

## COMPITO DI APPLICAZIONI DI INTELLIGENZA ARTIFICIALE

12 gennaio 2006 (Punteggio su 30/30; Tempo 2h )

### Esercizio 1 (punti 8)

Dato il seguente training set S:

Allele1	Allele2	Classe
A	A	+
A	B	-
B	?	-
B	A	+
A	B	-
B	C	+
A	B	-
B	A	-
A	B	-
A	?	+
A	B	-
B	C	+
B	A	-
A	C	+

- Si calcoli l'entropia del training set rispetto all'attributo Classe (punti 1)
- Si calcoli il rapporto di guadagno dei due attributi rispetto a questi esempi di training (punti 4)
- si costruisca un albero decisionale ad un solo livello per il training set dato, indicando le etichette delle foglie (numero di esempi finiti nella foglia/numero di esempi finiti nella foglia non appartenenti alla classe della foglia). (punti 1,5)
- si classifichi l'istanza: (punti 1,5)

B	?
---	---

### Esercizio 2 (punti 8)

Un mercante aveva tre figli che alla sua morte si dovevano dividere l'eredità. L'unica difficoltà si ebbe nel dividere le scorte di miele: il mercante aveva 21 barili di miele, di cui 7 pieni, 7 a metà e 7 vuoti. Il mercante lasciò nel testamento che ciascuno dei figli avrebbe dovuto avere la stessa quantità di miele e lo stesso numero di barili. Inoltre, non si doveva travasare del miele da un barile all'altro perché si sarebbe sprecato del miele. Sapendo che ciascuno dei figli non voleva avere più di 4 barili dello stesso tipo (pieni, vuoti, metà), si scriva un programma CLP che risolve il problema.

Alcuni predicati utili:

*matrix(+N, +M, -Rows, -Cols)*: crea una matrice di variabili, data come lista di liste, organizzata in N righe e M colonne. La matrice è fornita sia come lista di righe sia come lista di colonne (ovvero viene anche fornita la trasposta). Es

?- *matrix(2,3,Rows,Cols)*

Rows = [[A,B,C],[D,E,F]]

Cols = [[A,D],[B,E],[C,F]]

*flatten(+NestedList, ?FlatList)*: data una lista di liste, fornisce una lista "piatta"

?- *flatten([A,B,[C,D],E],L)*

$L = [A,B,C,D,E]$ .

*sumlist(+List, ?Sum)*: Vincolo che lega una lista di variabili con dominio con la variabile che rappresenta la loro somma

### Esercizio 3 (punti 8)

Si consideri il seguente problema: un robot è nella stanza roomA con la luce accesa, deve recarsi in una stanza adiacente (collegata alla prima da una porta) e accendere la luce . Abbiamo a disposizione le seguenti azioni

Accensione luce

`switch(Room)`

`PREC: in(Room), light(Room, off)`

`EFFECT: light(Room on), ¬ light(Room, off)`

Spostamento tra due stanze adiacenti

`move(Room1, Room2)`

`PREC: in(Room1), door(Room1, Room2, open)`

`EFFECT: in(Room2), ¬ in(Room1)`

Apertura porta (si supponga per semplicità che la porta si apre solo da Room1)

`open(Room1, Room2)`

`PREC: in(Room1), near(Room1, Room2), door(Room1, Room2, closed)`

`EFFECT: door(Room1, Room2, open), ¬ door(Room1, Room2, closed)`

Chiusura porta (si supponga per semplicità che la porta si apre solo da Room2)

`close(Room1, Room2)`

`PREC: door(Room1, Room2, closed)`

`EFFECT: door(Room1, Room2, open), ¬ door(Room1, Room2, closed)`

Stato iniziale:

`in(roomA), light(roomA, on), light(roomB, off),`

`door(roomA, roomB, closed), near(roomA, roomB)`

Stato goal: `door(roomA, roomB, closed), light(roomB, on), in (roomB)`

Si mostrino i passi compiuti dall'algorithm POP per risolvere il problema. Si mostrino i causal link e le eventuali minacce.

### Esercizio 4 (punti 6)

Scrivere un metainterprete per Prolog puro con regola di calcolo right-most-

## SOLUZIONE

### Esercizio 1:

a)  $\text{info}(S) = -6/14 * \log_2 6/14 - 8/14 * \log_2 8/14 = 0.985$

b)

$$\text{info}_{\text{Allele1}}(S) = 8/14 * (-3/8 * \log_2 3/8 - 5/8 * \log_2 5/8) + 6/14 * (-3/6 * \log_2 3/6 - 3/6 * \log_2 3/6) = 0.571 * 0.954 + 0.429 * 1 = 0.973$$

$$\text{gain}(\text{Allele1}) = 0.985 - 0.973 = 0.012$$

$$\text{splitinfo}(\text{Allele1}) = -8/14 * \log_2(8/14) - 6/14 * \log_2(6/14) = 0.985$$

$$\text{gainratio}(\text{Allele1}) = 0.012 / 0.985 = 0.012$$

Per calcolare il guadagno dell'attributo Allele2 non si usa l'entropia calcolata su tutto il training set ma solo sugli esempi che hanno Allele2 noto (insieme F):

$$\text{info}(F) = -5/12 * \log_2 5/12 - 7/12 * \log_2 7/12 = 0.980$$

$$\text{info}_{\text{Allele2}}(F) = 4/12 * (-2/4 * \log_2 2/4 - 2/4 * \log_2 2/4) + 5/12 * (-0/5 * \log_2 0/5 - 5/5 * \log_2 5/5) + 3/12 * (-3/3 * \log_2 3/3 - 0/3 * \log_2 0/3) =$$

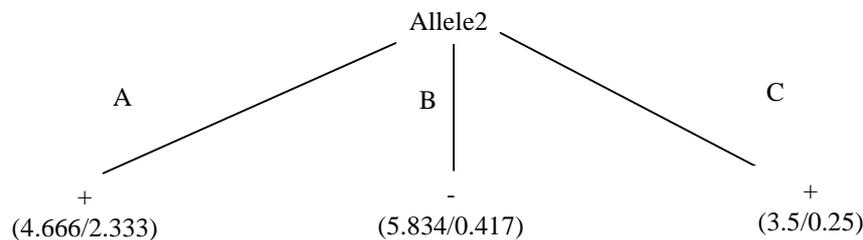
$$= 0.333 * 1 + 0.417 * 0 + 0.25 * 0 = 0.333$$

$$\text{gain}(\text{Allele2}) = 12/14 * (0.980 - 0.333) = 0.555$$

$$\text{splitinfo}(\text{Allele2}) = -4/14 * \log_2(4/14) - 5/14 * \log_2(5/14) - 3/14 * \log_2(3/14) - 2/14 * \log_2(2/14) = 1.924$$

$$\text{gainratio}(\text{Allele2}) = 0.555 / 1.924 = 0.288$$

c) Le istanze avente Allele2 sconosciuto vengono divise in tre parti di pesi rispettivamente  $4/12=0.333$ ,  $5/12=0.417$  e  $3/12=0.25$  che vengono mandate rispettivamente lungo i rami A, B e C.



d) l'istanza viene divisa in tre parti, di peso rispettivamente  $4.666/14=0.333$ ,  $5.834/14=0.417$  e  $3.5/14=0.25$ . La prima parte viene mandata lungo il ramo A e viene classificata come + con probabilità  $2.333/4.666 = 50\%$  e come - con probabilità  $1-50\%=50\%$ . La seconda parte viene mandata lungo il ramo B e viene classificata come - con probabilità  $5.417/5.834=92.9\%$  e come + con probabilità  $1-92.9\%=7.1\%$ . La terza parte viene mandata lungo il ramo C e viene classificata come + con probabilità  $3.25/3.5=92.9\%$  e come - con probabilità  $1-92.9\%=7.1\%$ . Quindi in totale la classificazione dell'istanza è

$$+: 0.333 * 50\% + 0.417 * 7.1\% + 0.25 * 92.9\% = 42.8\%$$

$$-: 0.333 * 50\% + 0.417 * 92.9\% + 0.25 * 7.1\% = 57.2\%$$

## Esercizio 2

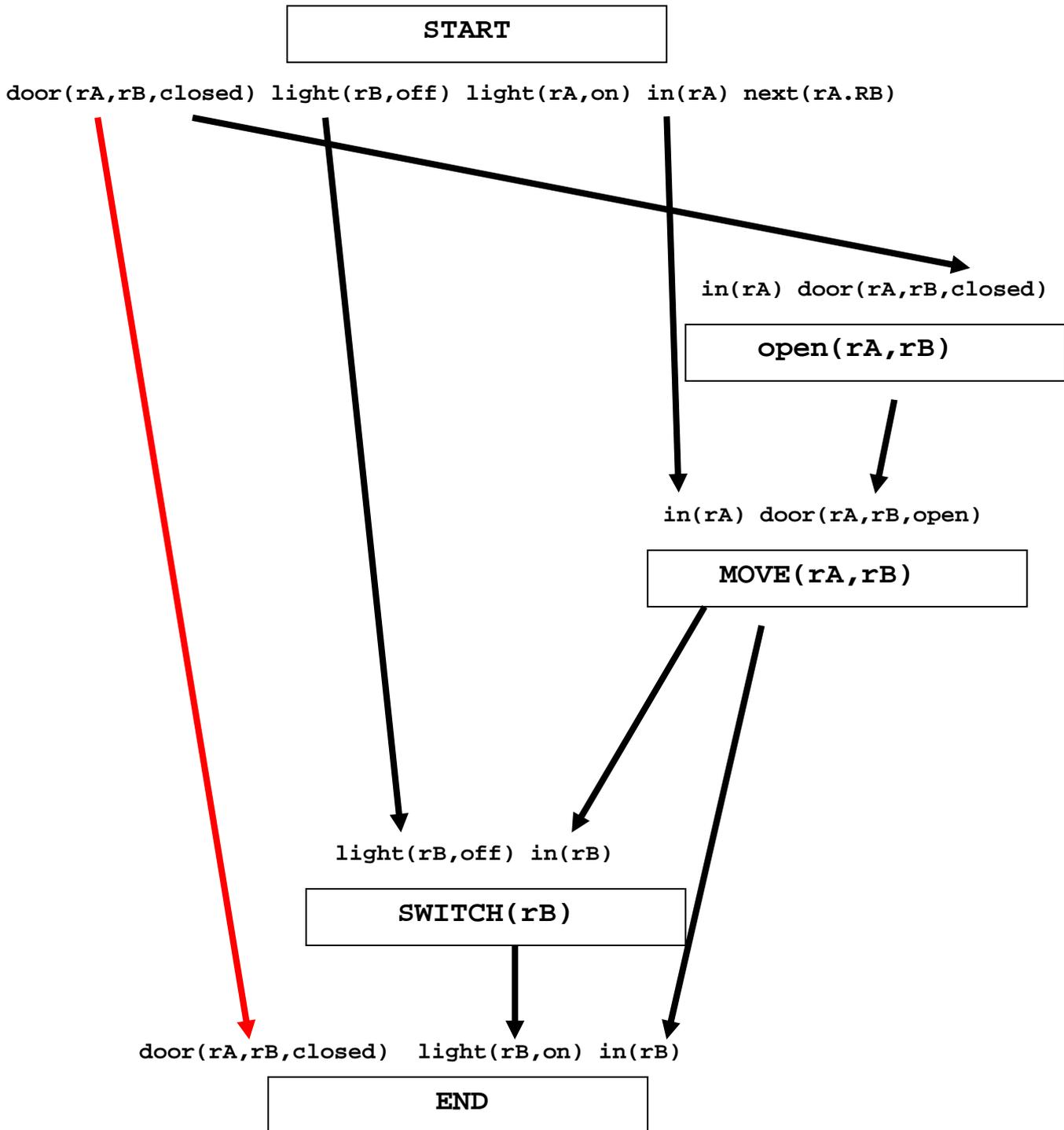
```
:- lib(fd).
:- lib(fd_global).
:- lib(matrix_util).

solve(Figli):-
    matrix(3, 3, Figli, Barili),
    flatten(Figli,Flat),
    % ciascun figlio non vuole piu` di 4 barili dello stesso tipo
    Flat :: 0..4,
    % Il totale di miele deve essere uguale per tutti
    totMiele(Figli,TotMiele),
    % Il numero di barili e` uguale per tutti
    somma_uguale(Figli,TotBarili),
    % Per colonne: il totale di barili di ogni tipo e` uguale a 7
    somma_uguale(Barili,7),
    labeling(Flat).

totMiele([],_).
totMiele([[Vuoto,Mezzo,Pieno]|T],Tot):-
    Vuoto*0+Mezzo+Pieno*2 #= Tot,
    totMiele(T,Tot).

% Impone che tutte le liste della lista di liste diano la stessa somma
somma_uguale([],_).
somma_uguale([H|T],Tot):-
    sumlist(H,Tot),
    somma_uguale(T,Tot).
```

### Esercizio 3



Questo piano contiene un threat infatti il causal link in rosso è minacciato dagli effetti dell'azione `open(rA,rB)` che come effetto contiene `not door(rA,rB,closed)`. Promotion e Demotion non si possono applicare e nemmeno la separazione. Bisogna usare una ulteriore azione per ristabilire la preconditione.

**START**

door(rA,rB,closed) light(rB,off) light(rA,on) in(rA) next(rA.RB)

in(rA) door(rA,rB,closed)

**open(rA,rB)**

in(rA) door(rA,rB,open)

**MOVE(rA,rB)**

light(rB,off) in(rB)

**SWITCH(rB)**

in(rA) door(rA,rB,open)

**close(rA,rB)**

door(rA,rB,closed) light(rB,on) in(rB)

**END**

