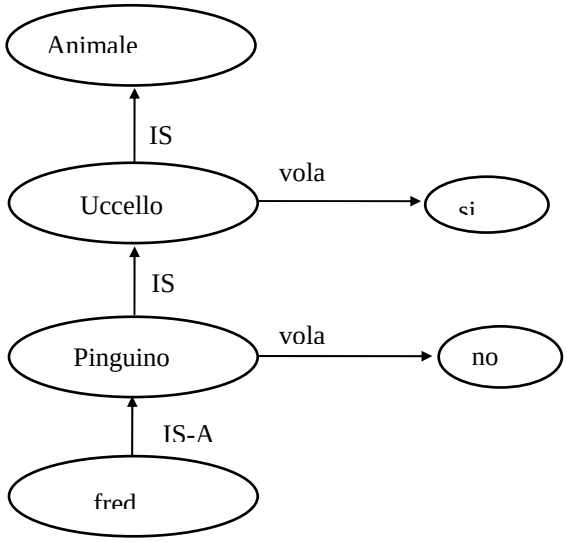
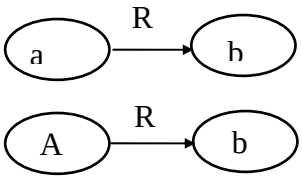
COMPITO DI INTELLIGENZA ARTIFICIALE (v.o.) – PARTE I
FONDAMENTI DI INTELLIGENZA ARTIFICIALE
16 Settembre 2005 (Tempo a disposizione 2 h, su 32 punti)

Esercizio 1 (punti 10)

Data la seguente rete Semantica:



Darne una traduzione ragionevole in First Order Logic (FOL) ricordando che:



r(a,b)

$\forall X \; a(X) \Rightarrow r(X,b)$

e che le relazioni IS e IS-A hanno una traduzione particolare (sottoinsieme e appartenenza).

a. Dire quale sarebbe la risposta alla domanda "Fred vola?", con il meccanismo proprio delle reti semantiche.
b. Si utilizzi il principio di risoluzione per mostrare quale sarebbe la risposta nella teoria logica FOL corrispondente.
c. Tale logica è traducibile in Logic Programming?

Esercizio 2 (punti 8)

Si consideri il seguente programma Prolog, che dovrebbe inserire un elemento in una lista ordinata, fornendo nel terzo parametro una lista ordinata:

```
insert(X,[H|T],[H|T1]) :- X>H, !, insert(X,T,T1).
insert(X,L,[X|L]).
```

Il programma è corretto? Se non lo è, lo si corregga. Si mostrino poi gli alberi SLD generati dalle query:

```
?- insert(9,[1,8],L).          ?- insert(9,[1,8],[1,9,8]).
```

Esercizio 3 (punti 9)

Uno dei puzzle logici più famosi nella letteratura recente è quello escogitato dal prof. Severus Snape (il puzzle di Snape) per proteggere la pietra filosofale. Questo puzzle, risolto da Hermione Granger (amica di Harry Potter), consiste nell'individuare due pozioni (una che consente di avanzare tra le fiamme e una per tornare indietro sani e salvi) tra una serie di sette bottiglie dal contenuto sconosciuto allineate sul tavolo. Il prof. Snape ha fornito una serie di indizi sotto forma di un poema. Per aiutarvi riportiamo sulla destra gli indizi fondamentali e la collocazione delle bottiglie così come si presume siano apparse ad Hermione.

Formulare come problema di soddisfacimento di vincoli e risolvere con una euristica MRV (cioè first-fail) e forward checking illustrando il procedimento passo per passo. In caso di parità si istanzino le bottiglie da sinistra a destra.

Le pozioni servono per avanzare o per tornare indietro oppure contengono vino d'ortica o contengono veleno (e quindi i domini saranno *{A, I, O, V} dove A=avanti, I=indietro, O=vino d'ortica, V=veleno*). Come euristica di selezione del valore, si assegnino i valori in ordine alfabetico (A,I,O,V).

Gli indizi sono cinque:

1. C'è sempre del veleno a sinistra del vino d'ortica.
2. Le bottiglie alle estremità hanno contenuti diversi, ma nessuna di queste serve per andare avanti.
3. Né la bottiglietta più piccola, né quella gigante contengono veleno.
4. La seconda da sinistra e la seconda da destra hanno lo stesso contenuto.
5. La prima bottiglia non contiene ortica.

Nell'interpretare i vincoli potete far riferimento alla figura a lato. Numerate le bottiglie da 1 a 7 partendo da sinistra.

Esercizio 4 (punti 5)

Si scriva un predicato Prolog `almost_same(L1,L2)` che dia valore vero se le due liste `L1` ed `L2` della medesima lunghezza contengono gli stessi elementi (con stesso ordine) tranne al più uno.

Esempio:

```
?- almost_same([1,3,2],[1,3,4]).
yes
?- almost_same([1,3,2],[1,4,3]).
no
?- almost_same([1,5,6],[1,2,3]).
no
```

Soluzione

Esercizio 1: Rete semantica

Teoria FOL corrispondente alla rete semantica:
1. $\forall$X uccello(X) $\Rightarrow$ animale(X)
2. $\forall$X pinguino(X) $\Rightarrow$ uccello(X)
3. pinguino (fred)
4. $\forall$X uccello(X) $\Rightarrow$ vola(X,si)
5. $\forall$X pinguino(X) $\Rightarrow$ vola(X,no)

a. Alla domanda “Fred vola?”, con il meccanismo proprio delle reti semantiche, viene data risposta negativa.
b. Applicando il principio di risoluzione, la risposta nella teoria logica FOL corrispondente alla query: ?- vola(fred,si) è positiva. La teoria FOL più il goal corrispondono alle clausole:

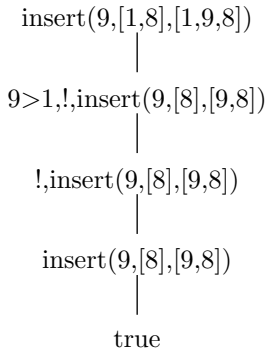
1. not uccello(X) or animale(X)
2. not pinguino(X) or uccello(X)
3. pinguino (fred)
4. not uccello(X) or vola(X,si)
5. not pinguino(X) or vola(X,no)
6. not vola(fred,sì)

Applicando la risoluzione:
Da (3) e (2): 7. uccello(fred)
Da (7) e (4): 8. vola(fred,sì)
Da (8) e (6): clausola vuota

c. Tale logica è traducibile in Logic Programming? Sì, la teoria FOL è un programma logico. Tuttavia, rispetto alla rete semantica originale il comportamento non è corretto (alla query ?- vola(fred,si) si dà risposta positiva, mentre nella rete originaria tale inferenza è falsa).

Esercizio 2

Il programma non è corretto, come si vede disegnando l’albero SLD della seconda query:

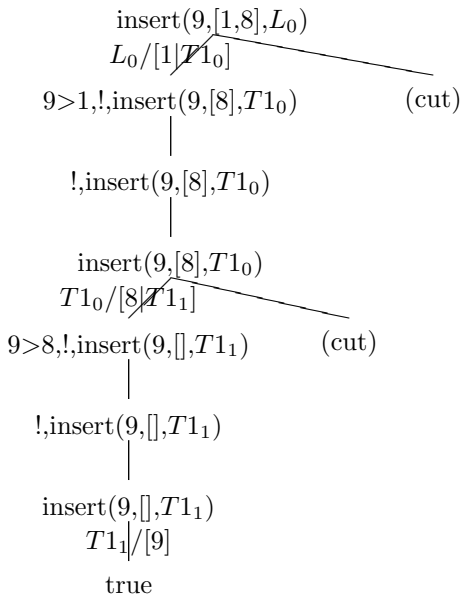


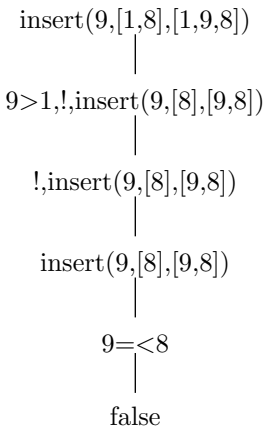
Il problema è nel fatto che se il goal non unifica con la prima clausola, nella seconda non viene effettuato il controllo su quale elemento sia il maggiore fra X e H. Si noti che la seconda clausola gestisce anche il caso della lista vuota.
Una versione corretta è la seguente:

```
insert(X,[],[X]).
```

```
insert(X,[H|T],[H|T1]) :- X>H, !, insert(X,T,T1).
insert(X,[H|T],[X,H|T]):- X=<H.
```

Gli alberi SLD corrispondenti alle invocazioni:





Esercizio 3

Suppongo di numerare le bottiglie con 1, 2 … 7 da sinistra a destra
Variabili: V1, V2 … V7
Dom(Vi) = {A, I, O, V} dove A=avanti, I=indietro, O=vino d'ortica, V=veleno
Vincoli:

1. *Vi = O => Vi-1 = V (i=2, … 7)*
2. *V1≠V7; V1≠A; V7≠A;*
3. *V4≠V; V6≠V*
4. *V2=V6*
5. *V1≠O*

Uso l'euristica MRV e Forward checking

V1={I, V} V2={A, I, O, V} V3={A, I, O, V} V4={A, I, O} V5={A, I, O, V} V6={A, I, O} V7= {I, O, V}	V1=I V2={A, I, V} V3={A, I, O, V} V4={A, I, O} V5={A, I, O, V} V6={A, I, O} V7={O, V}	V1=I V2={A, I, V} V3={A, I, O, V} V4={A, I, O} V5={A, I, O, V} V6={} BACKTR. V7=O	V1=I V2={A, I, V} V3={A, I, O, V} V4={A, I, O} V5={A, I, O, V} V6={A, I, O} V7=V
V1=I V2=A V3={A, I, V} V4={A, I, O} V5={A, I, O, V} V6={A} V7=V	V1=I V2=A V3={A, I, V} V4={A, I, O} V5={A, I, O, V} V6=A V7=V	V1=I V2=A V3=A V4={A, I} V5={A, I, O, V} V6=A V7=V	V1=I V2=A V3=A V4=A V5={A, I, V} V6=A V7=V

Soluzione: I, A, A, A, A, A, V

Esercizio 4:

```
almost_same([],[]).
almost_same([A|B],[A|C]):- !,almost_same(B,C).
almost_same([A|B],[_|B]).
```