

FONDAMENTI DI INTELLIGENZA ARTIFICIALE – 1° parte (6 CFU)

20 Dicembre 2012 – Tempo a disposizione: 2 h – Risultato: 32/32 punti

Esercizio 1 (5 punti)

Si consideri la seguente base di conoscenza KB espressa in *logica dei predicati del primo ordine*:

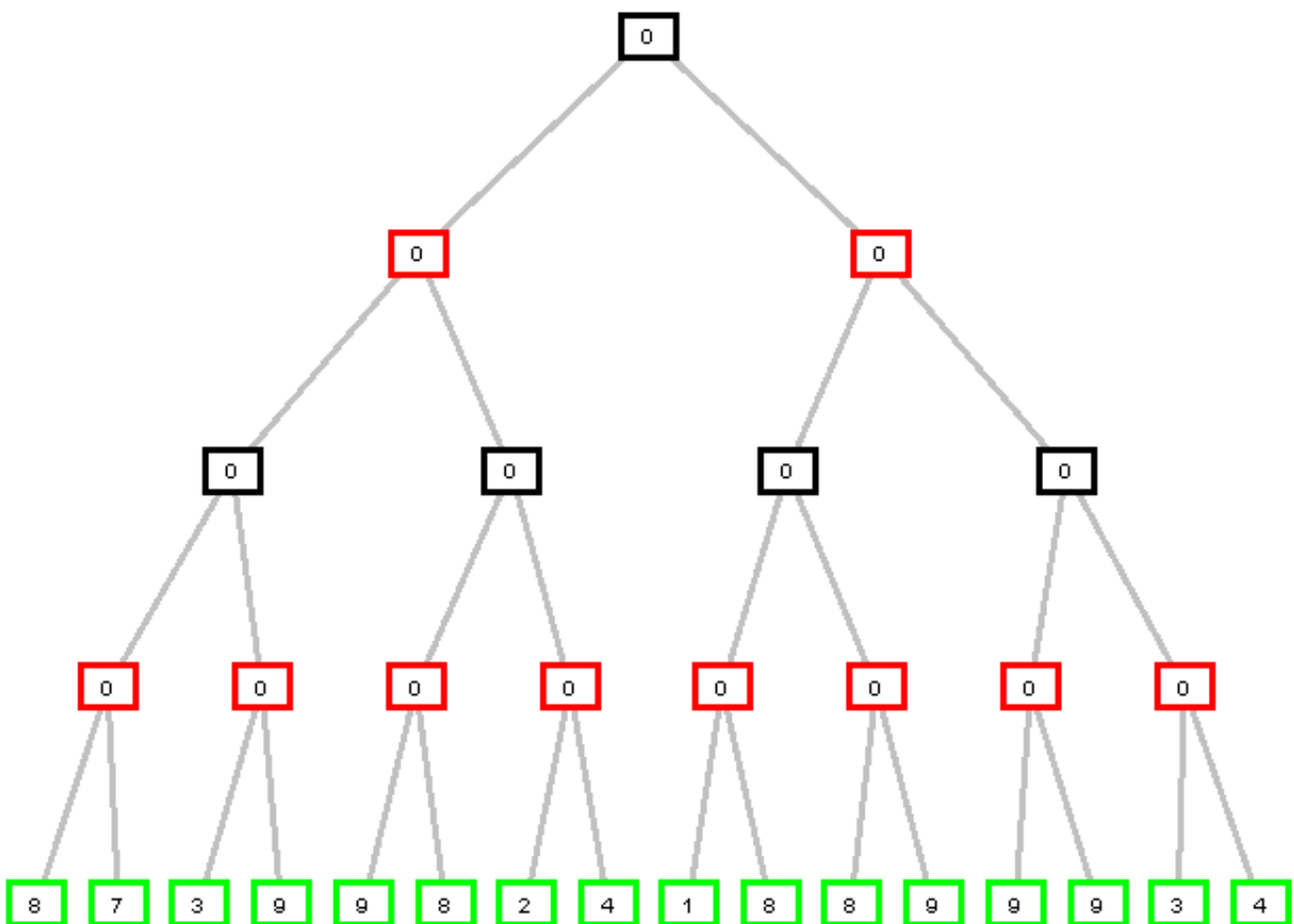
- a. $\forall x(\text{presente}(x) \wedge \neg \text{studente}(x) \Rightarrow \text{professore}(x))$
- b. $\forall x(\neg \text{fa}(x, \text{esame}) \Rightarrow \neg \text{studente}(x))$
- c. $\forall x(\text{fa}(\text{alice}, x) \Rightarrow \neg \text{fa}(\text{betta}, x))$
- d. $\text{fa}(\text{betta}, \text{esame})$
- e. $\text{presente}(\text{alice})$
- f. $\text{presente}(\text{betta})$
- g. $\text{presente}(\text{carlo})$

Si dimostri tramite risoluzione che KB implica logicamente la seguente formula α :

- α . $\exists x(\text{professore}(x) \wedge \neg \text{studente}(x) \wedge \text{presente}(x))$

Esercizio 2 (5 punti)

Si consideri il seguente albero di gioco in cui la valutazione dei nodi terminali è dal punto di vista del primo giocatore (che è *Max*). Si mostri come gli algoritmi *min-max* e *alfa-beta* risolvono il problema.



Esercizio 3 (6 punti)

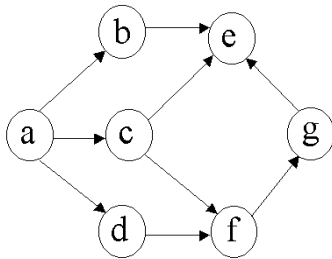
Dato il seguente programma Prolog:

```
prop([H|T], Ni, No) :- H >= Ni, Nt is Ni + 1, prop(T, Nt, No).  
prop([H|T], N, N) :- H >= N.
```

Si mostri l'albero di derivazione SLD-NF relativo al seguente goal:

```
?- not(prop([3,1],1,2)).
```

Esercizio 4 (5 punti)



Si consideri la situazione rappresentata nella figura a sinistra, dove il grafo disegnato è rappresentato in *Prolog* dai seguenti fatti:

```
edge(a,b). edge(a,d). edge(c,e). edge(d,f). edge(g,e).  
edge(a,c). edge(b,e). edge(c,f). edge(f,g).
```

(ogni fatto del tipo $\text{edge}(N1, N2)$ è vero quando esiste un arco orientato da $N1$ a $N2$).

Si scriva un programma *Prolog* $\text{path}(X, Z, L)$ che determina se in un grafo così rappresentato esiste una strada che connette i nodi X e Z e che ne determina la lunghezza L .

Esercizio 5 (7 punti)

Si consideri il problema *8-puzzle* con stato di partenza e stato obiettivo mostrati di seguito.

2	8	3
1	6	4
7		5

Stato di partenza

1	2	3
8		4
7	6	5

Stato obiettivo

Dimostrare che è possibile arrivare alla soluzione con l'*algoritmo A**, usando come euristica h il numero di tessere nella posizione sbagliata, ed avendo cura di evitare gli stati ripetuti.

Si usi $g(n) =$ profondità del nodo n dell'albero di ricerca (lo stato iniziale è a profondità zero).

Esercizio 6 (4 punti)

Si descrivano le tecniche di consistenza applicabili a una rete CSP, si mostri un esempio e se ne discuta poi l'applicazione sull'esempio mostrato.

VOTO:

- Esame da 6 CFU, il voto è determinato da questa I parte
- Esame da 9 CFU, è la media pesata della 1° parte (che vale 2/3) e della 2° (che vale 1/3) ovvero il voto finale è dato da: $2x(\text{votoIparte} + \text{votoIIparte})/3$ e varia quindi da 0 ad un massimo di 32 (equivalente alla lode).

FONDAMENTI DI INTELLIGENZA ARTIFICIALE – 2° parte (3 CFU)

20 Dicembre 2012 – Tempo a disposizione: 30(+15) min – Risultato: 16/16 punti

Esercizio 7 (4 punti)

Si descriva il "ciclo evolutivo" nelle tecniche di computazione evolutiva e si illustrino gli operatori genetici tramite un'istanza di ciascuno di essi.

Esercizio 8 (9 punti)

Si definisca un predicato `provable(Goal, Defs)` che risulta vero se il `Goal` può essere dimostrato utilizzando la lista di clausole `Defs`.

Le clausole sono rappresentate quali elementi della lista `Defs` come strutture del tipo `clausola(Head, Body)` e sono solo *ground*.

Possibili query sono ad esempio:

```
?-provable(g(X), [clausola(g(1),true), clausola(g(2),(a(1),b(2)))]).
```

che restituisce vero (e due soluzioni in backtracking), oppure

```
?-provable((g(X),r(X)), [clausola(g(1),true), clausola(g(2),(a(1),b(2)))]).
```

che restituisce false invece. In entrambe le query, la lista data al predicato `provable/2` come secondo argomento rappresenta il programma *Prolog*:

```
g(1).
```

```
g(2):- a(1),b(2).
```

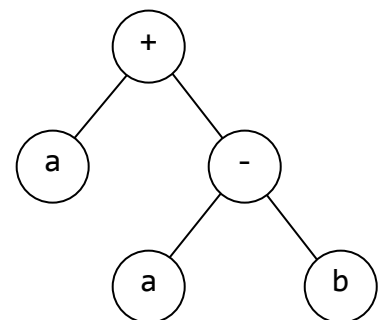
Esercizio 9 (3 punti – ulteriori 15 min di tempo)

Nota: lo deve svolgere solamente chi non ha partecipato alle due lezioni/esercitazioni (con valutazione) su Prolog e grammatiche.

Data la seguente grammatica:

$$G = (V_n, V_t, P, S)$$
$$V_n = \{E, T\}$$
$$V_t = \{+, -, a, b\}$$
$$P = \{E ::= T + E | T - E | T, \quad T ::= a | b\}$$
$$S = E$$

si scrivano le corrispondenti clausole in *versione DCG estesa* che verificano la correttezza di una frase e producono l'albero sintattico astratto. Ad esempio, per il *goal* `a+a-b` (il cui albero sintattico è riportato di seguito) è rappresentato dal termine: `piu(a, meno(a, b))`.



VOTO:

- *Esame da 9 CFU, è la media pesata della 1° parte (che vale 2/3) e della 2° (che vale 1/3) ovvero il voto finale è dato da: $2x(\text{votoIparte} + \text{votoIIparte})/3$ e varia quindi da 0 ad un massimo di 32 (equivalente alla lode).*

FONDAMENTI DI INTELLIGENZA ARTIFICIALE – 1° parte

20 Dicembre 2012 – Soluzioni

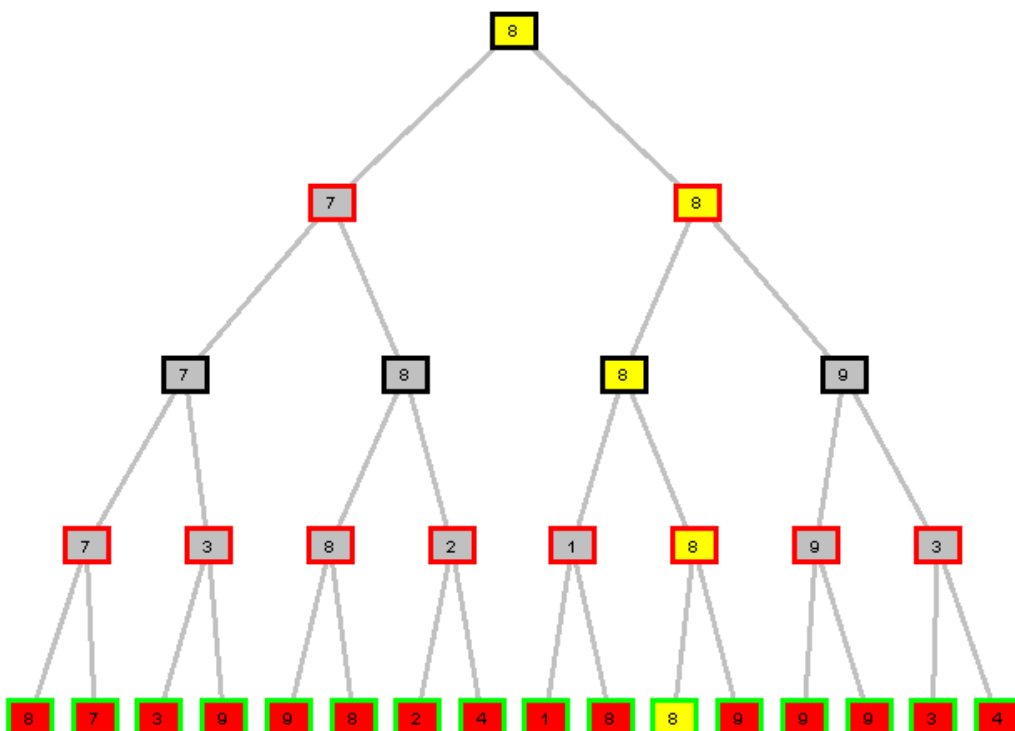
Esercizio 1

Trasformo KB e $\neg\alpha$ in CNF e cerco la clausola vuota:

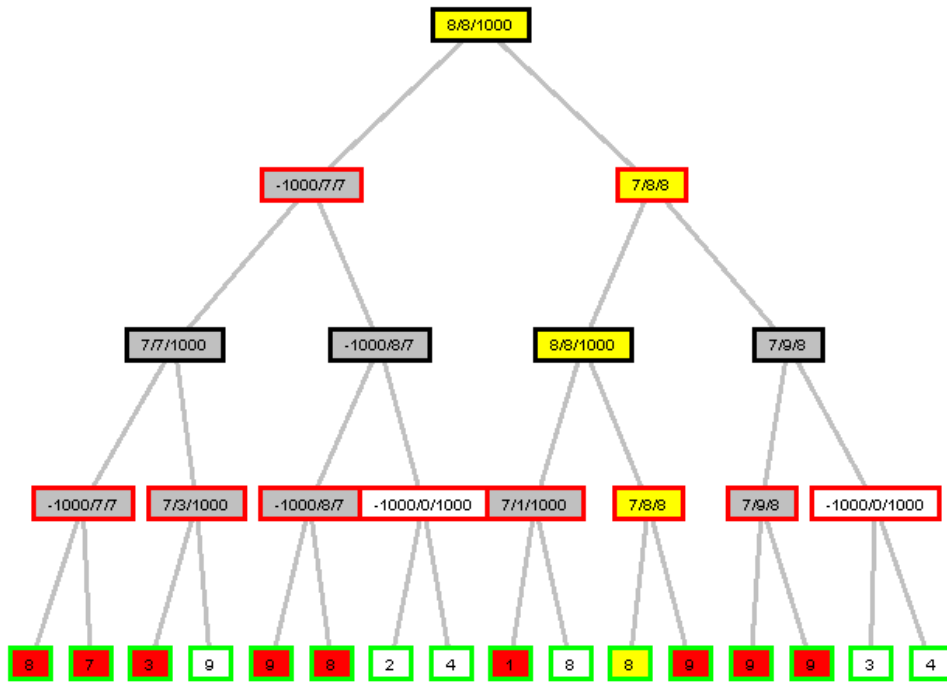
1. (= a) $\neg\text{presente}(X) \vee \text{studente}(X) \vee \text{professore}(X)$
2. (= b) $\text{fa}(X, \text{esame}) \vee \neg\text{studente}(X)$
3. (= c) $\neg\text{fa}(\text{alice}, X) \vee \neg\text{fa}(\text{betta}, X)$
4. (= d) $\text{fa}(\text{betta}, \text{esame})$
5. (= e) $\text{presente}(\text{alice})$
6. (= f) $\text{presente}(\text{betta})$
7. (= g) $\text{presente}(\text{carlo})$
8. (= $\neg\alpha$) $\neg\text{presente}(X) \vee \text{studente}(X) \vee \neg\text{professore}(X)$
9. (= $1 \cup 5$) $\text{studente}(\text{alice}) \vee \text{professore}(\text{alice})$ {X/alice}
10. (= $5 \cup 8$) $\text{studente}(\text{alice}) \vee \neg\text{professore}(\text{alice})$ {X/alice}
11. (= $9 \cup 10$) $\text{studente}(\text{alice})$
12. (= $3 \cup 4$) $\neg\text{fa}(\text{alice}, \text{esame})$ {X/esame}
13. (= $2 \cup 12$) $\neg\text{studente}(\text{alice})$ {X/alice}
14. (= $11 \cup 13$) \square

Esercizio 2

min-max:



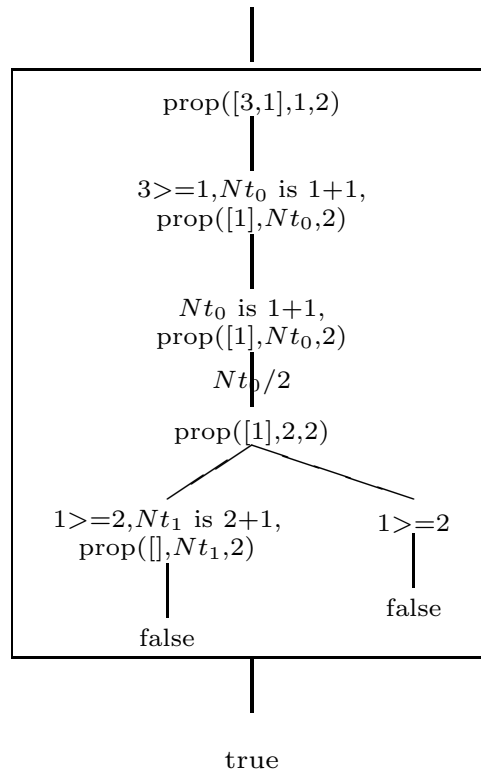
alfa-beta:



I nodi non colorati sono quelli che vengono tagliati nell'algoritmo alfa-beta.

Esercizio 3

Albero SLD-NF:



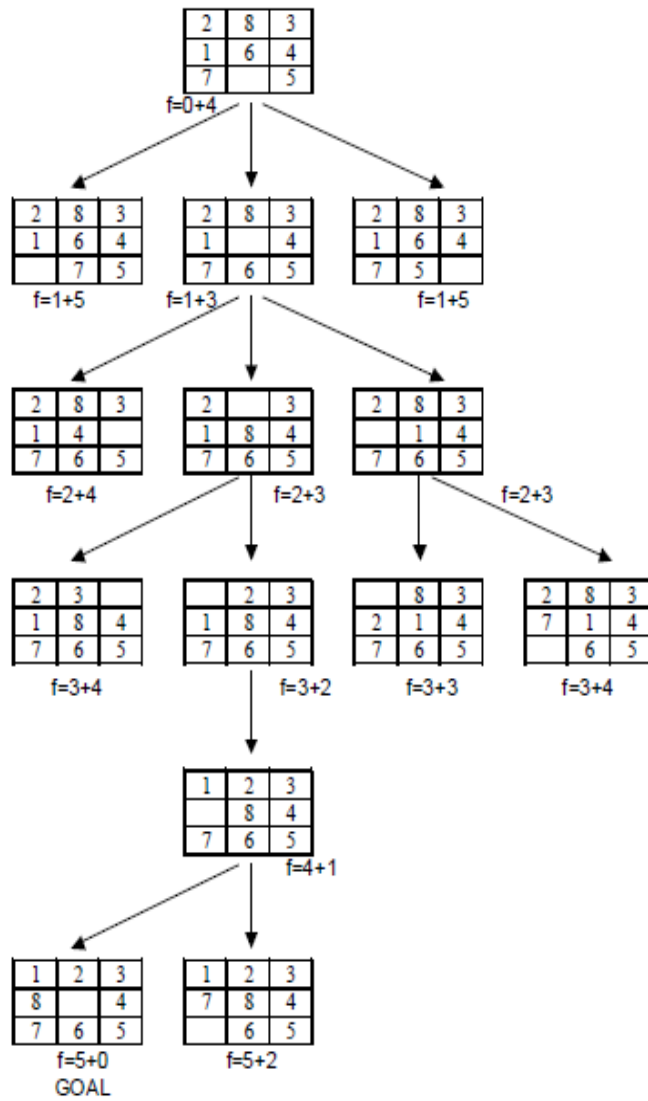
Esercizio 4

path(X,X,0).

path(X,Z,L1) :- edge(X,Y),

path(Y,Z,L2), L1 is L2+1.

Esercizio 5



Esercizio 6

Vedi slides del corso.

FONDAMENTI DI INTELLIGENZA ARTIFICIALE – 2° parte

20 Dicembre 2012 – Soluzioni

Esercizio 7

Vedi slides del corso.

Esercizio 8

```
provable(true, _) :- !.  
provable((G1,G2), Defs) :- !,  
    provable(G1, Defs), provable(G2, Defs).  
provable(Goal, Defs) :- member(clausola(Goal,Body), Defs),  
    provable(Body, Defs).
```

Esercizio 9

```
e(piu(X, Y)) --> t(X), [+], e(Y).  
e(meno(X, Y)) --> t(X), [-], e(Y).  
e(X) --> t(X).  
t(a) --> [a].  
t(a) --> [b].
```