

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

17 Marzo 2005 (Tempo a disposizione 2h; su 32 punti)

Esercizio 1 (punti 7)

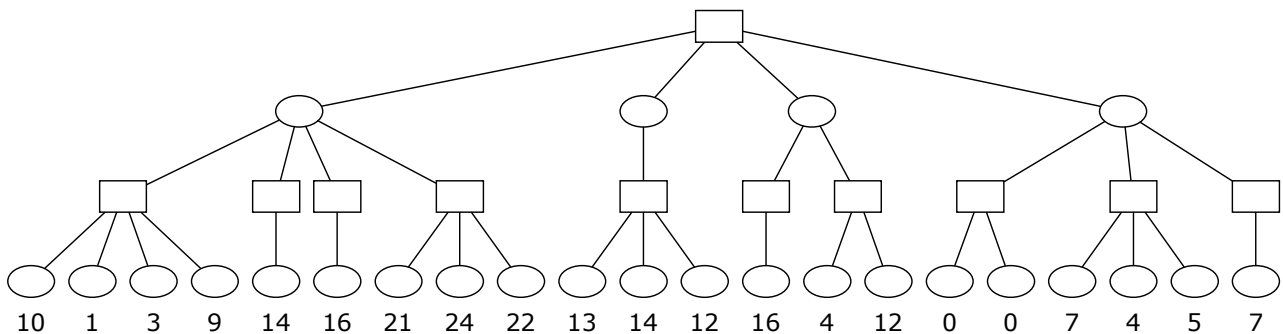
Si traducano le seguenti frasi nella logica dei predicati del primo ordine, poi in forma a clausole:

- Ogni autoveicolo è diesel o a benzina
- Se un autoveicolo è diesel, inquina molto
- Chi inquina molto, paga una soprattassa
- RenaultScenic TDI è un autoveicolo diesel

Si usi poi il principio di risoluzione per dimostrare che esiste un autoveicolo che paga una soprattassa.

Esercizio 2 (punti 8)

Dato il seguente albero di ricerca per un gioco a due giocatori, si mostri quale mossa selezionerà MAX, secondo l'algoritmo MIN-MAX. Si mostri inoltre come può essere ridotto l'albero applicando i tagli alfa-beta.



Esercizio 3 (punti 6)

Si scriva un programma Prolog *pari(Lin1, Lout)* che data una lista *Lin1* produca una lista in uscita *Lout* contenente gli elementi di *Lin1* che si trovano in posizione pari (seconda, quarta, sesta, etc). Si scrivano esplicitamente tutti i predicati Prolog usati nella soluzione.

Esempi:

```
?-pari([a, s, r, k, t, s], X)
```

restituisce

```
yes    X=[s,k,s]
```

Esercizio 4 (punti 7)

Sia dato il seguente programma Prolog:

```
liste([],_,[]) :- !.  
liste([H],L2,[H]) :- not member(H,L2),!.  
liste([H],L2,[]) :- !.  
liste([H|T],L2,[H|T1]) :- not member(H,L2),!, liste(T,L2,T1).  
liste([H|T],L2, T1) :- liste(T,L2,T1).
```

Disegnare l'albero SLDNF relativo al goal:

```
?- liste([a,b,a,d],[d],X).
```

indicando i tagli dovuti al predicato "cut" e anche la sostituzione di risposta calcolata.

Esercizio 5 (punti 4)

Si introduca l'algoritmo di unificazione. Si spieghi poi che cosa è l'occur check e che cosa implica la sua assenza nell'interprete del linguaggio Prolog.

SOLUZIONE

Esercizio 1

In logica:

$\forall X \text{ auto}(X) \rightarrow \text{diesel}(X) \text{ ex-or } \text{benzina}(X)$

$\forall X \text{ auto}(X) \text{ and } \text{diesel}(X) \rightarrow \text{inquinamolto}(X)$

$\forall X \text{ inquinamolto}(X) \rightarrow \text{sopratassa}(X)$

$\text{auto}(\text{rs_tdi})$

$\text{diesel}(\text{rs_tdi})$

Query: $\exists Y \text{ auto}(Y) \text{ and } \text{sopratassa}(Y)$

Trasformazione in clausole:

C1: not auto(X) or diesel(X) or benzina(X)

C2: not auto(X) or not diesel(X) or not benzina(X)

C3: not auto(X) or not diesel(X) or inquinamolto(X)

C4: not inquinamolto(X) or sopratassa(X)

C5: auto(rs_tdi)

C6: diesel(rs_tdi)

Goal: not auto(Y) or not sopratassa(Y)

Applicando il Principio di Risoluzione:

C7 (da C3 e Goal): not auto(Y) or not inquinamolto(Y)

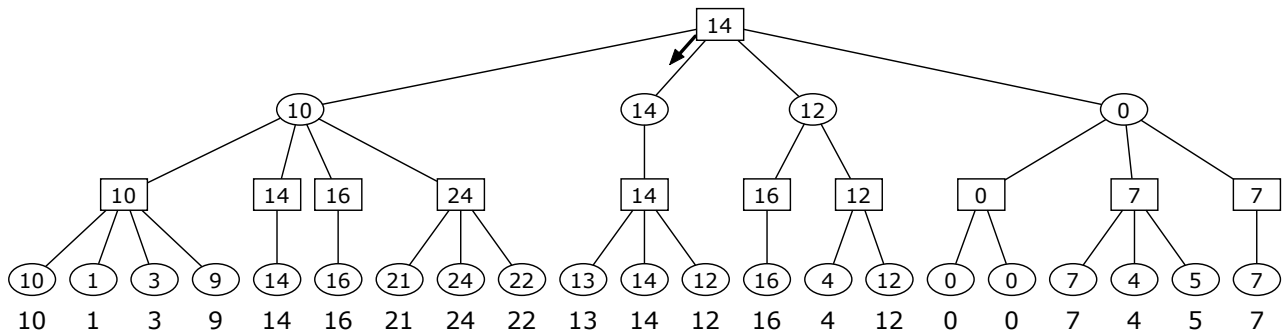
C8 (da C7 e C2): not auto(Y) or not diesel(Y)

C9 (da C8 e C5): not diesel(rs_tdi)

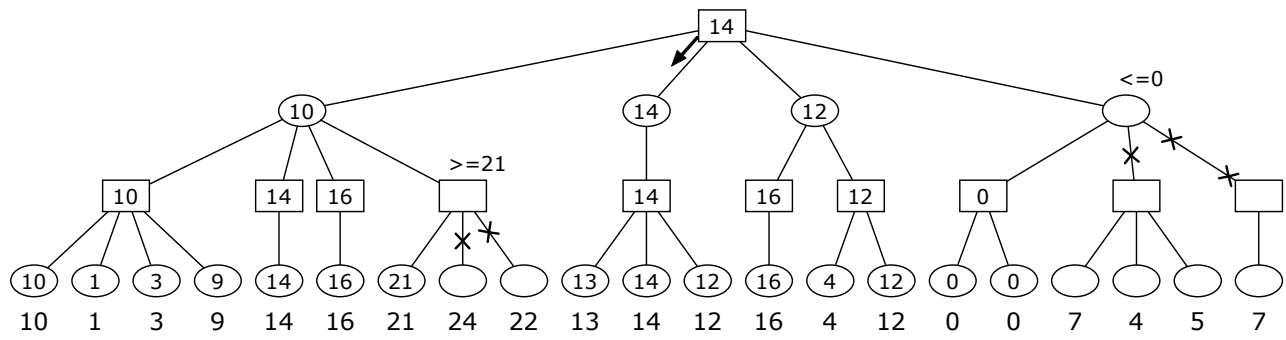
C10 (da C9 e C6): clausola vuota

Esercizio 2

min-max:



Tagli alfa-beta:



Esercizio 3

```

pari([], []) :- !.
pari([_], []) :- !.
pari([N1, N2 | Tail], [N2 | Tail2]) :- pari(Tail, Tail2).

```

Esercizio 4

