COMPITO DI APPLICAZIONI DI INTELLIGENZA ARTIFICIALE
8 Marzo 2004 (Tot. 30/30) Tempo: 2h

Esercizio 1 (punti 8)

Si consideri il problema di stabilire se un'automobile è sportiva o familiare sulla base del colore della carrozzeria, della presenza o meno di un'antenna e del numero di ruote. Si vuole apprendere un albero di decisione per risolvere tale problema partendo dal seguente insieme di dati

Colore	Antenna	Ruote	Tipo
Rosso	Sì	4	Sportiva
Rosso	No	4	Sportiva
Rosso	No	3	Sportiva
Verde	No	3	Sportiva
Verde	No	4	Familiare
Blu	No	4	Familiare
Blu	No	4	Familiare
Blu	No	4	Familiare
Blu	Sì	3	Familiare
?	Sì	3	Sportiva

a) si indichi l'entropia del training set rispetto alla variabile Tipo
b) si calcoli il rapporto di guadagno per i tre attributi Colore, Antenna e Ruote.
c) 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).
d) si classifichi la seguente istanza:

Colore	Antenna	Ruote
Verde	Sì	3

Esercizio 2 (punti 8)

C'è un'isola abitata esclusivamente da cavalieri e da mascalzoni. I cavalieri dicono sempre la verità, i mascalzoni mentono sempre. Su un autobus ci sono *N* passeggeri. Alla domanda "quanti cavalieri ci sono in questo autobus?" rispondono così:

primo passeggero : al massimo 1
secondo : al massimo 2
terzo : al massimo 3
...
N-esimo : al massimo *N*

Si scriva un programma ECLiPSe che, dato *N*, fornisce una lista con *N* elementi in cui l'*i*-esimo elemento assume valore 1 se il corrispondente passeggero è un cavaliere.

Esercizio 3 (punti 8)

È dato lo stato iniziale descitto dalle seguenti formule atomiche:

[ontable(a), ontable(d), on(c,d), clear(a), clear(c), handempty]

(a,c,d rappresentano dei blocchi e si suppone ci siano infinite posizioni occupabili del tavolo

- le azioni sono modellate opportunamente come segue:

pickup(X)
PRECOND: ontable(X), clear(X), handempty
DELETE: ontable(X), clear(X), handempty
ADD: holding(X)

putdown(X)
PRECOND: holding(X)
DELETE: holding(X)
ADD: ontable(X), clear(X), handempty

stack(X,Y)
PRECOND: holding(X), clear(Y)
DELETE: holding(X), clear(Y)
ADD: handempty, on(X,Y), clear(X)

unstack(X,Y)
PRECOND: handempty, on(X,Y), clear(X)
DELETE: handempty, on(X,Y), clear(X)
ADD: holding(X), clear(Y)

e il goal **on(a,c)**

si descriva come l'algoritmo lineare backward STRIPS trova un piano per questo goal (quindi si mostri SOLO una strada nell'albero di ricerca considerando come ordinamento tra precondizioni in and nello stack quello che considera prima le precondizioni che unificano direttamente con un letterale dello stato e poi quelle che per essere soddisfatte hanno bisogno di una azione). Si descriva lo stato e lo stack dei goal passo passo.

Esercizio 4 (punti 6)

Si descriva lo schema architetturale di un sistema esperto e le principali modalità di ragionamento del suo motore inferenziale.

SOLUZIONE

Esercizio 1

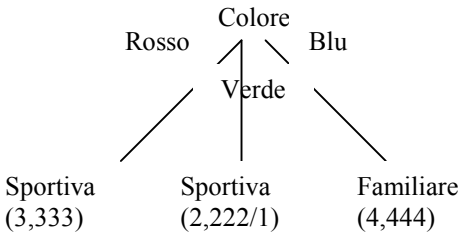
a) Entropia=$-5/10*\log_2 5/10 -5/10*\log_2 5/10=1$

b)Per caclolare il guadagno dell'attributo Colore non si usa l'entropia calcolata su tutto il training set ma solo sugli esempi che hanno Colore noto:
info(F)=$-4/9*\log_2 4/9 -5/9*\log_2 5/9=0,991$
Per gli altri attibuti invece si utilizza info(T)

Gain(Colore)=$9/10*(0,991-3/9*(-3/3*\log_2 3/3 -0/3*\log_2 0/3)-2/9*(-1/2*\log_2 1/2-1/2*\log_2 1/2)-4/9(-4/4*\log_2 4/4 -0/4*\log_2 0/4)=$
$0,9*(0,991-0,333*0-0,222*1-0,444*0)=0,692$
Splitinfo(Colore)=$-3/10*\log_2 3/10-2/10*\log_2 2/10-4/10*\log_2 4/10-1/10*\log_2 1/10-=1,846$
Gainratio(Colore)=$0,692/1,846=0,375$

Gain(Antenna)=$1-3/10*(-2/3*\log_2 2/3 -1/3*\log_2 1/3)-7/10*(-3/7*\log_2 3/7 -4/7*\log_2 4/7)=$
$1-0,3*0,918-0,7*0,985=0,035$
Splitinfo(Antenna)=$-3/10*\log_2 3/10-7/10*\log_2 7/10=0,881$
Gainratio(Antenna)=$0,035/0,881=0,039$

Gain(Ruote)=$1-6/10*(-2/6*\log_2 2/6 -4/6*\log_2 4/6)-4/10*(-3/4*\log_2 3/4 -1/4*\log_2 1/4)=$
$1-0,6*0,918-0,4*0,811=0,125$
Splitinfo(Ruote)=$-6/10*\log_2 6/10-4/10*\log_2 4/10=0,971$
Gainratio(Ruote)=$0,125/0,971=0,129$
d) Guadagno=$0,991-3/9*0-2/9*1-4/9*0=0,991-0,222=0,769$
e)



f) L'istanza viene classificata nella foglia centrale quindi la classificazione risultante e'
classe Sportiva, probabilità $1,222/2,222=0,550=55\%$
classe Familiare, probabilità $1/2,222=0,450=45\%$

Esercizio 2

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

cavalieri(L,N) :-
      length(L,N),     % Creo una lista di N elementi
      L :: 0..1,       % Assegno i domini
      % vincolo che calcola il numero di cavalieri
      sumlist(L,NumCavalieri),
```

```
      % impongo i vincoli che dicono chi ha detto la verita`
      vincoli_verita(L,1,N,NumCavalieri),
      labeling(L).

% X e` l'i-esimo elemento della lista.
% X=1 se il numero dei cavalieri e` minore o uguale ad I
vincoli_verita([],_,_,_).
vincoli_verita([X|L],I,N, NumCavalieri):-
    NumCavalieri #=< I #<=> X,
    Next is I+1,
    vincoli_verita(L,Next,N, NumCavalieri).
```

Esercizio 3

Stato:
ontable(a)
ontable(d)
on(c,d)
clear(a)
clear(c)
handempty

Stack:
on(a,c)

Goal regression con l'azione **stack(X,Y) :**

Stato:
ontable(a)
ontable(d)
on(c,d)
clear(a)
clear(c)
handempty

Stack:
holding(a) and clear(c)
stack(a,c)
on(a,c)

Stato:
ontable(a)
ontable(d)
on(c,d)
clear(a)
clear(c)
handempty

Stack:
clear(c)
holding(a)
stack(a,c)

on(a,c)

Stato:
ontable(a)
ontable(d)
on(c,d)
clear(a)
clear(c)
handempty

Stack:
holding(a)
stack(a,c)
on(a,c)

Goal regression con **pickup(a):**

Stato:
ontable(a)
ontable(d)
on(c,d)
clear(a)
clear(c)
handempty

Stack:
ontable(a) and clear(a) and handempty
pickup(a)
stack(a,c)
on(a,c)

Stato:
ontable(a)
ontable(d)
on(c,d)
clear(a)
clear(c)
handempty

Stack:
ontable(a)
clear(a)
handempty
pickup(a)
stack(a,c)
on(a,c)

Stato:
ontable(a)
ontable(d)
on(c,d)
clear(a)
clear(c)
handempty

Stack:
pickup(a)
stack(a,c)
on(a,c)

Esecuzione di **pickup(a):**

Stato:
ontable(d)
on(c,d)
clear(c)
holding(a)

Stack:
stack(a,c)
on(a,c)

Esecuzione di **stack(a,c):**

Stato:
ontable(d)
on(c,d)
on(a,c)
handempty

Stack:
on(a,c)

Stato:
ontable(d)
on(c,d)
on(a,c)
handempty

Stack: