Esercizi su Prolog per Fondamenti di IA

3° Esercitazione

Esercizio 3.1
Si vogliono risolvere problemi di crittoaritmetica del tipo:

```
  A B +                    A B +
  P Q =        oppure      P Q =
  -----                    -----
  X Y                      Q A
```

assegnando ad ogni lettera una cifra diversa. A questo scopo si vuole definire la relazione:

sum(N1,N2,N) "Sommando la lista di caratteri N1 alla lista di caratteri N2 si ottiene la lista di caratteri N e ad ogni carattere incognito viene assegnata una cifra diversa"

Per semplicità si considerano N1, N2 ed N tutte della stessa lunghezza.
Tra gli operatori e le relazioni predefinite del Prolog possono sicuramente essere utili:

nonvar(X) "X non è una variabile (oppure è una variabile già legata)"

A // B "Divisione intera tra A e B"

A mod B "Resto della divisione intera tra A e B"

Esempi di utilizzo di sum(N1,N2,N):

```
  A B +
  C D =
 -----
  E F
```

```
?- sum([A,B],[C,D],[E,F]).
A = 4
B = 1
C = 5
D = 2
E = 9
F = 3
yes
```

```
  D O N A L D +
  G E R A L D =
 ----------------
  R O B E R T
```

```
?- sum([D,O,N,A,L,D],[G,E,R,A,L,D],[R,O,B,E,R,T]).
D = 5
O = 2
N = 6
A = 4
L = 8
G = 1
E = 9
R = 7
B = 3
T = 0 ;
no
```

```
   S E N D +
   M O R E =
 --------------
  M O N E Y
```

```
?- sum([0,S,E,N,D],[0,M,O,R,E],[M,O,N,E,Y]).
S = 7
E = 5
N = 3
D = 1
M = 0
O = 8
R = 2
Y = 6
yes
```

Determinare quante soluzioni esistono per il primo degli esempi indicati; se utile allo scopo, impiegare anche predicati predefiniti non ancora utilizzati.

Esercizio 3.2
Si definisca un predicato `prodotto_cartesiano(L1,L2,L3)` che, date due liste qualsiasi L1 e L2 restituisca la lista L3 delle coppie ordinate che si possono formare con elementi di L1 e L2. Per esempio, se L1=[a,b,c], L2 = [a,d] la lista risultante deve essere L3=[[a,a],[a,d],[b,a],[b,d], [c,a],[c,d]]. Si definiscano tutti i predicati utilizzati, anche se già visti a lezione.

Esercizio 3.3
a) Si definisca il predicato Prolog `numat(List,Num)` che data una lista `List` che può contenere quali elementi atomi o liste a loro volta ricorsivamente contenenti atomi o liste, restituisce in uscita il numero di tutti gli atomi contenuti.

Esempio:
?- numat([[],a,[a, b, c], [[a,[g,[h]]]]],Num).
Num = 7;
No

b) Si scriva un predicato che appiattisca una lista (lista di liste).

plain(L1,L2) "L2 è la lista appiattita della lista di liste L1"

Esempio:
?- plain([1,[2,3,[4]],5,[6]],L).
L = [1,2,3,4,5,6];
no

c) Si scriva un predicato che, data una lista di liste, ordinata in base alla lunghezza delle liste componenti, inserisca in modo ordinato in essa una nuova lista, mantenendo la lista risultato ordinata secondo la lunghezza dei suoi elementi.
Si definiscano tutti i predicati per la soluzione del problema.

insertOL(L1,L,X) "X è la lista ordinata L in cui viene inserita in modo ordinato la lista L1"

Esempio:
?- insertOL([1,a,pippo], [[],[a,a], [t,7,1,pp]],X).
X = [[],[a,a],[1,a,pippo],[t, 7, 1, pp]];
no

Esercizio 3.4
Un giorno il mago rosso venne sfidato dal mago verde ad un duello di veleni. Ciascuno dei due doveva portare il veleno più potente che era riuscito a produrre; ciascuno avrebbe bevuto prima il veleno dell'altro e poi il proprio. Vale infatti la regola che se si beve un veleno e poi uno più potente, allora non si muore, ma il secondo fa da antidoto per il primo. Il mago rosso accettò la sfida, sapendo che il mago verde aveva un veleno molto più potente del suo: quello del mago verde ha potenza 8, mentre il mago rosso ha solo dell'acqua (che non è un veleno e ha potenza 0) ed un veleno a potenza 1. Il mago rosso, però ha pensato di bere qualcosa *prima* della sfida …
Si scriva un predicato Prolog `vivo/1` che prende in ingresso una lista di numeri ed ha successo se bevendo quella sequenza di veleni si sopravvive.

Esempio:

?- vivo([1,8,0,1,4]).
Yes
?- vivo([1,2,4,1]).
No

Esercizio 3.5
Si definisca un predicato Prolog `memberlist(L1,L2,L3)` che partendo da una lista L1 ed una lista di liste L2 ne restituisca una in uscita composta dai soli elementi della lista L1 che appartengono agli elementi (liste) di L2 in medesima posizione. In altre parole, l'*n*-esimo elemento di L1 va riportato nella lista L3 solo se appartiene anche all'*n*-esimo elemento di L2
Si definiscano tutti i predicati utilizzati, anche se già visti a lezione.

Esempio:

?- memberlist([2,5,7,9], [[3,4], [2,5], [7], []], L3).
L3= [5,7];
No