

Strutture dati in PROLOG

Esercitazione 2

Sommario

- Soluzione Esercizi
- Predicato is
- Liste
- Definizione dei termini
- Modello operativo (completo)
- Numeri naturali
- Esercizi

Soluzioni: Esercizio 1

```
fratello(X,Y) :-  
    figlio(X,Z), figlio(Y,Z), X \= Y.
```

```
cugino(C1,C2) :-  
    figlio(C1,G1),  
    figlio(C2,G2),  
    fratello(G1,G2).
```

\=

ha successo se i suoi argomenti sono diversi.

Esiste anche = ma non si usa !!!!

Soluzioni: Esercizio 3

- 1.
- 2.
3. Costruire l'albero di ricerca per ?- discendente(enzo, andrea) .
e verificare le differenze con ?- discendente2(enzo, andrea) .

```
/* Esercizio 3 */
```

```
discendente(X,Y):-figlio(X,Y). % 1
```

```
discendente(X,Y):-figlio(Z,Y),discendente(X,Z). % 2
```

```
discendente2(X,Y):-figlio(Z,Y),discendente2(X,Z). % 2
```

```
discendente2(X,Y):-figlio(X,Y). % 1
```

Prolog: predicato is

$A \text{ is } B$ è un predicato di sistema, vero quando la *valutazione* dell'espressione B restituisce un valore che viene **assegnato** alla variabile A .

La valutazione di B viene fatta utilizzando gli operatori di sistema.

I predicati definiti mediante is non sono invertibili: ponendo all'interprete la richiesta

?5 is X+Y.

non si ottengono le possibili assegnazioni ad X e Y che rendono vero il predicato.

?X is 3+4.

ha successo e ritorna $X=7$.

Prolog: programmi con `is`

```
factorial(0,1).  
factorial(Y,X):-  
    Y1 is Y-1,  
    factorial (Y1,X1),  
    X is Y*X1.
```

```
lunghezza([],0).  
lunghezza([_X|Xs],N) :- lunghezza(Xs,N1), N is N1 + 1.
```

Esercizi

Modellare in Prolog la funzione `erdosnum()`, date le seguenti specifiche:

1. Erdos ha `erdosnum=0`
2. X ha come `erdosnum` $1 +$ il minimo tra gli `erdosnum` delle persone con cui ha scritto almeno una pubblicazione.

Scrivere un programma Prolog `erdosnum(X,N)` in cui: X è il soggetto di cui si vuole conoscere il numero di Erdos; N è la distanza di Erdos tra i due.

Scrivere un programma Prolog `erdosnum(X,Y,N)` in cui: X è il soggetto di cui si vuole conoscere la distanza da Y; Y è il "mitico" ricercatore; N è la distanza tra i due.

Soluzioni: Numero di Erdos

```
pubblicazione(erdos,brachman).  
pubblicazione(erdos,einstein).  
pubblicazione(erdos,konolige).  
pubblicazione(brachman,nardi).  
pubblicazione(nardi,settembre).  
%pubblicazione(erdos,settembre).  
pubblicazione(brachman,erdos).  
pubblicazione(einstein,erdos).  
pubblicazione(konolige,erdos).  
pubblicazione(nardi,brachman).  
pubblicazione(settembre,nardi).  
%pubblicazione(settembre,erdos).  
%pubblicazione(X,Y):-pubblicazione(Y,X).
```


Soluzioni: Numero di Erdos (continua)

```
erdosnum(X,Limite):-erdosnumcicli(X,0,Limite).
```

```
erdosnumcicli(X,Limite,Limite):- erdosN(X,Limite).
```

```
erdosnumcicli(X,Limite,N1):- Limite < 10, L is Limite+1,  
                             erdosnumcicli(X,L,N1).
```

```
erdosN(erdos,0).
```

```
erdosN(X,Lung):- Lung>0,  
                 pubblicazione(X,Z), L is Lung-1,  
                 erdosN(Z,L).
```

Le liste

- `[a,b,c,d]` rappresenta una lista di quattro elementi;
- `[a | X]` rappresenta una lista il cui primo elemento è `a` e il resto della lista è denotato dalla variabile `X`;
- `[Y | X]` rappresenta una lista il cui primo carattere è denotato dalla variabile `Y` e il resto della lista è denotato dalla variabile `X`.

`[a | X]` equivale a `cons(a,X)`

`appartiene(X, [X|Xs]).`

`appartiene(X, [Y|Ys]) :- appartiene(X,Ys).`

Programmi sulle liste

```
append1([],Ys,Ys).
```

```
append1([X|Xs],Ys,[X|Zs]) :- append1(Xs,Ys,Zs).
```

```
prefisso([],_Ys).
```

```
prefisso([X|Xs],[X|Ys]) :- prefisso(Xs,Ys).
```

```
lunghezza([],0).
```

```
lunghezza([_X|Xs],s(N)) :- lunghezza(Xs,N).
```

```
reverse1([],[]).
```

```
reverse1([X|Xs],Zs) :- reverse1(Xs,Ys), append1(Ys,[X],Zs).
```

Termini

L'insieme Term dei *termini* è l'insieme induttivo definito come segue:

1. Ogni simbolo di costante è un termine;
2. Ogni simbolo di variabile è un termine;
3. Se $t_1 \dots t_n$ sono termini e f è un simbolo di funzione n -aria, $f(t_1, \dots, t_n)$ è un termine (detto *termine funzionale*).

Esempi: x , c , $f(x, y + c), \dots$

La definizione degli atomi e delle clausole rimane invariata.

Ripasso Unificazione

Verificare se le seguenti coppie di espressioni unificano. In caso positivo specificare l'unificatore altrimenti indicare altrimenti indicare una modifica del secondo termine che lo renda unificabile con il primo. X, Y, Z sono variabili e a, b sono costanti.

$$(1u) \ f(\text{cons}(\text{car}(X), \text{cdr}(Y)), Z, X) \text{ e } f(Z, Z, \text{cons}(\text{car}(X), \text{cdr}(a)))$$

$$(2u) \ f(g(x, a), g(b, a)) \text{ e } f(y, y)$$

$$(3u) \ P(g(x, a), f(b, a)) \text{ e } P(g(f(b, y), y), f(z, y))$$

$$(4u) \ P([X | [a, b]]) \text{ e } P([a | [a | [Xs]]])$$

Modello operativo del PROLOG

- interprete astratto (invariato)
- unificazione (esteso)
- ricerca delle soluzioni (invariato)

Sostituzioni

Una *sostituzione* è una funzione dall'insieme delle variabili Var all'insieme dei termini $Term$, cioè $\sigma : Var \mapsto Term$.

Dato un termine t , $t\sigma$ è definito ricorsivamente come segue:

- se c è un simbolo di costante, $c\sigma = c$;
- se x è un simbolo di variabile, $x\sigma = \sigma(x)$;
- se f è un simbolo di funzione di arità n , allora $f(t_1, \dots, t_n)\sigma = f(t_1\sigma, \dots, t_n\sigma)$.

Se t è un termine e σ è una sostituzione, allora $t\sigma$ è un termine.

Un programma per l'orario

```
corso(ia,orario(ma,10,12),docente(nardi,d),aula(spv,1)).
corso(ia,orario(me,10,12),docente(nardi,d),aula(spv,1)).
corso(ia,orario(ve,10,12),docente(nardi,d),aula(spv,1)).
...
insegna(Ins,Corso) :- corso(Corso,Orario,Ins,Aula).
durata(Corso,Lung) :-
    corso(Corso,orario(Giorno,Inizio,Fine),Ins,Aula),
    plus(Inizio,Lung,Fine).
haLezione(Ins,Giorno) :-
    corso(Cors,orario(Giorno,Inizio,Fine),Ins,Aula).
occupata(Aula,Giorno,Ora) :-
    corso(Corso,orario(Giorno,Inizio,Fine),Ins,Aula),
    Inizio =< Ora, Ora =< Fine.
```


Numeri naturali

```
natural_number(0).
```

```
natural_number(s(X)) :- natural_number(X).
```

```
plus1(0,X,X) :- natural_number(X).
```

```
plus1(s(X),Y,s(Z)) :- plus1(X,Y,Z).
```

```
lesseq1(0,X) :- natural_number(X).
```

```
lesseq1(s(X),s(Y)) :- lesseq1(X,Y).
```

```
lunghezza([],0).
```

```
lunghezza([_X|Xs],s(N)) :- lunghezza(Xs,N).
```

Esercizio sulle liste:ordinamento

```
sort1(Xs,Ys) :- permutation(Xs,Ys), ordered(Ys).
```

```
permutation(Xs,[Z|Zs]) :- select(Z,Xs,Ys),  
                           permutation(Ys,Zs).
```

```
permutation([],[]).
```

```
ordered([]).
```

```
ordered([_X]).
```

```
ordered([X,Y|Ys]) :- X =< Y, ordered([Y|Ys]).
```

```
select(X,[X|Xs],Xs).
```

```
select(X,[Y|Ys],[Y|Zs]):-select(X,Ys,Zs).
```

Esercizio in classe

Si considerino dei termini Prolog che rappresentano le informazioni relative ad uno studente: nome, cognome, matricola, lista esami, in cui lista esami (superati) è una lista di coppie (esame,voto).

- (2a) Scrivere un programma PROLOG che restituisce vero se il suo argomento è una lista che rappresenta un elenco di studenti appena immatricolati (con la lista degli esami vuota).
- (2b) Scrivere un programma PROLOG che, data una lista che rappresenta un elenco di studenti, calcola il numero di studenti che hanno superato almeno un esame.
- (2c) Scrivere un programma PROLOG che, dato un termine che rappresenta uno studente, verifica se lo studente è

"in corso", cioè ha superato almeno 3 esami ed ha la media superiore a 24.

(2d) Scrivere un programma PROLOG che, data una lista che rappresenta un elenco di studenti, calcola la percentuale di studenti non "in corso".

Esercizi per casa

1. costruire l'albero di ricerca per: `?- appartiene(c,[a,c,b]).`
`?- plus1(Y,X,s(s(s(s(s(0)))))).` e `?- reverse([a,b,c],X).`
2. Scrivere i programmi prodotto, potenza, fattoriale, minimo usando le definizioni date per i numeri naturali;
3. Scrivere i programmi suffisso, sottoinsieme, intersezione usando le definizioni date per le liste e rappresentando gli insiemi sottoforma di lista;
4. scrivere un programma per la visita in profondità di un grafo con cicli, rappresentato tramite la relazione `arco(X,Y)`
5. scrivere un programma per l'ordinamento per inserzione
6. Risolvere l'esercizio del numero di Erdos usando le liste

Esercizi per casa

Si consideri la rappresentazione delle informazioni relative ad un elenco di telefonate (numero chiamato, numero chiamante, durata), in cui il numero è caratterizzato da prefisso internazionale prefisso nazionale e numero.

- (2a) Scrivere un programma PROLOG che restituisce vero se il suo argomento è un elenco di telefonate con le caratteristiche specificate.
- (2d) Scrivere un programma PROLOG che dato un un elenco di telefonate ed un numero telefonico del tipo sopra indicato, restituisca il numero totale delle telefonate ricevute da quel numero.

(2c) Scrivere un programma PROLOG che dato un elenco di telefonate ed un numero telefonico del tipo sopra indicato, restituisca il prefisso da cui il numero indicato ha ricevuto il maggior numero di telefonate.