

## COMPITO DI APPLICAZIONI DI INTELLIGENZA ARTIFICIALE

13 settembre 2006 (Punteggio su 30/30; Tempo 2h )

### Esercizio 1 (punti 8)

Dato il seguente training set S:

Vagoni	Carico	Direzione
Due	Pesante	Est
Tre	Pesante	Ovest
Due	Leggero	Ovest
Tre	Leggero	Est
Quattro	Pesante	Ovest
?	Leggero	Est
Quattro	Pesante	Ovest
?	Leggero	Ovest
Tre	Pesante	Est
Quattro	Pesante	Ovest
Quattro	Pesante	Est
Quattro	Pesante	Ovest
Tre	Leggero	Est
Due	Leggero	Ovest
Due	Leggero	Ovest
Quattro	Pesante	Est

- Si calcoli l'entropia del training set rispetto all'attributo Direzione (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)

?	Pesante
---	---------

### Esercizio 2 (punti 8)

Un telefono cellulare è dotato di un microprocessore che può funzionare a tre diverse frequenze di clock (1, 2 e 3 MHz). Il cellulare deve eseguire la stessa sequenza di operazioni ogni TD millisecondi. Ciascuna delle operazioni da eseguire ha una durata base DB, che è la durata dell'operazione se eseguita a 1MHz (se eseguita a 2MHz la durata sarà DB/2, se eseguita a 3MHz sarà DB/3). Ciascuna delle operazioni consuma energia dalla batteria. Il consumo per un'operazione di durata  $D$  eseguita alla frequenza  $F$  è pari a  $E = (2 * F - 1) * D$ . Si scriva un predicato CLP

`cell(ListaDurateBase, TD, ListaFreq)`

che prende in ingresso

- ListaDurateBase: le durate base delle operazioni da svolgere,
- TD: il tempo entro cui tutte le operazioni devono essere svolte

e che fornisce in uscita la ListaFreq, che contiene le frequenze a cui devono essere svolte le varie operazioni. Il predicato deve calcolare per ogni operazione la frequenza a cui deve essere eseguita, in modo tale che tutte le operazioni vengano svolte entro il tempo TD, minimizzando l'energia consumata.

Esempio:

```
?- cell([24,30,18],60,F).
```

```
Found a solution with cost 84, F = [1, 1, 3]
```

Si supponga che il linguaggio CLP utilizzato contenga il vincolo `sumlist(L,S)`, che vincola la variabile `S` ad assumere come valore la somma dei valori nella lista `L`.

### Esercizio 3 (punti 8)

Si vuole pianificare l'attività di una periferica costituita da un carrello sospeso su un filo. Nello stato iniziale descritto dalle seguenti formule atomiche:

**[at(middle, carrello), empty(carrello), at(middle, oggettoA), at(middle, oggettoB), nonHanged(carrello)]**

si vuole raggiungere il goal:

**[at(destra, oggettoA), at(sinistra, oggettoB), at(middle, carrello), empty(carrello)]**

- le azioni sono modellate opportunamente come segue:

**move(Carr, Position1, Position2)**

PRECOND: at(Position1, Carr), hanged(Carr)

DELETE: at(Position1, Carr)

ADD: at(Position2, Carr)

**load(Carr, Oggetto, Position)**

PRECOND: at(Position, Carr), at(Position, Oggetto)

DELETE: at(Position, Oggetto), empty(Carr)

ADD: in(Oggetto, Carr)

**unload(Carr, Oggetto, Position)**

PRECOND: at(Position, Carr), in(Oggetto, Carr)

DELETE: in(Oggetto, Carr)

ADD: at(Position, Oggetto), empty(Carr)

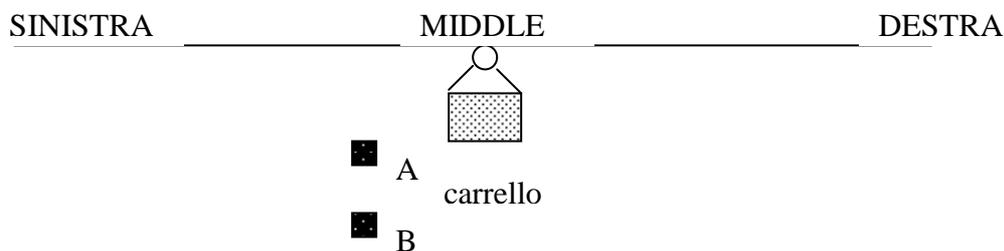
**hang\_on(Carrello)**

PRECOND: nonHanged(Carrello)

DELETE: nonHanged(Carrello)

ADD: hanged(Carrello)

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



### Esercizio 4 (punti 6)

Si descrivano le tecniche di meta-interpretazione in Prolog e si scriva un meta-interprete per Prolog che adotta la regola di calcolo right-most, anziché la left-most.

## SOLUZIONE

### Esercizio 1:

a)  $\text{info}(S) = -9/16 * \log_2 9/16 - 7/16 * \log_2 7/16 = 0.989$

b)

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

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

$$\text{info}_{\text{Vagoni}}(F) = 4/14 * (-3/4 * \log_2 3/4 - 1/4 * \log_2 1/4) + 4/14 * (-1/4 * \log_2 1/4 - 3/4 * \log_2 3/4) + 6/14 * (-4/6 * \log_2 4/6 - 2/6 * \log_2 2/6) =$$

$$= 0.286 * 0.811 + 0.286 * 0.811 + 0.429 * 0.918 = 0.858$$

$$\text{gain}(\text{Vagoni}) = 14/16 * (0.989 - 0.858) = 0.115$$

$$\text{splitinfo}(\text{Vagoni}) = -4/16 * \log_2(4/16) - 4/16 * \log_2(4/16) - 6/16 * \log_2(6/16) - 2/16 * \log_2(2/16) = 1.906$$

$$\text{gainratio}(\text{Vagoni}) = 0.115 / 1.906 = 0.060$$

$$\text{info}_{\text{Carico}}(S) = 9/16 * (-5/9 * \log_2 5/9 - 4/9 * \log_2 4/9) + 7/16 * (-4/7 * \log_2 4/7 - 3/7 * \log_2 3/7) =$$

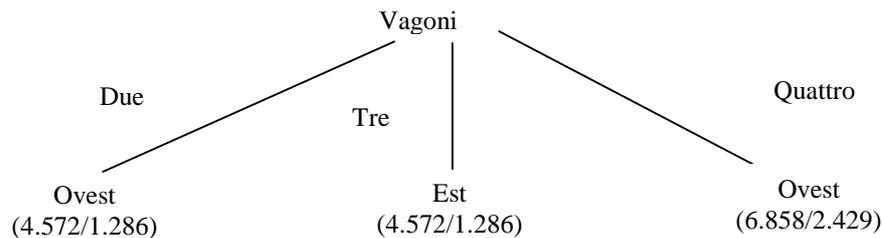
$$= 0.562 * 0.991 + 0.437 * 0.985 = 0.987$$

$$\text{gain}(\text{Carico}) = 0.989 - 0.987 = 0.002$$

$$\text{splitinfo}(\text{Carico}) = -9/16 * \log_2(9/16) - 7/16 * \log_2(7/16) = 0.989$$

$$\text{gainratio}(\text{Carico}) = 0.002 / 0.989 = 0.002$$

c) L'attributo scelto per la radice dell'albero è Vagoni.



d) l'istanza viene divisa in tre parti, di peso rispettivamente  $4.572/16=0.286$ ,  $4.572/16=0.286$  e  $6.858/16=0.429$ . La prima parte viene mandata lungo il ramo Due e viene classificata come Ovest con probabilità  $3.286/4.572=71.9\%$  e come Est con probabilità  $1-71.9\%=28.1\%$ . La seconda parte viene mandata lungo il ramo Tre e viene classificata come Est con probabilità  $3.286/4.572=71.9\%$  e come Ovest con probabilità  $1-71.9\%=28.1\%$ . La terza parte viene mandata lungo il ramo Quattro e viene classificata come Ovest con probabilità  $4.429/6.858=64.6\%$  e come Est con probabilità  $1-64.6\%=35.4\%$ . Quindi in totale la classificazione dell'istanza è

$$\text{Ovest: } 0.286 * 71.9\% + 0.286 * 28.1\% + 0.429 * 64.6\% = 56.3\%$$

$$\text{Est: } 0.286 * 28.1\% + 0.286 * 71.9\% + 0.429 * 35.4\% = 43.7\%$$

### Esercizio 2

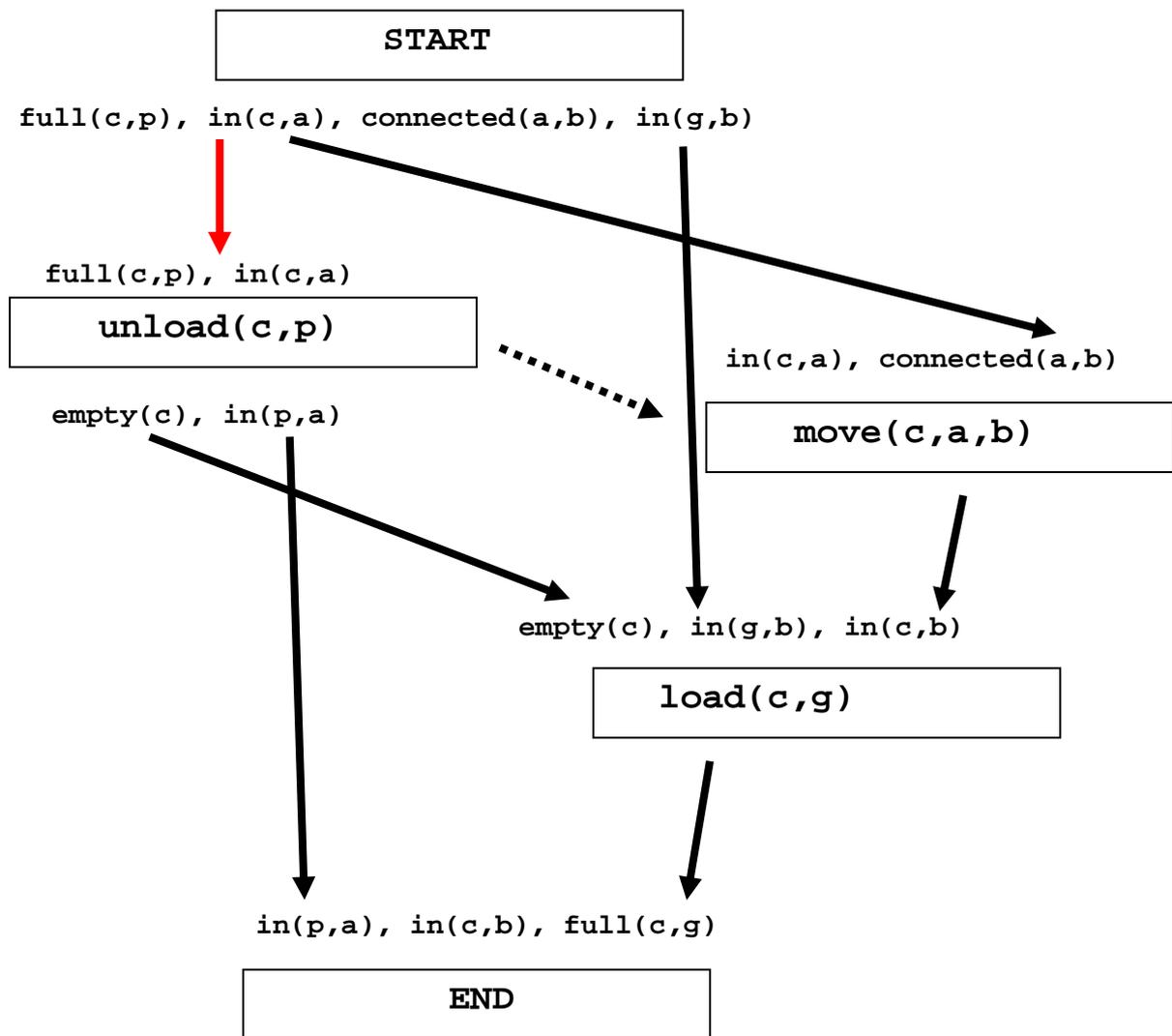
```
cell(DurBase, Dead, F) :-
    length(DurBase, N),
    length(Dur, N),
    length(F, N),
    F :: 1..3,
    prodotto(F, Dur, DurBase),
    energia(F, Dur, Energy),
    sumlist(Energy, EnergyTot),
```

```
sumlist(Dur,DurTot), DurTot #<= Dead,  
minimize(labeling(F),EnergyTot).
```

```
prodotto([],[],[]).  
prodotto([B|LB],[D|LD],[P|LP]):-  
    B*D #= P,  
    prodotto(LB,LD,LP).
```

```
energia([],[],[]).  
energia([F|LF],[D|LD],[E|LE]):-  
    E #= (2*F-1)*D,  
    energia(LF,LD,LE).
```

### Esercizio 3



Questo piano contiene un threat: infatti il *causal link* in rosso (`in(c,a)`) è minacciato dagli effetti dell'azione `move(c,b,a)` che come effetto contiene `not in(c,a)`.

In questo caso si può applicare la Demotion e far sì che `move(c,b,a)` segua `unload(c,p)` (vincolo di successione tratteggiato in figura).