COMPITO DI INTELLIGENZA ARTIFICIALE (v.o.) – PARTE I
FONDAMENTI DI INTELLIGENZA ARTIFICIALE

23 Giugno 2004 (Tempo a disposizione 2h; su 32 punti)

Esercizio 1: (punti 7)
Si supponga di sapere che "i cavalli sono animali". Operando per contraddizione e utilizzando il principio di risoluzione, si dimostri che la formula "se esiste la testa di un cavallo allora esiste la testa di un animale" è un teorema. A tale scopo, si utilizzino i predicati: head_of(H,X), horse(X), animal(X). Si proceda negando la formula da dimostrare, trasformando in clausole e operando per contraddizione applicando il principio di risoluzione.

Esercizio 2 (punti 6)
Si scriva un programma in Prolog che data una lista di liste, ordinata in base alla lunghezza delle liste componenti, inserisca in modo ordinato una nuova lista secondo la sua lunghezza. Si definiscano tutti i predicati per la soluzione del problema.
Esempio:

```
?- insertOL([1,a, pippo], [[], [a,a],  [t, 7, 1, pp]], X).
Yes X= [[], [a,a],  [1,a, pippo], [t, 7, 1, pp]]
```

Esercizio 3 (punti 6)
Si consideri il seguente programma Prolog:

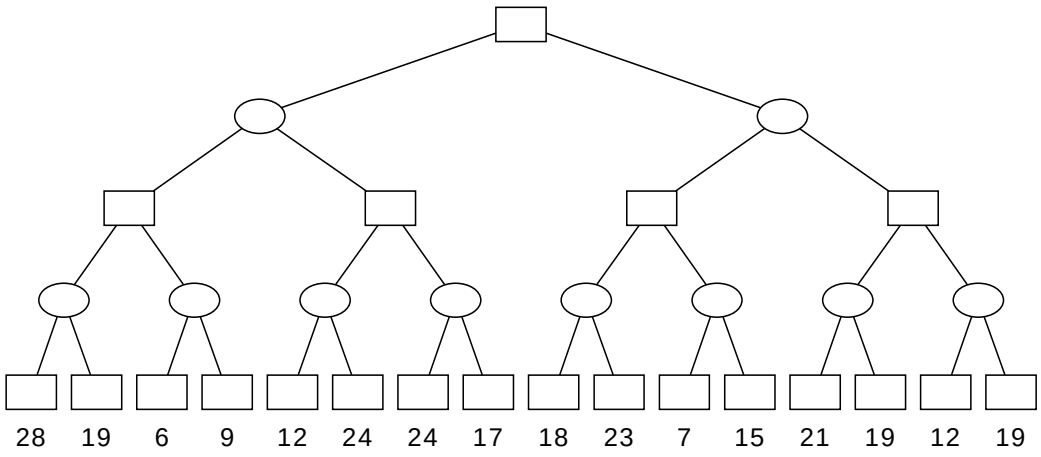
```
q([H|T]):- not p(H,T), q(T).
q([]).

p(H,T) :- not r(H).
p(H,T) :- r(T).

r(a).
```

Si mostrino gli alberi SLDNF relativi alle query `?-q([X])` e `?-q([b])` e si dica qual è la risposta calcolata in ciascuno dei casi. Quale dei due risultati è scorretto e perché?

Esercizio 4 (punti 5)
Si consideri il seguente albero di gioco:



Si mostri come viene esplorato l'albero dagli algoritmi min-max e alfa-beta supponendo che il primo giocatore sia Max.

Esercizio 5 (punti 6)
Si descrivano e confrontino i differenti gradi di consistenza che si possono ottenere in un grafo rappresentante un CSP, mostrando opportuni esempi.

SOLUZIONE

Esercizio 1:

a. Si traslano premessa e conclusione nel linguaggio della logica del I ordine:

Premessa: horse(X) => animal(x)
Conclusione: [exists X, exists H (head_of(H,X) ^ horse(X))] => exists T, exists Y (head_of(T,Y) ^ animal(Y))

b. Si nega la conclusione, si converte la premessa e la conclusione negate in forma normale congiuntiva:

Premessa: ~horse(X) v animal(X)
Conclusione:

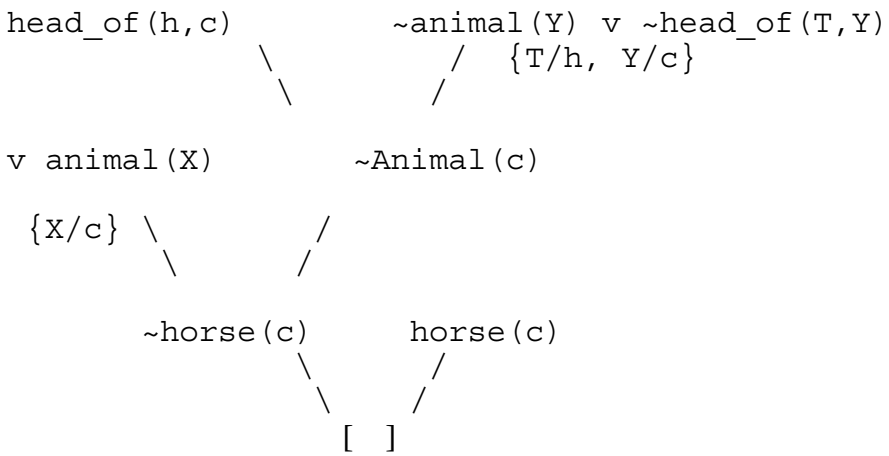
~{[exists X,exists H (head_of(H,X) ^ horse(X))] => exists T,exists Y (head_of(T,Y) ^ animal(Y))}
~ { ~[exists X, exists H (head_of(H,X) ^ horse(X))] v [exists T, exists Y (head_of(T,Y) ^ animal(Y))] }
~ { forall X,H exists T, exists Y ~[horse(X) ^ head_of(H,X)] v [animal(Y) ^ head_of(T,Y)] }
exists X, exists H, forall Y, forall T {[horse(X) ^ head_of(H,X)] ^ ~[animal(Y) ^ head_of(T,Y)] }
exists X, exists H, forall Y, forall T horse(X) ^ head_of(H,X) ^ ~[animal(Y) ^ head_of(T,Y)]

Skolemizzazione e eliminazione dei quantificatori:
horse(c) ^ head_of(h,c) ^ ~[animal(Y) ^ head_of(T,Y)]

Applicando DeMorgan's:
horse(c) ^ head_of(h,c) ^ (~animal(Y) v ~head_of(T,Y))

c. Si utilizza la risoluzione per dimostrare che la conclusione segue dalla premessa:

horse(c)
head_of(h,c)
~animal(Y) v ~head_of(T,Y)
~horse(X) v animal(X)



Quindi la testa di un cavallo è la testa di un animale.

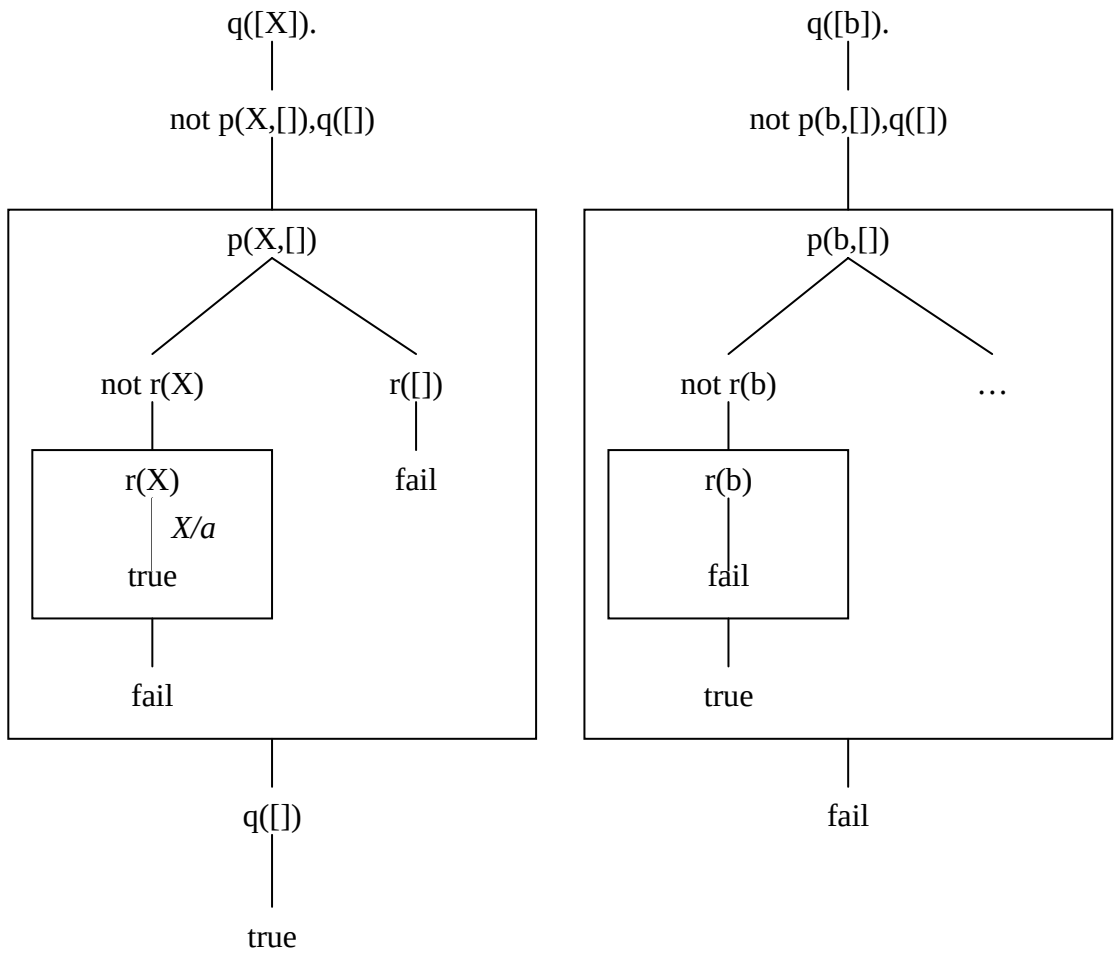
Esercizio 2

```
insertOL(X, [], [X]).
insertOL(X, [Y|Ys], [Y|Zs]):-
        lung(X,N),
lung(Y,N1), N>N1, !,
        insertOL(X,Ys, Zs).

insertOL(X, Zs, [X|Zs]).

lung([],0).
lung([A|B],N):- lung(B,N1),N is N1 +1.
```

Esercizio 3

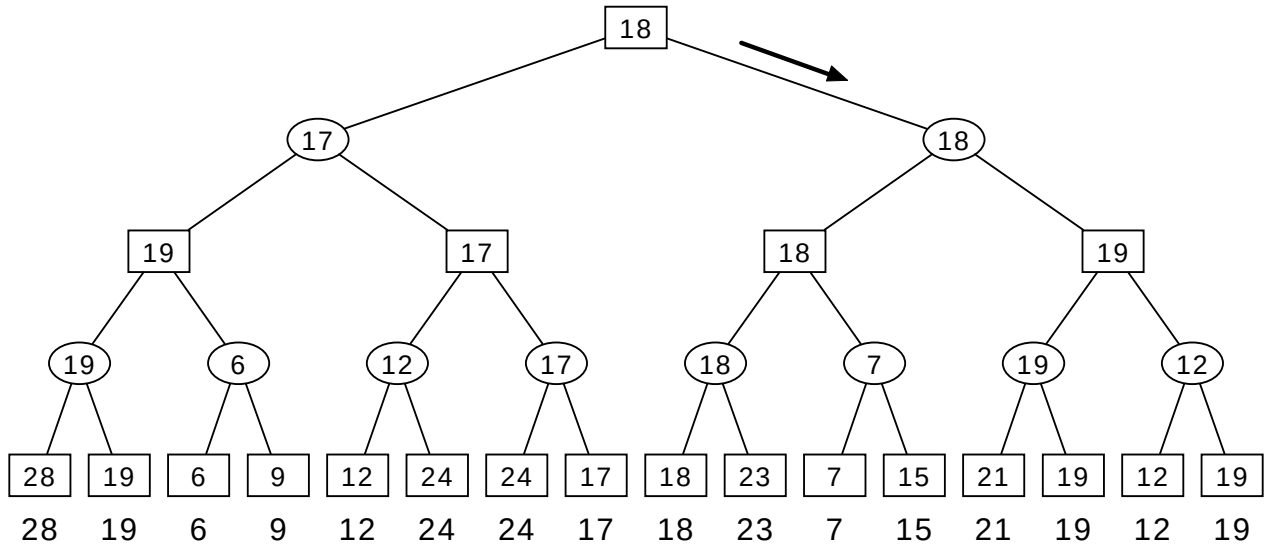
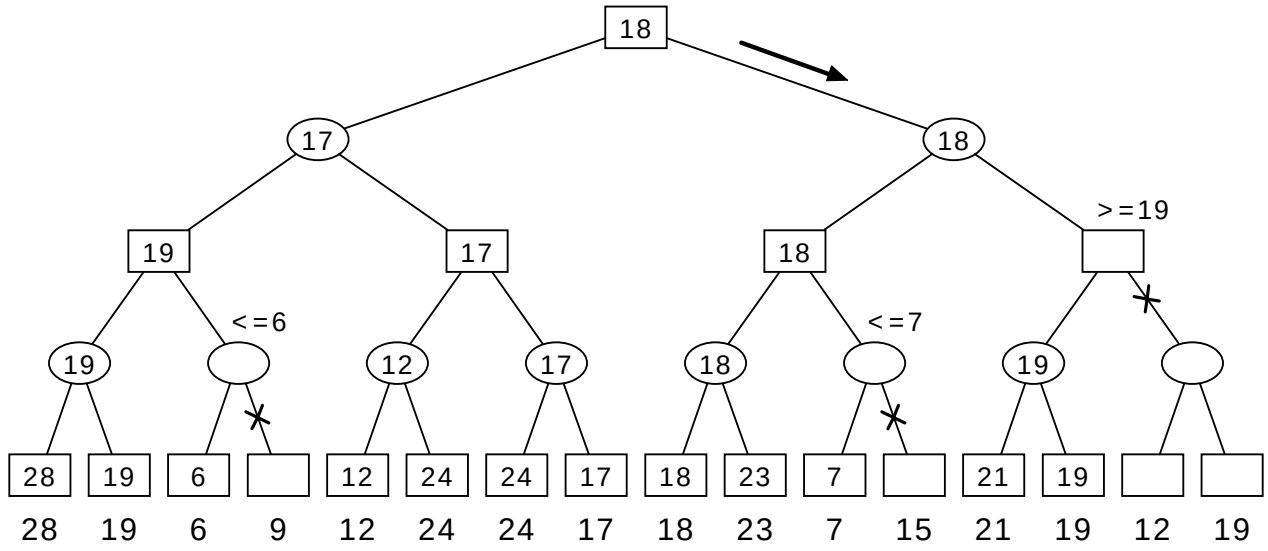


La risposta calcolata, nel primo caso, è la sostituzione vuota (non viene dato nessun binding per la variabile nel goal *X*). Nel secondo caso il goal fallisce, quindi nuovamente non c'è risposta calcolata.

Nel secondo caso il goal è un caso più specifico rispetto al primo, quindi, evidentemente, se fossero entrambi corretti, il successo del goal p(X) implicherebbe il successo del goal p(b). In questo caso, l'errore sta nel primo caso: siccome c'è un goal negato non ground, il risultato risulta scorretto.

Esercizio 4

Min-max:



Alfa-Beta: