FONDAMENTI DI INTELLIGENZA ARTIFICIALE

19 Giugno 2009 – Tempo a disposizione 2h – Risultato 32/32 punti

Esercizio 1 (punti 7)

Si modellino in logica dei predicati del I ordine le seguenti frasi (si utilizzino i predicati unari scolaro/1, ins/1, e il predicato binario risolve/2):

Qualsiasi scolaro della primaria è in grado di risolvere alcune operazioni e non è in grado di risolvere alcune operazioni.

Alcuni insegnanti sono in grado di risolvere qualsiasi operazione.

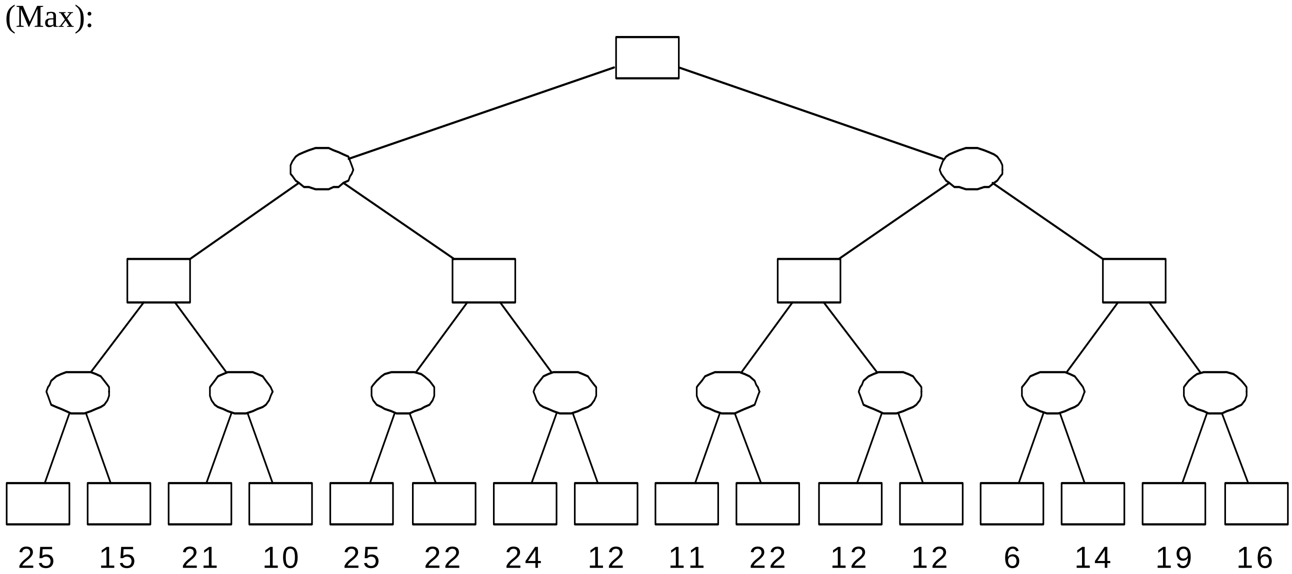
a) Le si trasformi in logica a clausole e si dimostri, applicando il principio di metodo di risoluzione, che:

Alcuni insegnanti non sono scolari.

b) E' possibile scrivere la teoria a clausole come programma logico? Motivare la risposta data.

Esercizio 2 (punti 5)

Si consideri il seguente albero di gioco, dove i punteggi sono dal punto di vista del primo giocatore (Max):



Si mostri come l'algoritmo min-max risolve il problema. Si mostrino poi i tagli alfa-beta.

Esercizio 3 (punti 6)

Sia dato il seguente programma Prolog:

```
intero(0).
intero(N):-intero(K), N is K+1.
giu19(X):-intero(X), X2 is X*X, not(X=X2), !.
```

Si mostri l'albero di derivazione SLDNF relativo al goal `?- giu19(X).`

Esercizio 4 (punti 6)

Siano dati due numeri S e G e un insieme di numeri P, tutti di tre cifre, compresi tra 100 e 999, L'obiettivo è trasformare S in G. Le azioni possibili per trasformare un numero in un altro consistono nell'incrementare o decrementare di una unità una delle sue cifre, rispettando i seguenti vincoli:

- non è consentito incrementare la cifra 9 o decrementare la cifra 0;
- non è consentito trasformare un numero in uno appartenente all'insieme P;

- non è consentito modificare la stessa cifra in due mosse successive.

Con questi vincoli esistono al più 6 mosse possibili dallo stato di partenza e al più quattro da qualsiasi altro stato. Ogni azione ha un costo pari a 1.
Si risolva il problema con l'algoritmo di ricerca A*, visualizzando l'albero di ricerca, nel caso in cui:

S = 567, G= 777 P={666, 667}

Si usi la seguente euristica: la distanza tra un qualsiasi numero e G è stimata pari alla somma delle differenze in valore assoluto tra le cifre corrispondenti. Tale euristica è ammissibile? Nei rami dell'albero si indichi +1 o -1 a seconda dell'operazione eseguita e si sottolinei nei nodi la cifra che è stata modificata.

Suggerimento: in caso di più possibili nodi da espandere, si scelga tra quelli con il valore di *g* più alto. Nel caso di nodi con lo stesso valore di f, si espanda per primo il nodo corrispondente al valore numerico (stato) maggiore.

Esercizio 5 (punti 5)
Si scriva un programma Prolog per un predicato `potenze(I,J,P)` che, dati i tre argomenti in ingresso, è vero se `P` è la lista (ordinata) delle potenze di `I` con esponente compreso tra `1` e `J` (I^1 e I^J incluse). Si supponga che `I`, `J` e `P` siano sempre ground e che `I` e `J` siano entrambi positivi.
Esempi:

```
?-potenze(2,4,[2,4,8,16]).
yes
?-potenze(2,3,[4,8,2]).
no
?-potenze(2,3,[2,4,8]).
yes
```

Esercizio 6 (punti 3)
Si discutano gli algoritmi di consistenza di una rete CSP e in particolare si descriva (in pseudo-codice) l'algoritmo di arc-consistenza.

SOLUZIONE:

Esercizio 1

a1. Formalizzazione:

1. $\forall x\ Scolaro(x) \Rightarrow \exists\ y\ Risolve(x, y) \wedge \exists\ z\ \neg Risolve(x, z)$
2. $\exists\ x\ Ins(x) \wedge \forall y\ Risolve(x, y)$
3. Da dimostrare: $\exists\ x\ Ins(x) \wedge \neg\ Scolaro(x)$

a2. Trasformazione in forma a clausole:

1. $\forall x\ Scolaro(x) \Rightarrow \exists\ y\ Risolve(x, y) \wedge \exists\ z\ \neg Risolve(x, z)$

$\forall x\ \neg Scolaro(x) \vee (\exists\ y\ Risolve(x, y) \wedge \exists\ z\ \neg Risolve(x, z))$ [*eliminazione* $\Rightarrow$]

$\forall x\ \neg Scolaro(x) \vee (Risolve(x, p1(x)) \wedge \neg Risolve(x, p2(x)))$ [*skolemizzazione*]

$\neg Scolaro(x) \vee (Risolve(x, p1(x)) \wedge \neg Risolve(x, p2(x)))$ [*eliminazione* $\forall$]

$(\neg Scolaro(x) \vee Risolve(x, p1(x))) \wedge (\neg Scolaro(x) \vee \neg Risolve(x, p2(x)))$

1.1 $\{\neg Scolaro(x1), Risolve(x1, p1(x1))\}$

1.2 $\{\neg Scolaro(x2), \neg Risolve(x2, p2(x2))\}$

2. $\exists\ x\ Ins(x) \wedge \forall y\ Risolve(x, y)$

$Ins(i) \wedge \forall y\ Risolve(i, y)$ [*skolemizzazione*]

$\{Ins(i)\}$

$\{\ Risolve(i, y)\}$

Goal negato: $\neg\exists\ x\ Ins(x) \wedge \neg\ Scolaro(x)$

$\forall\ x\ \neg Ins(x) \vee Scolaro(x)$

$\{\neg Ins(z), Scolaro(z)\}$

a3. Dimostrazione per refutazione:

$\{\neg Scolaro(x2), \neg Risolve(x2, p2(x2))\}$ $\{Risolve(i, y)\}$ $\{\neg Ins(z), Scolaro(z)\}$ $\{Ins(i)\}$

$\{\neg Scolaro(i)\}$

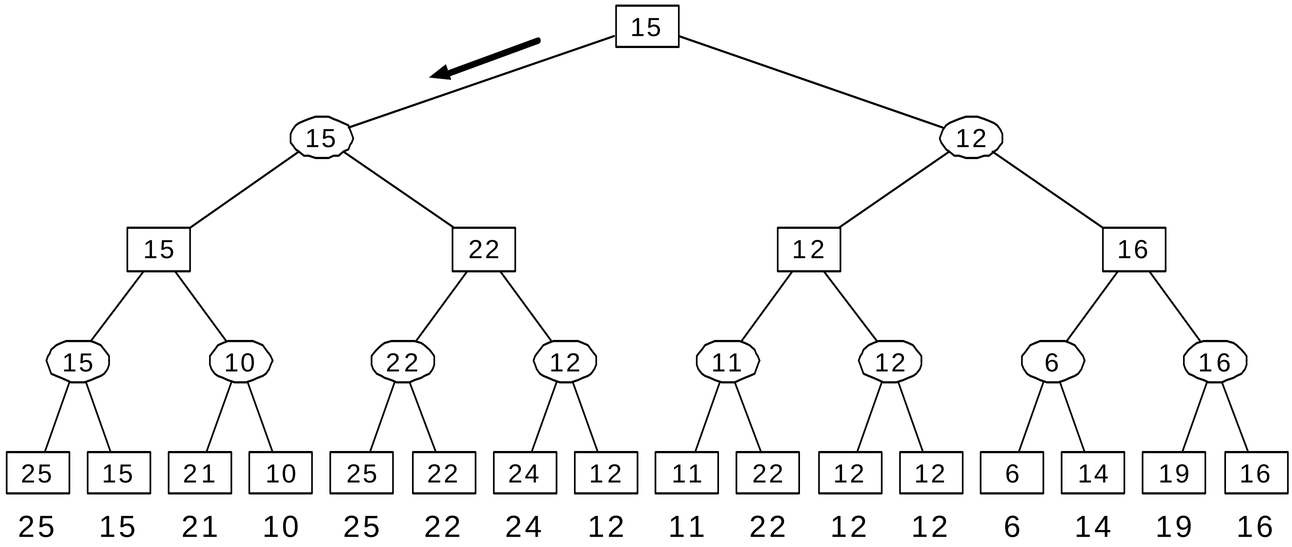
$\{\neg Ins(i)\}$

$\{\ \}$

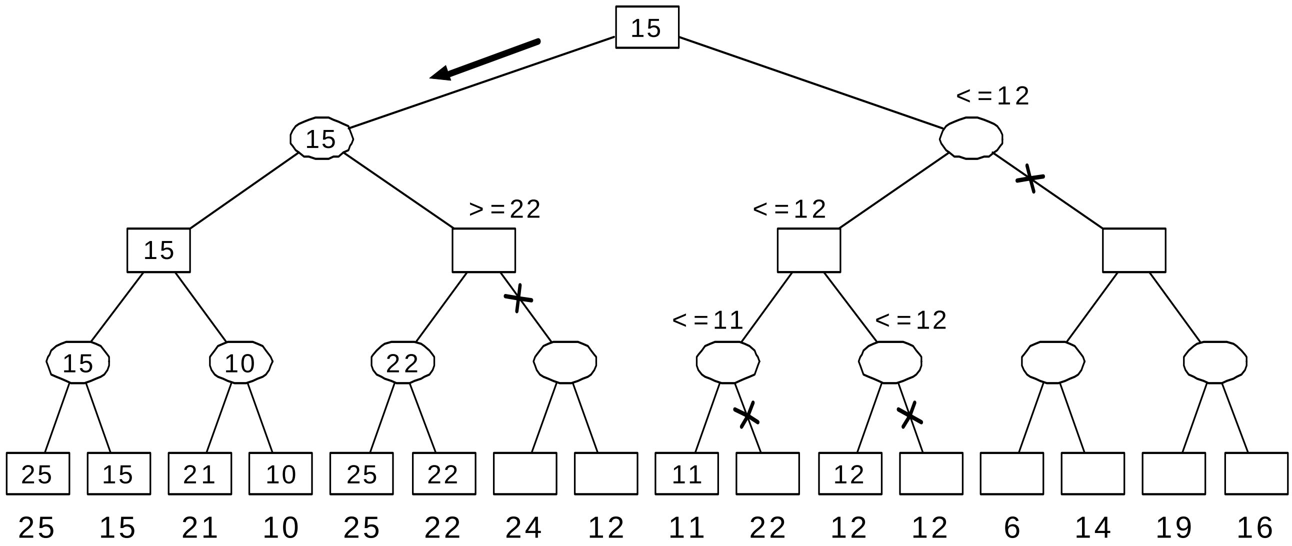
b. Non è possibile rendere come programma logico la KB iniziale in quanto la prima formula, trasformata in forma a clausole, non è una clausola Horn **definita** (non ci sono letterali positivi).

Esercizio 2

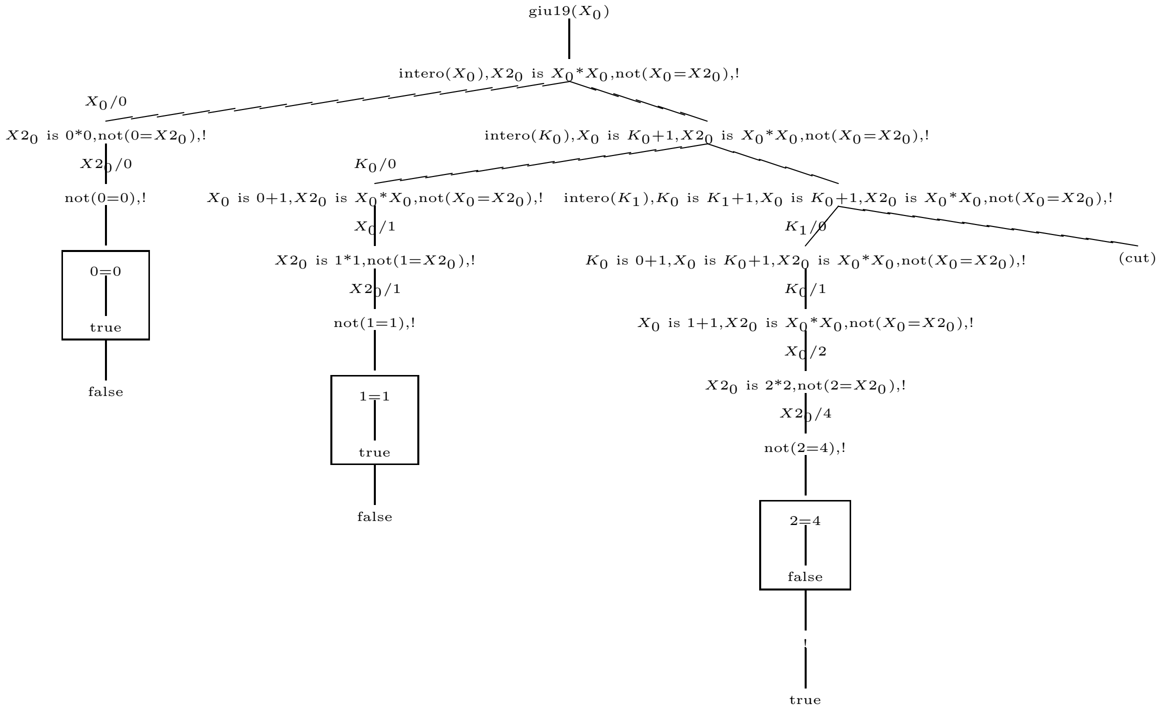
Min-max:



Alfa-beta:

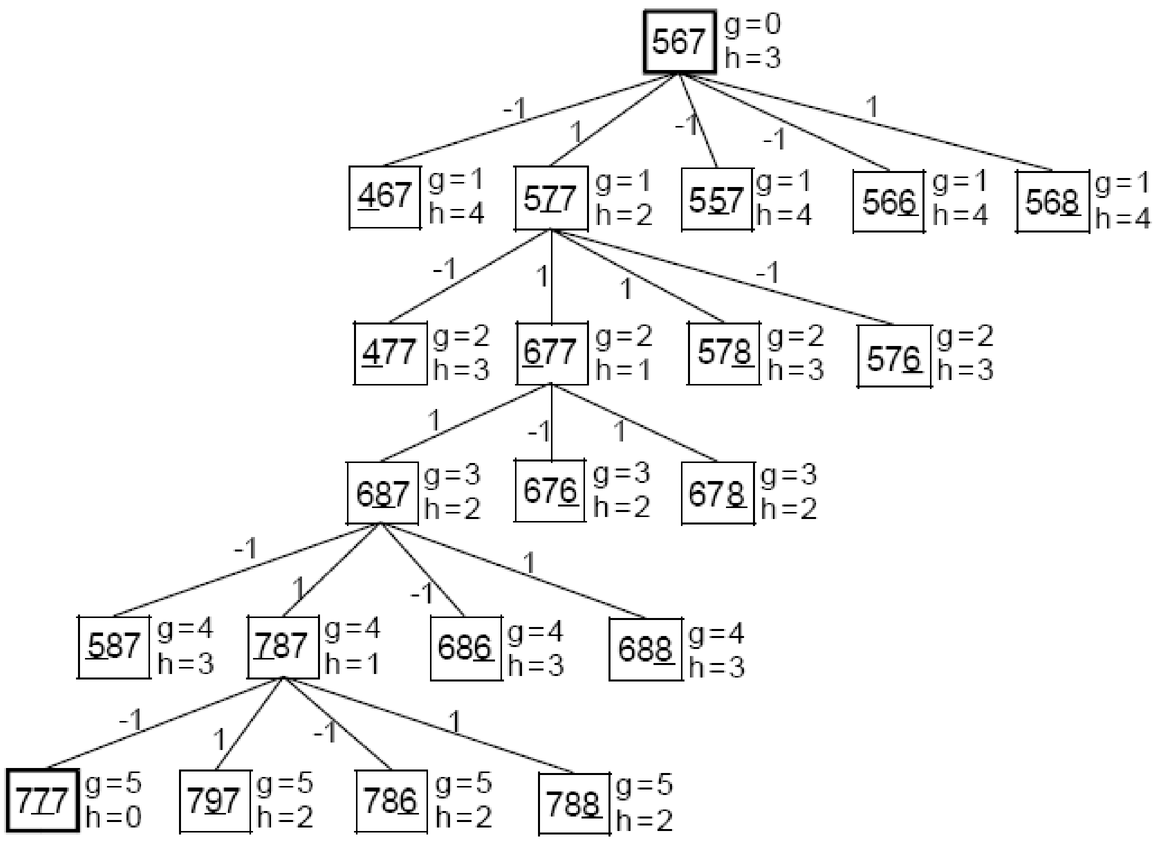


Esercizio 3



Esercizio 4

In ogni nodo dell'albero di ricerca è sottolineata la cifra che è stata modificata. Su ogni arco è indicata l'azione eseguita su tale cifra: 1 indica un incremento, -1 un decremento. Tra i tre nodi a profondità 3, aventi tutti lo stesso valore di f = g+h, si è scelto arbitrariamente di espandere quello più a sinistra.



Esercizio 5

```
/* potenze(I,J,Ks) e' vero se Ks e' la lista ordinata       */
/* di potenze di I comprese tra I alla 1 e I alla J         */
/* incluse  */

potenze(I,J,Ks):- potenze (I,J,1,Ks).

potenze (I,J,J,[T]):- power(I,J,T),!.
potenze (I,J,N,[T|C]):- power(I,N,T), N1 is N+1, potenze(I,J,N1,C).

power(I,0,1):-!.
power(I,1,I):-!.
power(I,J,T):- J1 is J-1, power(I,J1,T1), T is T1*I.
```