FONDAMENTI DI INTELLIGENZA ARTIFICIALE

PARTE II

15 Luglio 2010 – Tempo a disposizione 45minuti – Risultato 32/32 punti

Esercizio 1 (punti 16)

Si scriva un meta interprete per Prolog che per la selezione delle clausole non segua l'ordine testuale, ma selezioni sempre prima quella che ha un numero minore di sottogoals (a partire dai fatti).
Ad esempio, nel caso delle seguenti clausole:

```
1. p(X,Y,Z):- b(Y, Y),a(X, Y), c(Z).
2. p(X,Y,Z).
3. p(X,Y,Z):- s(Y, Y), s(Y,Y).
4. p(X,Y,Z):- s(Y, Y).
```

L'ordine di selezione sarà 2. 4. 3. e 1.
Si supponga di avere a disposizione un predicato `sort(L, L1, L2)` che fornisce in L2 la lista L ordinata (in senso crescente) secondo i valori contenuti in L1.
Si supponga, inoltre, di avere un programma nella forma `clausola(Head, Body)` dove `Body` è una lista di sottogoal. I fatti hanno come Body la lista vuota.

Esercizio 2 (punti 6)

Si consideri il seguente problema di pianificazione. Ci sia un robot in grado di muoversi, di afferrare e di depositare oggetti all'interno di un ambiente composto da due stanze comunicanti *S1* e *S2*. Lo stato iniziale, l'obiettivo, le precondizioni e gli effetti di ogni azione sono formulati in STRIPS:

STATO INIZIALE (At(Robot,S1) ∧ At(Object2,S1) ∧ At(Object1,S2) ∧ Room(S1) ∧ Room(S2) ∧ HandEmpty)
GOAL (At(Object1,S1) ∧ At(Object2,S1))
ACTION (Go(x,y),
PRECOND: Room(x) ∧ Room(y) ∧ At(Robot,x) ∧ x≠y
EFFECT: ¬ At(Robot,x) ∧ At(Robot,y))
ACTION (Pick(o),
PRECOND: Room(x) ∧ At(Robot,x) ∧ At(o,x) ∧ HandEmpty ∧ o≠Robot
EFFECT: ¬HandEmpty ∧ ¬At(o,x) ∧ Holding(o))
ACTION (Drop(o),
PRECOND: Room(x) ∧ At(Robot,x) ∧ Holding(o)
EFFECT: HandEmpty ∧ At(o,x) ∧ ¬Holding(o))

Le variabili iniziano con lettera minuscola, le costanti conn lettera maiuscola. Si risolva il problema di pianificazione con una **pianificazione in avanti** mediante una ricerca **A* con eliminazione degli stati ripetuti** e considerando come funzione euristica il numero di letterali di goal *non* soddisfatti nello stato.
Nella costruzione dell'albero, si applichino le azioni nell'ordine dato sopra. A parità di altro, si scelga il nodo generato prima. Si rappresentino gli stati in modo grafico. Si riporti l'albero di ricerca generato (indicando, per ogni nodo, il valore della funzione di valutazione e l'ordine di espansione) e la soluzione trovata.
Si noti che l'azione Pick(o) ha come conseguenza che per l'oggetto 'o' verrà a valere ¬At(o,x).

Esercizio 3 (punti 5)

Si spieghi in cosa consiste l'anomalia di Sussmann con un esempio.

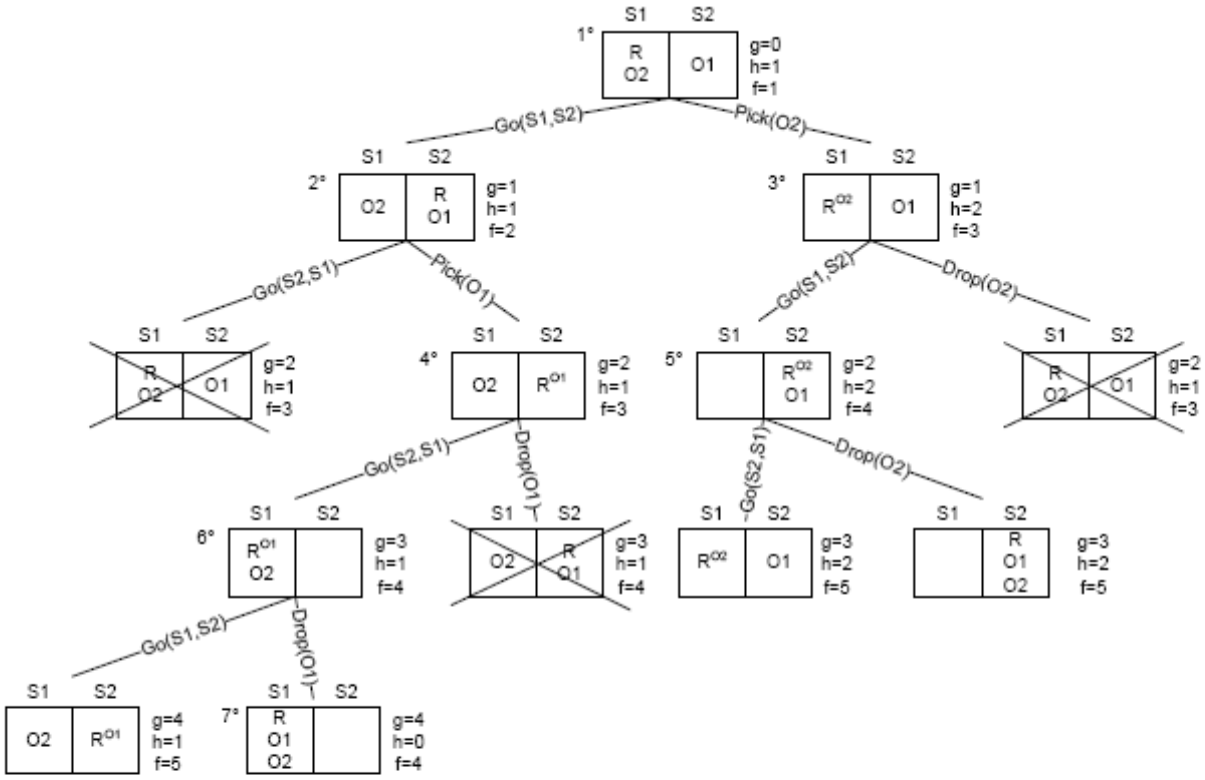
Esercizio 4 (punti 5)

Descrivere brevemente in cosa consiste la Open World Assumption (OWA) e la Close World Assumption (CWA), fornendo un breve esempio di base di conoscenza che mostri come tali assunzioni modificano il ragionamento su tale base di conoscenza.

Soluzione

```
solve([]).
solve([Head|Tail]):-
     solve(Head),
     solve(Tail).

solve(Goal):-
     findall((Goal,Body),clausola(Goal,Body),
                                                    L),
     ordina(L,Lord),
     member((Goal,FirstBody), Lord),
     solve(FirstBody).

ordina(L,Lord):-
     conta_sottogoal(L,Nsottogoal),
     sort(L,Nsottogoal, Lord).

conta_sottogoal([],[]).
conta_sottogoal([H|T],[NH|NT]):-
     conta_sottogoal1(H,NH),
     conta_sottogoal(T,NT).

conta_sottogoal1((_,Body),Nsottogoal):-conta(Body, Nsottogoal)

conta([], 0):- !.
conta([Arg|Tail],  Nout):-
     conta(Tail, Nin), Nout is Nin +1.
```

Esercizio 2



La soluzione trovata è Go(S1,S2), Pick(Object1), Go(S2,S1), Drop(Object1).