

# COMPITO DI APPLICAZIONI DI INTELLIGENZA ARTIFICIALE

13 Luglio 2005 (Punteggio su 30/30; Tempo 2h)

## Esercizio 1 (punti 8)

Dato il seguente training set S:

Att1	Att2	Classe
Rosso	Antico	Tirannosauro
Rosso	?	Stegosauro
Rosso	Recente	Stegosauro
Verde	Antico	Tirannosauro
Giallo	Antico	Tirannosauro
Rosso	Antico	Tirannosauro
Verde	Recente	Stegosauro
Giallo	Antico	Stegosauro
Rosso	?	Tirannosauro
Giallo	Recente	Stegosauro
Rosso	Antico	Tirannosauro
Giallo	Recente	Stegosauro

- Si calcoli l'entropia del training set rispetto all'attributo Classe (punti 1)
- Si calcoli il 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)

Verde	?
-------	---

## Esercizio 2 (punti 8)

Il Sudoku è un rompicapo che si gioca su una scacchiera 9x9. In ciascuna casella va scritto un numero da 1 a 9, in modo che

- in ogni riga non ci siano due numeri uguali
- in ogni colonna non ci siano due numeri uguali
- la scacchiera è suddivisa in 9 sottoscacchiere 3x3: in ciascuna di queste sottoscacchiere non ci devono essere due numeri uguali

4	2	1	9	6	7	8	5	3
6	7	5	3	1	8	4	9	2
3	8	9	2	4	5	6	1	7
1	9	8	7	3	4	5	2	6
7	4	2	8	5	6	1	3	9
5	6	3	1	2	9	7	4	8
2	1	6	5	7	3	9	8	4
8	3	7	4	9	1	2	6	5
9	5	4	6	8	2	3	7	1

Si scriva un programma CLP che risolve il rompicapo.

Alcuni predicati utili:

*matrix(+NRows, +NCols, -Rows, -Cols)*: crea una matrice di variabili, data come lista di liste, organizzata in NRows righe e NCols 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].

*append(A,B,C)*: concatena le liste A e B nella lista C

### Esercizio 3 (Punti 8)

Nello stato iniziale descritto dalle seguenti formule atomiche:

**[at(me,home), in(book,shelf), at(shelf,home), at(table,home), in(coke,fridge), at(fridge,home), handempty, off(stereo),electronic\_equipment(stereo)]**

si vuole raggiungere il goal:

**have(me,book), on(stereo), ontable(coke)**

- le azioni sono modellate opportunamente come segue:

#### **take(Me,Item)**

PRECOND: handempty, at(Me,Location), in(Item,Box), at(Box,Location)

DELETE: handempty, in(Item,Box)

ADD: have(Me,Item)

#### **putonTable(Me,Item)**

PRECOND: have(Me,Item)

DELETE: have(Me,Item)

ADD: handempty, ontable(Item)

#### **switch\_on(ElectrEquipment)**

PRECOND: electronic\_equipment(ElectrEquipment), off(ElectrEquipment)

DELETE: off(ElectrEquipment)

ADD: on(ElectrEquipment)

Si risolva il problema utilizzando l'algoritmo POP. Si evidenzino i causal link e le minacce incontrate.

### Esercizio 4 (Punti 6)

Sia dato il seguente metainterprete Prolog:

```
solve(true):-!.  
solve((A,B)):-!, solve(A), solve(B).  
solve(A):-clause(A,Body), solve(Body).
```

Mostrare come deve essere variato il metainterprete per avere una regola di calcolo right-most e accesso alle clausole rappresentate da fatti del tipo:

```
db( Testa , Corpo ) .
```

dove Testa è un atomo e Corpo è una lista di atomi.

## SOLUZIONE

### Esercizio 1:

a)  $\text{info}(S) = -6/12 \cdot \log_2 6/12 - 6/12 \cdot \log_2 6/12 = 1$

b)

$$\text{info}_{\text{Att1}}(S) = 6/12 \cdot (-4/6 \cdot \log_2 4/6 - 2/6 \cdot \log_2 2/6) + 2/12 \cdot (-1/2 \cdot \log_2 1/2 - 1/2 \cdot \log_2 1/2) + 4/12 \cdot (-1/4 \cdot \log_2 1/4 - 3/4 \cdot \log_2 3/4) =$$

$$= 0,5 \cdot 0,918 + 0,167 \cdot 1 + 0,333 \cdot 0,811 = 0,896$$

$$\text{gain}(\text{Att1}) = 1 - 0,896 = 0,104$$

$$\text{splitinfo}(\text{Att1}) = -6/12 \cdot \log_2(6/12) - 2/12 \cdot \log_2(2/12) - 4/12 \cdot \log_2(4/12) = 1,459$$

$$\text{gainratio}(\text{Att1}) = 0,104 / 1,459 = 0,071$$

Per calcolare il guadagno dell'attributo Att2 non si usa l'entropia calcolata su tutto il training set ma solo sugli esempi che hanno Att2 noto (insieme F):

$$\text{info}(F) = -5/10 \cdot \log_2 5/10 - 5/10 \cdot \log_2 5/10 = 1$$

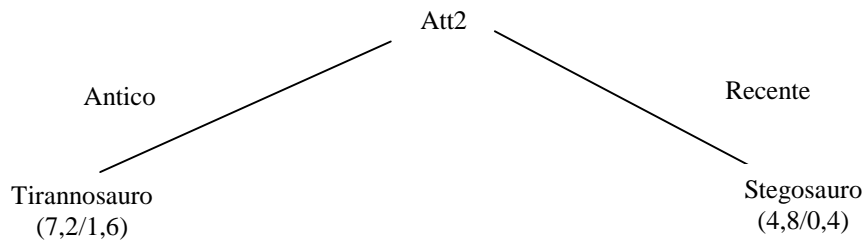
$$\text{info}_{\text{Att2}}(F) = 6/10 \cdot (-5/6 \cdot \log_2 5/6 - 1/6 \cdot \log_2 1/6) + 4/10 \cdot (-0/4 \cdot \log_2 0/4 - 4/4 \cdot \log_2 4/4) =$$
$$= 0,6 \cdot 0,650 + 0,4 \cdot 0 = 0,390$$

$$\text{gain}(\text{Att2}) = 10/12 \cdot (1 - 0,390) = 0,508$$

$$\text{splitinfo}(\text{Att2}) = -6/12 \cdot \log_2(6/12) - 4/12 \cdot \log_2(4/12) - 2/12 \cdot \log_2(2/12) = 1,459$$

$$\text{gainratio}(\text{Att2}) = 0,508 / 1,459 = 0,348$$

c)



d) l'istanza viene divisa in due parti, una di peso  $7,2/12=0,6$  e l'altra di peso  $4,8/12=0,4$ . La prima parte viene mandata lungo il ramo Antico e viene classificata come Tirannosauro con probabilità  $5,6/7,2=77,8\%$  e come Stegosauro con probabilità  $1,6/7,2=22,2\%$ . La seconda parte viene mandata lungo il ramo Recente e viene classificata come Stegosauro con probabilità  $4,4/4,8=91,7\%$  e come Tirannosauro con probabilità  $0,4/4,8=8,3\%$ . Quindi in totale la classificazione dell'istanza è

$$\text{Tirannosauro: } 0,6 \cdot 77,8\% + 0,4 \cdot 8,3\% = 50\%$$

$$\text{Stegosauro: } 0,6 \cdot 22,2\% + 0,4 \cdot 91,7\% = 50\%$$

## Esercizio 2

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

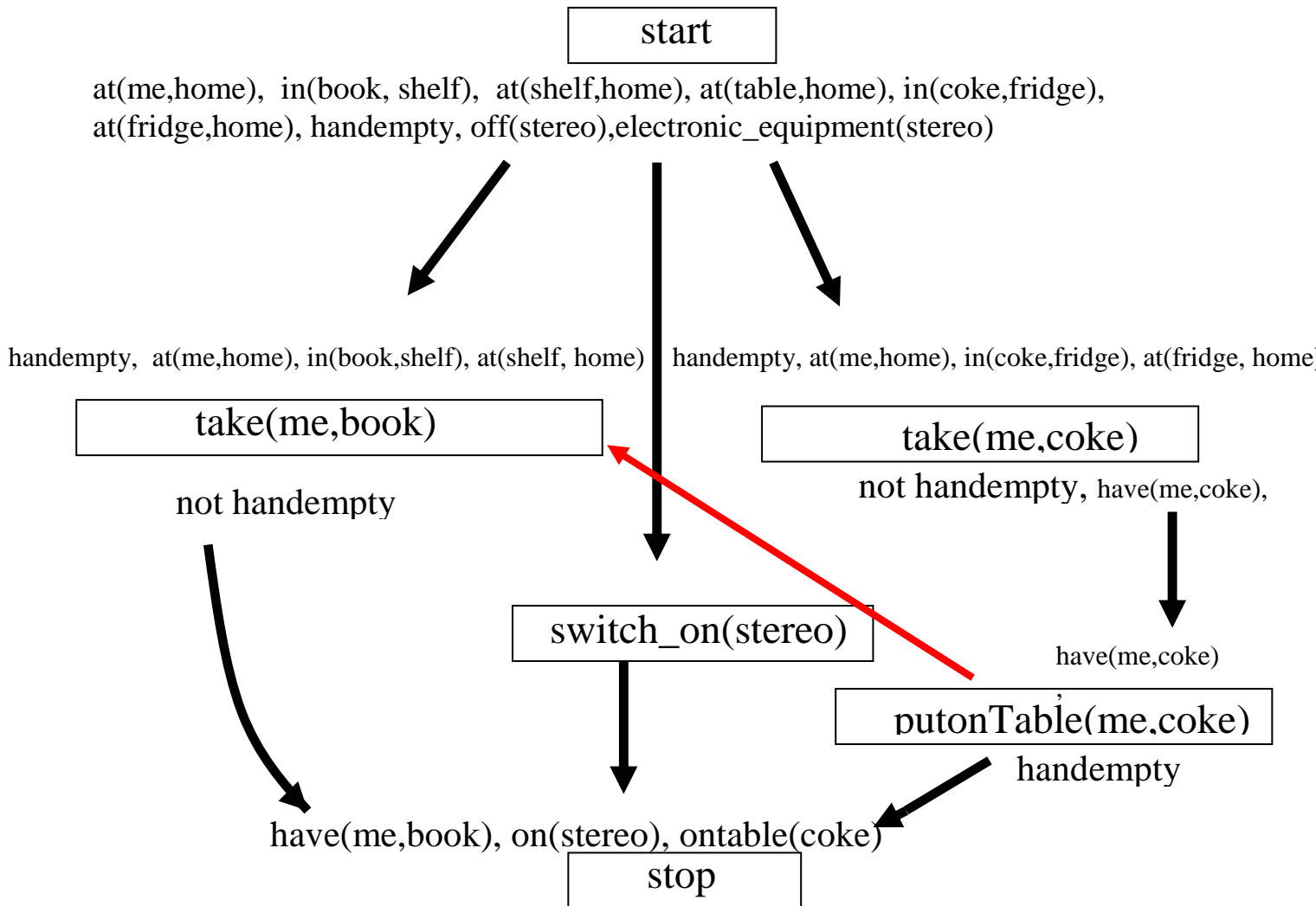
sudoku(Mat):-
    matrix(9,9,Mat,MatT),
    flatten(Mat,List),
    List :: 1..9,
    impose_all_diff_row(Mat), % impongo che le righe siano diverse
    impose_all_diff_row(MatT),% impongo che le colonne siano diverse
    group(Mat,G1,G2,G3), % Divido la matr. in 3 liste (di 3 righe)
    subgroup(G1,G11,G12,G13), % Divido ciascuna lista in 3 parti
    subgroup(G2,G21,G22,G23),
    subgroup(G3,G31,G32,G33),
    impose_all_diff_row([G11,G12,G13,G21,G22,G23,G31,G32,G33]),
    labeling(List).

impose_all_diff_row([]).
impose_all_diff_row([H|T]):-
    fd_global:alldifferent(H),
    impose_all_diff_row(T).

group(L,G1,G2,G3):-
    length(G1,3),
    length(G2,3),
    length(G3,3),
    append(G1,G2,G12),
    append(G12,G3,L).

subgroup([],[],[],[]).
subgroup([H|T],G1,G2,G3):-
    group(H,H1,H2,H3),
    append(H1,L1,G1),
    append(H2,L2,G2),
    append(H3,L3,G3),
    subgroup(T,L1,L2,L3).
```

### Esercizio 3



Il piano fino a qui creato contiene delle minacce in particolare i causal link <Start, take(me,book), handempty> e <Start, take(me,coke), handempty> sono minacciati rispettivamente da due azioni take(me,coke) e take(me,book). E' possibile inserire un vincolo di ordinamento tra l'azione putonTable e take(me,book) per risolvere il conflitto.