

COMPITO DI APPLICAZIONI DI INTELLIGENZA ARTIFICIALE

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

Esercizio 1 (punti 8)

Dato il seguente training set S:

Helical	Single	Classe
Uno	Sì	histone
Tre	No	ire
Due	No	histone
Uno	No	ire
Tre	No	ire
Due	?	histone
Uno	Sì	ire
Tre	Sì	histone
Due	Sì	histone
Tre	No	ire
Uno	Sì	histone
Tre	No	ire
Due	Sì	ire
Uno	No	histone
Due	?	ire
Tre	Sì	histone

- 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)

Tre	?
-----	---

Esercizio 2 (punti 8)

Si supponga che queste siano le regole della battaglia navale:

- la battaglia si gioca su una matrice $N \times M$. In ciascuna cella può esserci una nave oppure acqua;
- le navi sono tutte di dimensioni 1×1 ;
- due navi non possono essere adiacenti in verticale e orizzontale, mentre possono essere adiacenti in diagonale;
- Il numero delle navi in uno schema è noto.

Si scriva un predicato CLP

`battleship(N, M, NNav, Mat)`

che calcola le possibili soluzioni della battaglia navale. Ad esempio, il goal

`?- L=[[A,B,C],[D,E,F],[0,G,1]],battleship(3,3,4,L).`

deve fornire come (unica) soluzione

`L= [[1, 0, 1],
[0, 1, 0],
[0, 0, 1]]`

Si supponga che il linguaggio CLP utilizzato abbia i seguenti predicati già definiti:

`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].

Esercizio 3 (punti 8)

Si consideri il seguente problema: un'auto è nella posizione '1' ed ha a bordo il solo autista (a). Inoltre, vi è una persona (b) nella posizione '2' connessa ad '1'.

Il goal che si vuole raggiungere è che le due persone siano tutte in '1', fuori dell'auto, avendo a disposizione le seguenti azioni

Scaricamento persona

unload(Car,Person)

PREC: on(Car,Person), in(Car,X)

EFFECT: ¬on(Car,Person), in(Person,X)

Caricamento persona

load(Car,Person)

PREC: in(Car,X), in(Person,X)

EFFECT: on(Car,Person), ¬in(Person,X)

Spostamento dell'auto tra due posizioni connesse

move(Car,Loc1,Loc2)

PREC: in(Car,Loc1), connected(Loc1,Loc2)

EFFECT: in(Car,Loc2), ¬in(Car,Loc1),

Stato iniziale:

in(c,1), on(c,a), connected(1,2), connected(2,1), in(b,2)

Stato goal: **in(a,1), in(b,1), in(c,1), ¬on(c,a), ¬on(c,b)**

Si mostrino i passi compiuti dall'algoritmo POP per risolvere il problema. Si mostrino i causal link e le eventuali minacce.

Esercizio 4 (punti 6)

Si descriva l'architettura di un sistema basato su conoscenza e le modalità di funzionamento in forward reasoning e backward reasoning.

SOLUZIONE

Esercizio 1:

a) $\text{info}(S) = -8/16 \cdot \log_2 8/16 - 8/16 \cdot \log_2 8/16 = 1$

b) $\text{info}_{\text{Helical}}(S) = 5/16 \cdot (-3/5 \cdot \log_2 3/5 - 2/5 \cdot \log_2 2/5) + 5/16 \cdot (-3/5 \cdot \log_2 3/5 - 2/5 \cdot \log_2 2/5) + 6/16 \cdot (-2/6 \cdot \log_2 2/6 - 4/6 \cdot \log_2 4/6) =$

$= 0.312 \cdot 0.971 + 0.312 \cdot 0.971 + 0.375 \cdot 0.918 = 0.950$

$\text{gain}(\text{Helical}) = 1 - 0.950 = 0.050$

$\text{splitinfo}(\text{Helical}) = -5/16 \cdot \log_2(5/16) - 5/16 \cdot \log_2(5/16) - 6/16 \cdot \log_2(6/16) = 1.579$

$\text{gainratio}(\text{Helical}) = 0.050 / 1.579 = 0.032$

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

$\text{info}(F) = -7/14 \cdot \log_2 7/14 - 7/14 \cdot \log_2 7/14 = 1$

$\text{info}_{\text{Single}}(F) = 7/14 \cdot (-5/7 \cdot \log_2 5/7 - 2/7 \cdot \log_2 2/7) + 7/14 \cdot (-2/7 \cdot \log_2 2/7 - 5/7 \cdot \log_2 5/7) =$

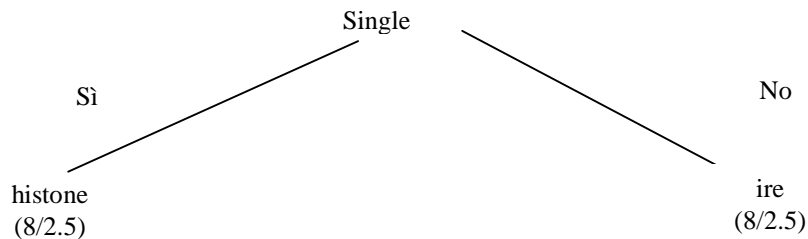
$= 0.5 \cdot 0.863 + 0.5 \cdot 0.863 = 0.863$

$\text{gain}(\text{Single}) = 14/16 \cdot (1 - 0.863) = 0.120$

$\text{splitinfo}(\text{Single}) = -7/16 \cdot \log_2(7/16) - 7/16 \cdot \log_2(7/16) - 2/16 \cdot \log_2(2/16) = 1.419$

$\text{gainratio}(\text{Single}) = 0.120 / 1.419 = 0.085$

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



d) l'istanza viene divisa in due parti, di peso rispettivamente $8/16=0.5$ e $8/16=0.5$. La prima parte viene mandata lungo il ramo Sì e viene classificata come histone con probabilità $5.5/8=68.7\%$ e come ire con probabilità $1-68.7\%=31.3\%$. La seconda parte viene mandata lungo il ramo No e viene classificata come ire con probabilità $5.5/8=68.7\%$ e come histone con probabilità $1-68.7\%=31.3\%$. Quindi in totale la classificazione dell'istanza è

histone: $0.5 \cdot 68.7\% + 0.5 \cdot 31.3\% = 50\%$

ire: $0.5 \cdot 31.3\% + 0.5 \cdot 68.7\% = 50\%$

Esercizio 2

```
battleship(NRows, NCols, NNav, Rows):-
    matrix(NRows, NCols, Rows, Cols),
    flatten(Rows, Flat),
    Flat :: 0..1,
    no_vicini(Rows),
    no_vicini(Cols),
    sumlist(Flat, NNav),
    labeling(Flat).
```

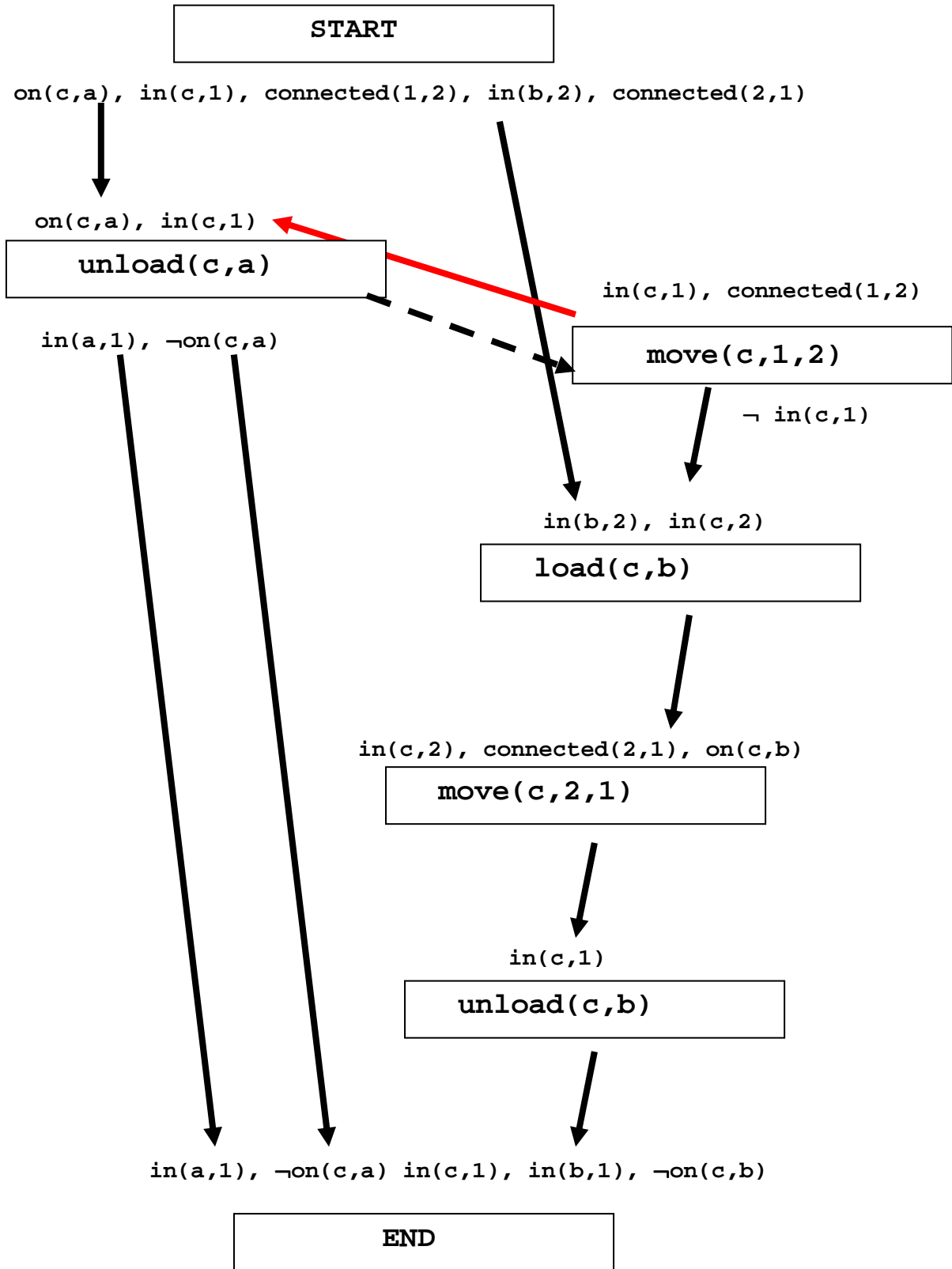
```
no_vicini([]).
```

```
no_vicini([Riga|Righe]):-
    no_vicini_riga(Riga),
    no_vicini(Righe).
```

```
no_vicini_riga([_]).
```

```
no_vicini_riga([A,B|T]):-
    A+B #< 2, % Due celle vicine non possono essere a 1
    no_vicini_riga([B|T]).
```

Esercizio 3



Questo piano contiene un threat: infatti il *causal link* in rosso (`in(c,1)`) è minacciato dagli effetti dell'azione `move(c,1,2)` che come effetto contiene `¬in(c,1)`.

In questo caso si può applicare la Demotion e far sì che `unload(c,a)` preceda `move(c,1,2)` (vincolo di precedenza tratteggiato in figura) e di fatto tutte le azioni successive fino all'azione `END`.