

## COMPITO DI INTELLIGENZA ARTIFICIALE.

### Fondamenti di Intelligenza Artificiale e Intelligenza Artificiale Modulo A

16 Dicembre 2005 Tempo 2,30 ore

#### **Esercizio 1 (punti 5)**

Date due liste di atomi, si determini una terza lista che contiene le occorrenze di ogni elemento della prima lista nella seconda lista:

es:

```
?- occorre([e,5,ciao], [e, ciao, a, e, 2], L3).  
yes L3=[2,0,1]
```

#### **Esercizio 2 (punti 7)**

Si consideri il seguente programma Prolog:

```
isground(X):-  
    not(X=skolem).
```

```
reversible(S=Op):-  
    rewrite(S,Op,R=NewOp),  
    R is NewOp.
```

```
rewrite(S,A+B,S=A+B):- isground(A), isground(B), !.  
rewrite(S,A+B,A=S-B):- isground(S), isground(B), !.  
rewrite(S,A+B,B=S-A):- isground(S), isground(A).
```

Si rappresenti l'albero di derivazione SLD relativo al goal: `?- reversible(6=X+2)`. e si dica qual è la risposta calcolata.

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

Un corso che vale 6 crediti è un corso difficile.

Un corso che vale 3 crediti è un corso facile.

Se un corso è facile, c'è almeno uno studente del corso che è felice.

Tutti gli studenti di un corso difficile non sono felici

Il corso di inglese vale 3 crediti

Il corso di analisi vale 6 crediti

Luca segue il corso di inglese e il corso di analisi

Si usi la risoluzione per dimostrare che Luca non è felice.

**Nota:** si utilizzino i predicati:

corso(Corso)

crediti(Corso,N)

facile(Corso)

studente(Studente,Corso)

felice(Studente)

#### **Esercizio 4 (punti 8)**

Il complesso degli U2 sta per fare un concerto a Dublino.

Mancano 17 minuti all'inizio del concerto ma, per raggiungere il palco, i membri del gruppo devono attraversare un piccolo ponte che è tutto al buio, disponendo di una sola torcia elettrica. Sul ponte

non possono andare più di due persone per volta. La torcia è essenziale per l'attraversamento, per cui deve essere portata avanti e indietro (non può essere lanciata da una parte all'altra) per consentire a tutti di passare. Tutti sono dalla stessa parte del ponte.

Ciascun componente del complesso cammina a una velocità diversa. I tempi individuali per attraversare il ponte sono:

- Bono, 1 minuto
- Edge, 2 minuti
- Adam, 5 minuti
- Larry, 10 minuti

Se attraversano in due, la coppia camminerà alla velocità del più lento.

Ad esempio: se Bono e Larry attraversano per primi, quando arrivano dall'altra parte saranno trascorsi 10 minuti. Se Larry torna indietro con la torcia saranno passati altri dieci minuti, per cui la missione sarà fallita.

Si stabilisca una funzione euristica ammissibile per questo problema e lo si risolva tramite l'algoritmo A\* per i grafi, mostrando il grafo generato. Per limitare la dimensione dello spazio di ricerca, si supponga che i componenti si muovano sempre in 2 in una direzione e sempre in 1 nella direzione opposta (quando devono riportare indietro la torcia per prendere gli altri componenti della band).

Per definire la funzione euristica  $h'$  si usi la seguente idea:

Possono spostarsi solo 2 persone alla volta, per cui raggruppiamo le persone sulla riva di partenza (sinistra) in gruppi da 2. Ordino i tempi di percorrenza dal più alto al più basso e metto

- nel primo gruppo i primi due
- nel secondo gruppo il terzo ed il quarto.

Ciascun gruppo si muoverà alla velocità del più lento, per cui per ogni gruppo prendo il massimo. La funzione  $h'$  è data dalla somma dei tempi di percorrenza di ciascun gruppo. In altre parole, il tempo di percorrenza totale sarà almeno il tempo di percorrenza del più lento, più quello del terzo più lento (il secondo più lento potrebbe essersi mosso con il più lento, per cui il suo tempo non viene calcolato).

Nell'algoritmo A\* per i grafi si considerano cammini alternativi per arrivare ad uno stesso stato; in caso di cammini alternativi viene tenuto il costo minore. Rappresentiamo con una freccia piena il cammino migliore e con una freccia tratteggiata i cammini peggiori.

La funzione  $h'$  così definita è ammissibile? L'algoritmo troverà la strada migliore?

### **Esercizio 5 (punti 4)**

Si spieghi la strategia di ricerca non-informata di *iterative deepening*, se ne discuta la complessità spaziale e temporale confrontandola con le altre tecniche di ricerca non informate. Si scriva poi in pseudo-codice il relativo algoritmo di ricerca.

## SOLUZIONE

### Esercizio 1

```

occorre([], _L, []).
occorre([H|Tail], L, [A|B]) :- conta(H,L,A), occorre(Tail, L, B).

```

```

conta(_, [], 0).
conta(A, [A|L], N) :- !,
    conta(A, L, N1),
    N is N1 + 1.

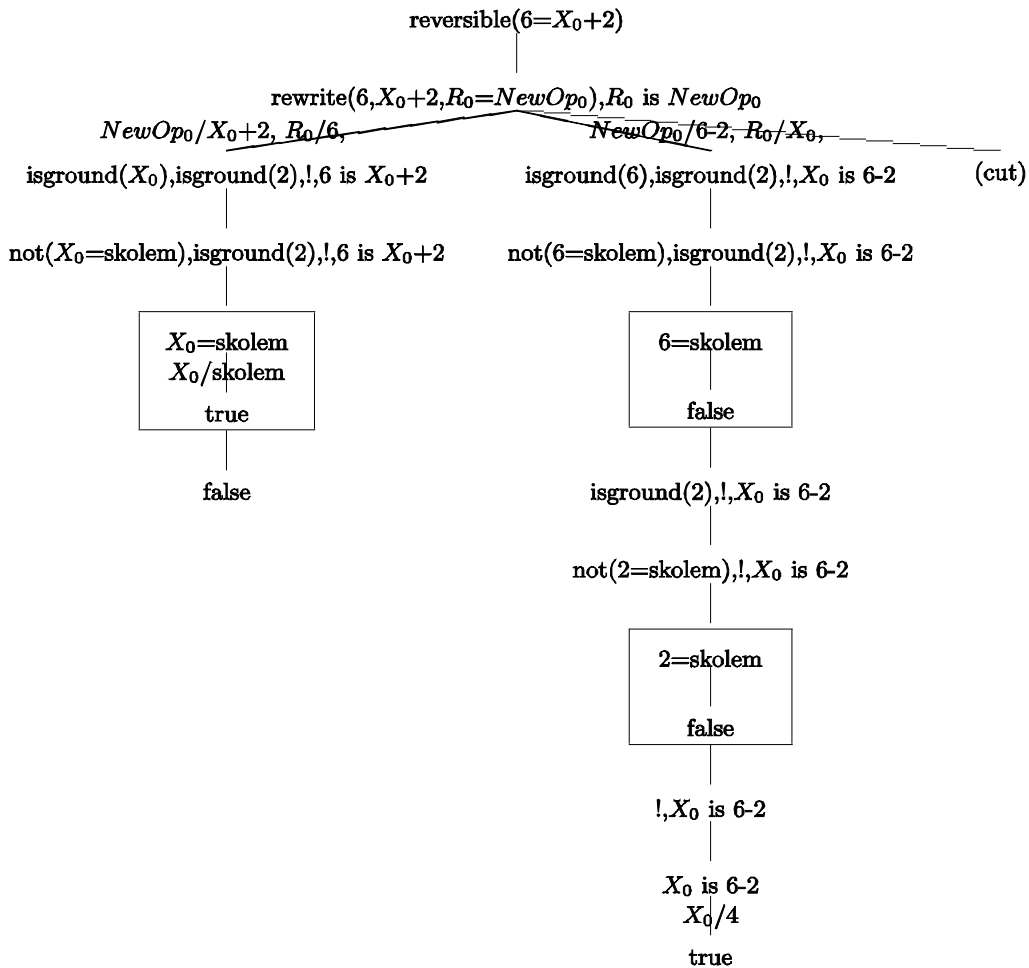
```

```

conta(A, [_|L], N) :-
    conta(A, L, N).

```

### Esercizio 2



Risposta calcolata: X/4.

### Esercizio 3

- 1)  $\forall X$  (corso(X), crediti(X,3)  $\rightarrow$  facile(X))
- 2)  $\forall X$  (corso(X), crediti(X,6)  $\rightarrow$  not facile(X))
- 3)  $\forall X$  (corso(X), facile(X)  $\rightarrow$   $\exists S$  (studente(S,X), felice(S)))
- 4)  $\forall X \forall S$  (corso(X), not facile(X), studente(S,X)  $\rightarrow$  not felice(S))
- 5) corso(analisi)

- 6) crediti(analisi,6)
- 7) corso(inglese)
- 8) crediti(inglese,3)
- 9) studente(luca,inglese)
- 10) studente(luca,analisi)

Goal: not felice(luca)).

Il quantificatore esistenziale deve essere eliminato. Siccome rientra nello scope di  $\forall X$ , si può procedere alla skolemizzazione:

$\forall X$  (corso(X), facile(X)  $\rightarrow$  studente(f(X), X), felice(f(X)))

- C1. not corso(X) or not crediti(X,3) or facile(X)
- C2. not corso(X) or not crediti(X,6) or not facile(X)
- C3. not corso(X) or not facile(X) or studente(f(X), X)
- C4. not corso(X) or not facile(X) or felice(f(X))
- C5. not corso(X) or facile(X) or not studente(S,X) or not felice(S)
- C6. corso(analisi)
- C7. crediti(analisi,6)
- C8. corso(inglese)
- C9. crediti(inglese,3)
- C10. studente(luca,inglese)
- C11. studente(luca,analisi)

notG: felice(luca)

Applico la risoluzione:

- C12: not G e : not corso(X) or facile(X) or not studente(luca,X)
- C13: C12 e : not corso(analisi) or facile(analisi)
- C14: C13 e : facile(analisi)
- C15: C14 e C2: not corso(analisi) or not crediti(analisi,6)
- C16: C15 e : not crediti(analisi,6)
- C17: C16 e : clausola vuota

#### Esercizio 4

Rappresentiamo lo stato con

- i tempi di percorrenza dei componenti della band che si trovano alla sinistra del ponte
- una freccia che indica da quale lato del ponte si trova la lampada

Come funzione euristica  $h'$  usiamo la seguente idea:

Possono spostarsi solo 2 persone alla volta, per cui raggruppiamo le persone sulla riva sinistra in gruppi da 2. Ordino i tempi di percorrenza dal più alto al più basso e metto

- nel primo gruppo i primi due
- nel secondo gruppo il terzo ed il quarto.

Ciascun gruppo si muoverà alla velocità del più lento, per cui per ogni gruppo prendo il massimo. La funzione  $h'$  è data dalla somma dei tempi di percorrenza di ciascun gruppo. In altre parole, il

tempo di percorrenza totale sarà almeno il tempo di percorrenza del più lento, più quello del terzo più lento (il secondo più lento potrebbe essersi mosso con il più lento, per cui il suo tempo non viene calcolato).

Nell'algoritmo A\* per i grafi si considerano cammini alternativi per arrivare ad uno stesso stato; in caso di cammini alternativi viene tenuto il costo minore. Rappresentiamo con una freccia piena il cammino migliore e con una freccia tratteggiata i cammini peggiori (che vengono quindi scartati). Rappresentiamo in rosso i nodi di fallimento (quelli con funzione euristica maggiore di 17).

