammatica EBNF di un linguaggio in cui ogni frase corrisponde ad un modulo SQL-like. Tale modulo contiene una lista (non vuota) di istruzioni. Ogni istruzione può essere una creazione di tabella o una interrogazione, come nel seguente esempio:

```

create table R
(
    A    character(20),
    B    numeric(5),
    C    character(15)
);

create table S
(
    D    character(20),
    E    numeric(6)
);

select A, E
from R, S
where A = D and B > 100 or E < 50;

select *
from R, S
where A = 'alfa' and B in (select E
                           from S
                           where D = A);

```

I domini degli attributi delle tabelle sono **character** e **numeric**, entrambi qualificati con una dimensione. Ogni interrogazione ha la forma *select-from-where*, in cui le clausole **select** e **from** sono obbligatorie. La clausola **select** specifica una lista di attributi o il metacarattere '*'. La clausola **from** specifica una lista di tabelle. La clausola **where** specifica il predicato di selezione che coinvolge gli operatori logici **and** ed **or**, i quali sono applicati ad operazioni di confronto (=, <>, >, >=, <, <=) o di appartenenza (**in**). Il secondo operando dell'operatore di appartenenza (necessariamente fra parentesi) è a sua volta (ricorsivamente) una espressione *select-from-where*.

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

Nome	Operatore	Tipo	Associatività
selezione, proiezione	σ, π	unario	destra
intersezione	\cap	binario	sinistra
unione, differenza	$\cup, -$	binario	sinistra
inclusione	\supset, \subset	binario	nonassoc
coniunzione, disgiunzione	\wedge, \vee	binario	sinistra

specificare la semantica operativa del seguente assegnamento della variabile complessa R:

$$R := \sigma \pi \sigma A \cup \sigma B - C \supset D \cup E \cap F \vee G \subset \pi \sigma H$$

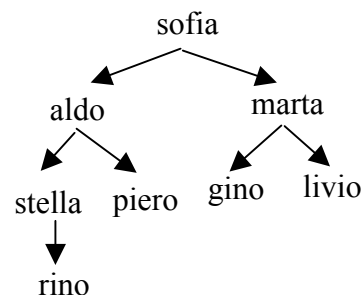
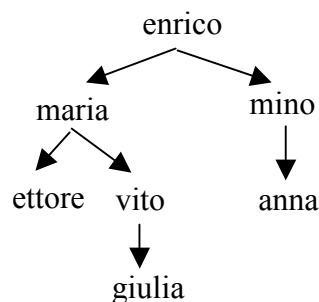
NB: Il linguaggio di specifica operativa contempla le seguenti limitazioni:

- Non contiene gli operatori logici \wedge , \vee ;
- Contiene tutti gli operatori insiemistici, che però possono essere applicati solo a variabili.

- Definire nel linguaggio *Smalltalk* il metodo `greaterThan: num`, il quale, applicato ad un oggetto vettore di interi, restituisce il sotto-vettore di interi che hanno un valore maggiore di num.

- Data una base di fatti *Prolog* relativa ad alberi genealogici, come nel seguente esempio:

```
genitore(enrico, maria).
genitore(enrico, mino).
genitore(maria, ettore).
...
genitore(sofia, aldo).
genitore(sofia, marta).
genitore(aldo, stella).
...
```



definire il predicato `parenti(X, Y)`, che stabilisce se X ed Y appartengono allo stesso albero genealogico.

- Discutere la correlazione che sussiste tra i linguaggi di programmazione e l'affidabilità del software.
- Descrivere il concetto di coroutine.