

COMPITO DI APPLICAZIONI DI INTELLIGENZA ARTIFICIALE

11 Luglio 2006 (Punteggio su 30/30; Tempo 2h)

Esercizio 1 (punti 8)

Dato il seguente training set S:

Struttura	Direzione	Classe
Helix	Left	Pos
Strand	Right	Pos
Coil	Center	Neg
Coil	Right	Pos
Coil	Left	Neg
Helix	Left	Neg
?	Center	Pos
Strand	Right	Neg
Coil	Center	Neg
Strand	Left	Pos
Coil	Right	Neg
Helix	Right	Pos
?	Left	Neg
Strand	Center	Pos
Helix	Center	Pos

- 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)

Helix	?
-------	---

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)

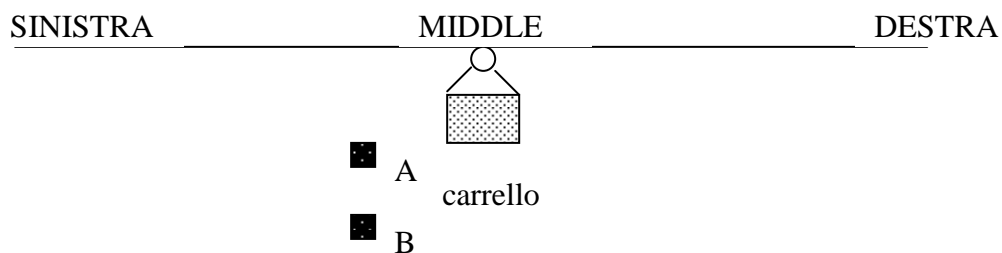
hang_on(Carrello)

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) $\text{info}(S) = -8/15 * \log_2 8/15 - 7/15 * \log_2 7/15 = 0.997$

b)

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

$$\text{info}(F) = -7/13 * \log_2 7/13 - 6/13 * \log_2 6/13 = 0.996$$

$$\text{info}_{\text{Struttura}}(F) = 4/13 * (-3/4 * \log_2 3/4 - 1/4 * \log_2 1/4) + 4/13 * (-3/4 * \log_2 3/4 - 1/4 * \log_2 1/4) + 5/13 * (-1/5 * \log_2 1/5 - 4/5 * \log_2 3/5) =$$

$$= 0.308 * 0.811 + 0.308 * 0.811 + 0.385 * 0.722 = 0.778$$

$$\text{gain}(\text{Struttura}) = 13/15 * (0.996 - 0.778) = 0.189$$

$$\text{splitinfo}(\text{Struttura}) = -4/15 * \log_2(4/15) - 4/15 * \log_2(4/15) - 5/15 * \log_2(5/15) - 2/15 * \log_2(2/15) = 1.933$$

$$\text{gainratio}(\text{Struttura}) = 0.189 / 1.909 = 0.099$$

$$\text{info}_{\text{Direzione}}(S) = 5/15 * (-2/5 * \log_2 2/5 - 3/5 * \log_2 3/5) + 5/15 * (-3/5 * \log_2 3/5 - 2/5 * \log_2 2/5) + 5/15 * (-3/5 * \log_2 3/5 - 2/5 * \log_2 2/5) =$$

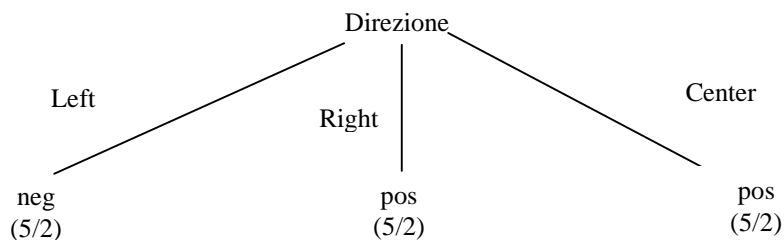
$$= 0.333 * 0.722 + 0.333 * 0.722 + 0.333 * 0.722 = 0.722$$

$$\text{gain}(\text{Direzione}) = 0.997 - 0.722 = 0.275$$

$$\text{splitinfo}(\text{Direzione}) = -7/15 * \log_2(7/15) - 8/15 * \log_2(8/15) = 1.585$$

$$\text{gainratio}(\text{Direzione}) = 0.275 / 1.585 = 0.174$$

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



d) l'istanza viene divisa in tre parti, di peso rispettivamente $5/15=0.333$, $5/15=0.333$ e $5/15=0.333$. La prima parte viene mandata lungo il ramo Left e viene classificata come neg con probabilità $3/5=60\%$ e come pos con probabilità $1-60\%=40\%$. La seconda parte viene mandata lungo il ramo Right e viene classificata come pos con probabilità $3/5=60\%$ e come neg con probabilità $1-60\%=40\%$. La terza parte viene mandata lungo il ramo Center e viene classificata come pos con probabilità $3/5=60\%$ e come neg con probabilità $1-60\%=40\%$. Quindi in totale la classificazione dell'istanza è

$$\text{pos: } 0.333 * 40\% + 0.333 * 60\% + 0.333 * 60\% = 53.3\%$$

$$\text{neg: } 0.333 * 60\% + 0.333 * 40\% + 0.333 * 40\% = 46.7\%$$

Esercizio 2

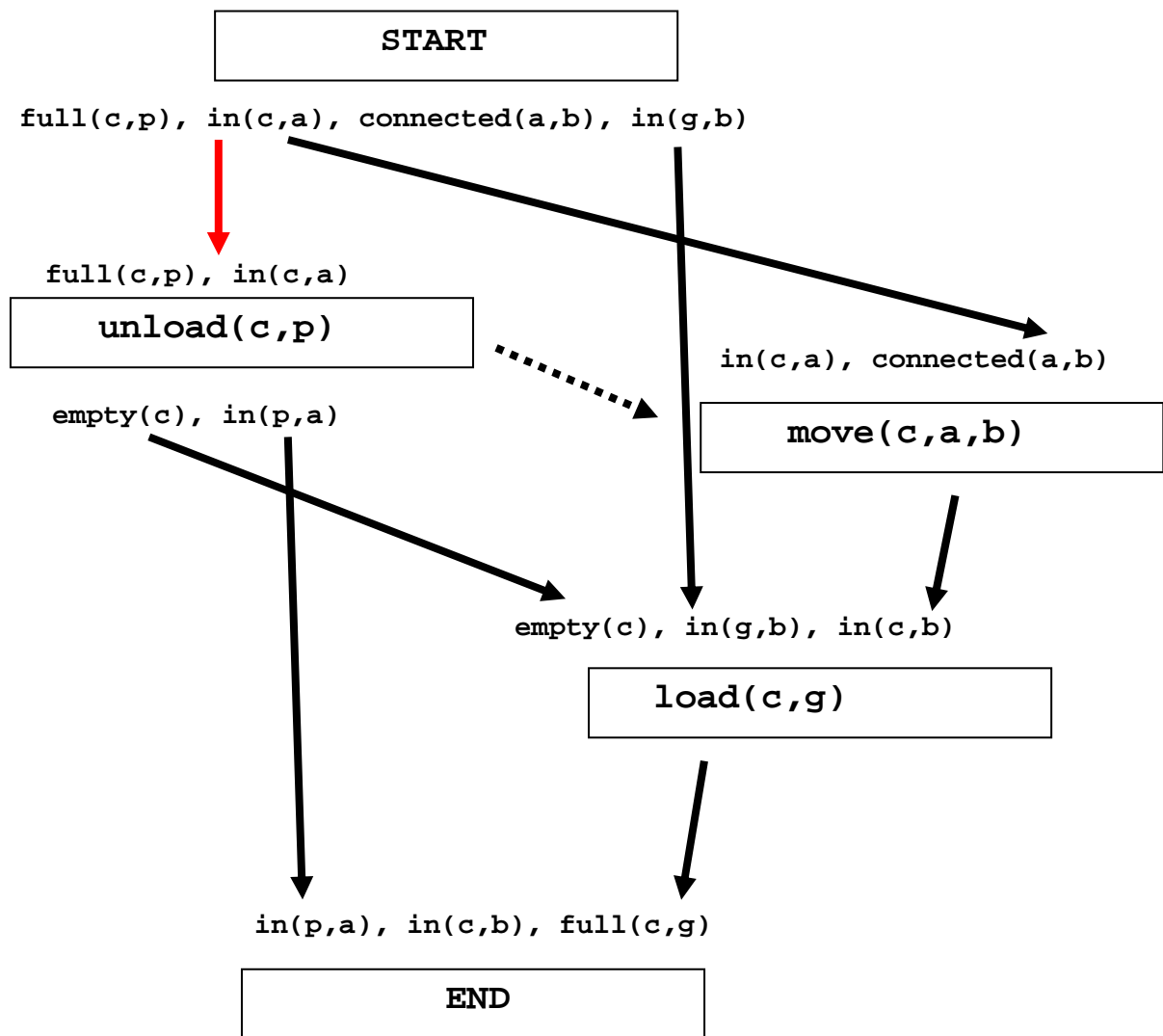
```
cell(DurBase, F, Dead) :-
    length(DurBase, N),
    length(Dur, N),
    length(F, N),
    F :: 1..3,
    prodotto(F, Dur, DurBase),
    energia(F, Dur, Energy),
```

```
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).