

## COMPITO DI APPLICAZIONI DI INTELLIGENZA ARTIFICIALE

13 settembre 2004 (Punteggio su 30/30; Tempo 1h)

### Esercizio 1 (punti 8)

Dato il seguente training set S:

A1	A2	Classe
?	A	+
True	A	-
False	C	+
False	A	+
True	C	-
False	C	-
False	B	+
True	B	+
True	B	-
False	C	+

- Si calcoli l'entropia del training set rispetto all'attributo Classe
- Si calcoli il gain ratio dei due attributi rispetto a questi esempi di training.
- 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).
- si classifichi l'istanza

True	A
------	---

### Esercizio 2 (punti 8)

Si consideri il problema di trasportare un carico tramite un carrello ferroviario che puo' percorrere le tratte tra città connesse. Inizialmente il carico è a milano, già caricato sul carrello ferroviario. Come goal il carico deve arrivare a roma.

Caricamento di un oggetto

```
load(Oggetto,Carrello,Location)
PREC:at(Oggetto,Location), at(Carrello,Location)
ADD LIST: in(Oggetto,Carrello)
DELETE LIST: at(Oggetto,Location)
```

Trasporto

```
drive(Carrello,Location1,Location2)
PREC: at(Carrello,Location1), connected(Location1,Location2)
ADD LIST: at(Carrello,Location2)
DELETE LIST: at(Carrello,Location1)
```

Scaricamento di un oggetto

```
unload(Oggetto,Carrello,Location)
PREC:at(Carrello,Location), in(Oggetto,Carrello)
ADD LIST: at(Oggetto,Location)
```

**DELETE LIST:** `in(Oggetto,Carrello)`

Stato iniziale: `in(carico1,carrello1), at(carrello1,milano)`  
`connected(milano,bologna), connected(bologna,roma)`

Stato goal: `at(carico1,roma)`

Si modellino il problema di pianificazione tramite la formulazione di Kowalski.

### **Esercizio 3 (punti 8)**

Cinque uomini e una scimmia fecero naufragio su un'isola deserta e passarono il primo giorno a raccogliere noci di cocco. Poi le ammassarono tutte insieme e andarono a dormire. Ma mentre tutti dormivano uno di essi si svegliò e pensando che il mattino dopo ci sarebbero stati dei litigi alla spartizione decise di prendersi la sua parte. Perciò divise le noci in cinque mucchi uguali. Rimaneva una noce che egli dette alla scimmia, poi nascose la sua parte e mise tutto il resto assieme. Subito dopo un secondo uomo si svegliò e fece lo stesso. Anch'egli dette una noce residua alla scimmia. Uno dopo l'altro tutti e cinque gli uomini fecero la stessa cosa, ognuno prendendo un quinto del mucchio che trovava svegliandosi e dando una noce alla scimmia.

Alla mattina seguente i cinque uomini si divisero le noci che restavano e la divisione non diede resto.

Si scriva un programma CLP che calcola quante noci c'erano all'inizio (supponendo che fossero meno di 10.000).

### **Esercizio 4 (punti 6)**

Si descrivano sinteticamente analogie e/o differenze tra la tecnica di pianificazione come ricerca nello spazio degli stati e la pianificazione come deduzione (vedi esercizio 2).

## SOLUZIONE

### Esercizio 1:

a)  $\text{info}(S) = -6/10 \cdot \log_2 6/10 - 4/10 \cdot \log_2 4/10 = 0,971$

b) Per calcolare il guadagno dell'attributo A1 non si usa l'entropia calcolata su tutto il training set ma solo sugli esempi che hanno A1 noto (insieme F):

$$\text{info}(F) = -4/9 \cdot \log_2 4/9 - 5/9 \cdot \log_2 5/9 = 0,991$$

$$\text{info}_{A1}(F) = 4/9 \cdot (-1/4 \cdot \log_2 1/4 - 3/4 \cdot \log_2 3/4) + 5/9 \cdot (-4/5 \cdot \log_2 4/5 - 1/5 \cdot \log_2 1/5) = 0,556 \cdot 0,811 + 0,444 \cdot 0,722 = 0,771$$

$$\text{gain}(A1) = 9/10 \cdot (0,991 - 0,771) = 0,198$$

$$\text{splitinfo}(A1) = -4/10 \cdot \log_2(4/10) - 5/10 \cdot \log_2(5/10) - 1/10 \cdot \log_2(1/10) = 1,361$$

$$\text{gainratio}(A1) = 0,198 / 1,361 = 0,145$$

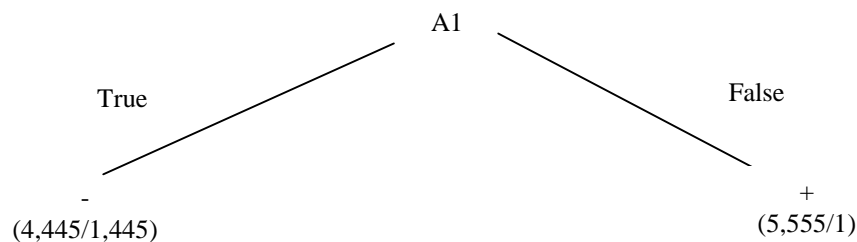
$$\text{info}_{A2}(S) = 3/10 \cdot (-2/3 \cdot \log_2 2/3 - 1/3 \cdot \log_2 1/3) + 3/10 \cdot (-1/3 \cdot \log_2 1/3 - 2/3 \cdot \log_2 2/3) + 4/10 \cdot (-2/4 \cdot \log_2 2/4 - 2/4 \cdot \log_2 2/4) = 0,3 \cdot 0,918 + 0,3 \cdot 0,918 + 0,4 \cdot 1 = 0,951$$

$$\text{gain}(A2) = 0,971 - 0,951 = 0,020$$

$$\text{splitinfo}(A2) = -3/10 \cdot \log_2(3/10) - 3/10 \cdot \log_2(3/10) - 4/10 \cdot \log_2(4/10) = 1,571$$

$$\text{gainratio}(A2) = 0,020 / 1,571 = 0,013$$

c)



d) l'istanza viene classificata nella foglia di sinistra, quindi appartiene alla classe - con probabilita'  $3/4,445 = 0,675$  e alla classe + con probabilita'  $1,445/4,445 = 0,325$

### Esercizio 2:

Stato iniziale

```
holds(in(caricol,carrello1),s0)
holds(at(carrello1,milano),s0)
holds(connected(milano,bologna),s0)
holds(connected(bologna,roma),s0)
```

Caricamento di un oggetto:

effetti azione

```
holds(in(Oggetto,Carrello),do(load(Oggetto,Carrello,Z),S)).
```

Frame axiom

```
holds(V,do(load(Oggetto,Carrello,Location),S)):-
```

```
holds(V,S),
V\=at(Oggetto,Location).
```

Precondizioni

```
pact(load(Oggetto,Carrello,Location),S):-
holds(at(Oggetto,Location),S),
holds(at(Carrello,Location),S).
```

Scaricamento di un oggetto

effetti azione

```
holds(at(Ogg,Location),do(unload(Ogg,Carrello,Location),S)).
```

Frame axiom

```
holds(V, do(unload(Oggetto,Carrello,Location),S)):-
holds(V,S),
V\= in(Oggetto,Carrello).
```

Precondizioni

```
pact(unload(Oggetto,Carrello,Location),S):-
holds(in(Oggetto,Carrello),S),
holds(at(Carrello,Location),S).
```

Trasporto

effetti azione

```
holds(at(CArrello,Location2),
do(drive(Carrello,Location1,Location2),S)).
```

Frame axiom

```
holds(V, do(drive(Carrello,Location1,Location2),S)):-
holds(V,S),
V\= at(Carrello, Location1).
```

Precondizioni

```
pact(unload(Oggetto,Carrello,Location),S):-
holds(connected(Location1,Location2),S),
holds(at(Carrello,Location1),S).
```

%Clausola per esprimere la raggiungibilità di uno stato:

```
poss(s0).
poss(do(U,S)):-
poss(S),
pact(U,S).
```

### Esercizio 3

```
noci(Tot) :-  
  [NR1,NP1,NR2,NP2,NR3,NP3,NR4,NP4,NR5,NP5,Div,Tot] :: 0..10000,  
  Tot #> 0,  
  NP1*5+1 #= Tot,           % Numero prese dal Primo  
  NR1 #= Tot - NP1-1,      % Numero rimaste dopo il primo (e la scimmia)  
  NP2*5+1 #= NR1,         % Numero prese dal Secondo  
  NR2 #= NR1-NP2-1,       % Numero rimaste dopo il secondo  
  NP3*5+1 #= NR2,         % ...  
  NR3 #= NR2-NP3-1,  
  NP4*5+1 #= NR3,  
  NR4 #= NR3-NP4-1,  
  NP5*5+1 #= NR4,  
  NR5 #= NR4-NP5-1,      % Numero rimaste dopo il quinto  
  Div*5 #= NR5,          % Numero delle Noci che si divisero alla fine  
  labeling([NR1,NP1,NR2,NP2,NR3,NP3,NR4,NP4,NR5,NP5,Div,Tot]).
```

(avevano raccolto 3121 noci di cocco).