

COMPITO DI INTELLIGENZA ARTIFICIALE – PARTE I
COMPITO DI FONDAMENTI DI INTELLIGENZA ARTIFICIALE

9 Dicembre 2003

(Tempo a disposizione: 2h; Totale su 32 punti)

Esercizio 1 (punti 8)

Si esprimano in logica dei predicati del primo ordine le seguenti frasi:

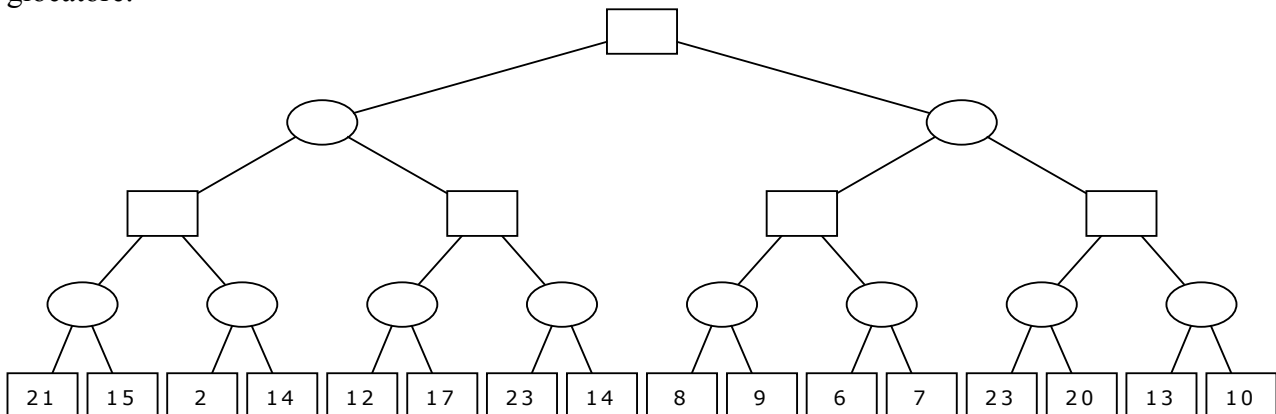
1. Tutti i bimbi, quando sono in casa, o giocano o guardano la tv (or esclusivo)
2. Esiste sempre almeno un bimbo in casa che guarda la tv
3. Elena e Marianna sono due bimbe che sono in casa
4. Elena non sta guardando la tv

Le si trasformi in clausole e si dimostri poi la seguente query applicando il principio di risoluzione:

Esiste un bimbo in casa che sta giocando

Esercizio 2 (punti 8)

Si consideri il seguente albero di gioco dove i punteggi sono tutti dal punto di vista del primo giocatore:



Supponendo che il primo giocatore sia MAX, quale mossa dovrebbe scegliere? Si risolva l'esercizio tramite l'algoritmo MIN-MAX. Successivamente si mostrino i tagli alfa-beta.

Esercizio 3 (punti 6)

Si scriva un programma Prolog `listindex(Lin1, Lin2, Lout)` che date due liste di elementi `Lin1` e `Lin2` contenenti numeri, produca una lista in uscita `Lout` contenente come elementi solo gli elementi della lista `Lin2` di posizione specificata da `Lin1`. Si noti che `Lin1` contiene sempre numeri ≥ 1 . Si scrivano esplicitamente tutti i predicati Prolog usati nella soluzione.

Esempio:

```
listindex([5,3,1], [9,20,11,5,19,21,0], X)
```

restituisce

```
X=[19, 11, 9]
```

Esercizio 4 (punti 8)

Considerando la base di conoscenza:

```
bin(1).  
bin(2).
```

```
parity(0,even):- !.  
parity(N,even):- N>0, M is N-2, parity(M,even), !.  
parity(N,odd).
```

si disegni l'albero SLD corrispondente all'invocazione:

```
?- bin(X), parity(X,even), parity(X,odd).
```

Esercizio 5 (punti 2)

Si descriva la strategia lineare applicata al principio di risoluzione. E' completa? Si motivi la risposta.

SOLUZIONE

Esercizio 1

Trasformazione in logica

Tutti i bimbi, quando sono in casa, o giocano o guardano la tv (or esclusivo)

1. $\forall B \text{ bimbo}(B) \wedge \text{incasa}(B) \Rightarrow \text{gioca}(B) \text{ xor } \text{guardatv}(B)$

Esiste sempre almeno un bimbo in casa che guarda la tv

2. $\exists B \text{ bimbo}(B) \wedge \text{incasa}(B) \wedge \text{guardatv}(B)$

Elena e Marianna sono due bimbe che sono in casa

3. $\text{bimbo}(\text{elena}) \wedge \text{bimbo}(\text{marianna}) \wedge \text{incasa}(\text{elena}) \wedge \text{incasa}(\text{marianna})$

Elena non sta guardando la tv

4. $\text{not } \text{guardatv}(\text{elena})$

Esiste un bimbo in casa che sta giocando

5. $\exists X \text{ bimbo}(X) \wedge \text{incasa}(X) \wedge \text{gioca}(X)$

Trasformazione in clausole

1. $\forall B \text{ bimbo}(B) \wedge \text{incasa}(B) \Rightarrow \text{gioca}(B) \text{ xor } \text{guardatv}(B)$
C1) $\text{not } \text{bimbo}(B) \text{ or } \text{not } \text{incasa}(B) \text{ or } \text{gioca}(B) \text{ or } \text{guardatv}(B)$
C2) $\text{not } \text{bimbo}(B) \text{ or } \text{not } \text{incasa}(B) \text{ or } \text{not } \text{gioca}(B) \text{ or } \text{not } \text{guardatv}(B)$
2. $\exists B \text{ bimbo}(B) \wedge \text{incasa}(B) \wedge \text{guardatv}(B)$ (Skolemizzazione)
C3) $\text{bimbo}(b)$
C4) $\text{incasa}(b)$
C5) $\text{guardatv}(b)$
3. $\text{bimbo}(\text{elena}) \wedge \text{bimbo}(\text{marianna}) \wedge \text{incasa}(\text{elena}) \wedge \text{incasa}(\text{marianna})$
C6) $\text{bimbo}(\text{elena})$
C7) $\text{bimbo}(\text{marianna})$
C8) $\text{incasa}(\text{elena})$
C9) $\text{incasa}(\text{marianna})$
4. $\text{not } \text{guardatv}(\text{elena})$
C10) $\text{not } \text{guardatv}(\text{elena})$
5. $\exists X \text{ bimbo}(X) \wedge \text{incasa}(X) \wedge \text{gioca}(X)$
GN) $\text{not } \text{bimbo}(X) \text{ or } \text{not } \text{incasa}(X) \text{ or } \text{not } \text{gioca}(X)$

Risoluzione

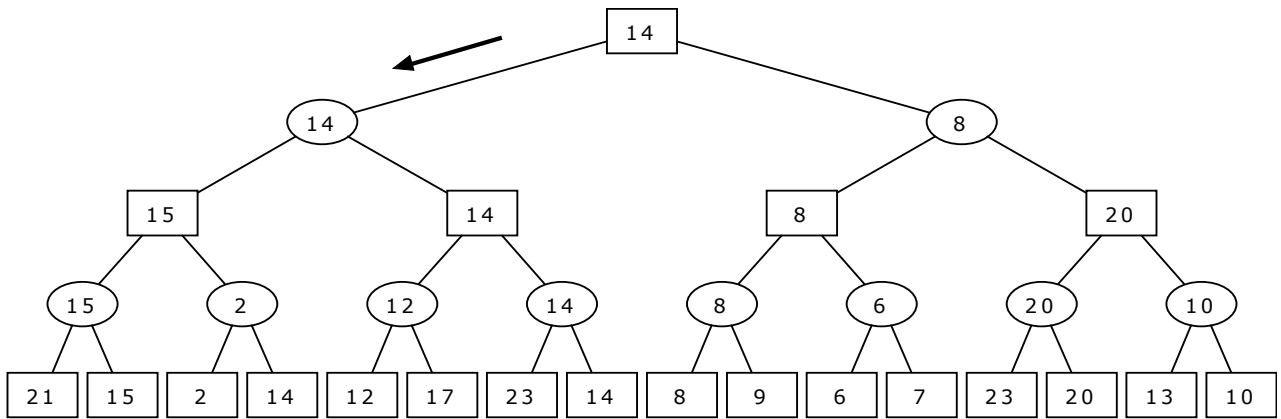
C11 = C6+C8+GN: $\text{not } \text{gioca}(\text{elena})$

C12 = C1+C11: $\text{not } \text{bimbo}(\text{elena}) \text{ or } \text{not } \text{incasa}(\text{elena}) \text{ or } \text{guardatv}(\text{elena})$

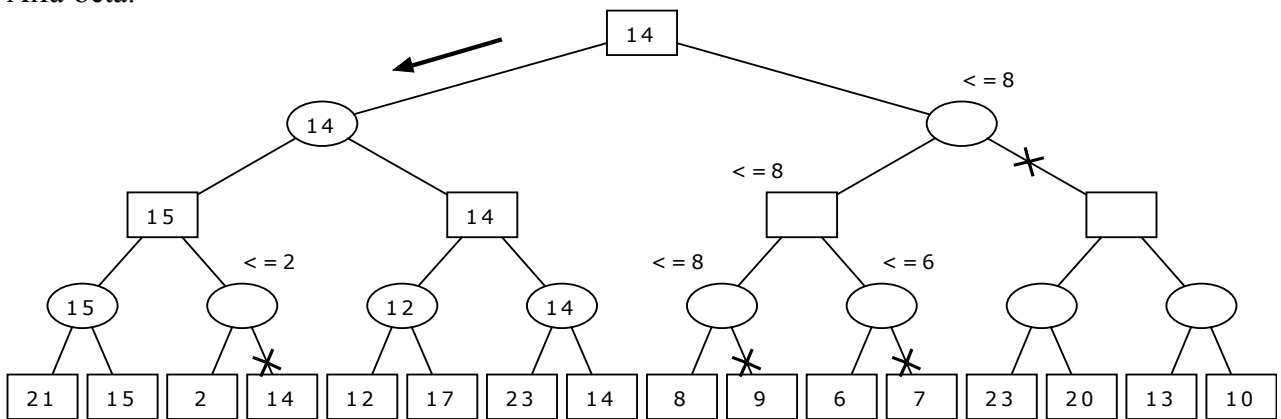
C13 = C12 + C6 + C8 + C10: $[\]$

Esercizio 2

Min-max:



Alfa-beta:



Esercizio 3:

```
listindex([], L, []).
listlindex([E1|Rest1], L2, [N3|Rest3):- estrai(E1, L2, N3),
listindex(Rest1, L2, Rest3).
```

```
estrai(1, [A|B], A):-!.
estrai(Nin, [A|B], Nout):-N is Nin 1, estrai(N, B, Nout).
```

Esercizio 4

- 1 bin(1).
- 2 bin(2).
- 3 parity(0, even):-
!.
- 4 parity(N, even):-
N=0,
M is N-2,
parity(M, even).
!.
- 5 parity(N odd).

