Midterm I - Artificial Intelligence I

3:30pm – 4:45 pm

February 22, 2003

Name: ....

1. (20 points) Consider a language $\mathcal{L}$ with

- The set of constants consists of a, b, and c;
- The set of function symbols consits of f and g, both are 2-ary functions (functions with two parameters);
- Three predicate symbols p, q, and r where p, q, and r is a unary, binary, and 3-ary predicate symbol, respectively (i.e. they have one, two, and three parameters, respectively).

1.1. List

- 5 terms – a term is either a variable, a constant, or a term of the form $f(t_1, \ldots, t_n)$ where t_i are terms and f is a n-ary function symbol. Thus, the following are five terms in the language $\mathcal{L}$: a, b, X, $f(X, a)$, $g(f(X, a), b)$
- 5 ground atoms – an atom is of the form $p(t_1, \ldots, t_n)$ where t_i are terms and p is a n-ary predicate symbol. An atom is a ground atom if it does not contain variables. Hence, the following five atoms are ground atoms in the language $\mathcal{L}$: $p(a)$, $q(a, b)$, $r(f(a, b), a, b)$, $p(b)$, $q(f(a, b), g(a, b))$

that can be constructed using $\mathcal{L}$.

1.2 Answer the following questions.

- Is $p(f(a, b))$ a term or an atom? It is an atom since p is an unary predicate symbol, $f(a, b)$ is a term;
- Is $q(p, X, Y)$ a term or an atom? It is neither an atom nor a term; An predicate symbol(like p) cannot be used in place of a term. Besides, q is a binary predicate symbol. Thus, having three arguments is wrong.
- Is $g(p, q)$ a term or an atom? It is neither an atom nor a term; Predicate symbols (like p) cannot be used as a term.
- Is $\{X/a, Y/b\}$ a unifier of $q(X, b)$ and $q(a, Y)$? Yes, it is. Because $q(X, b)\{X/a, Y/b\} = q(a, b)$ and $q(a, Y)\{X/a, Y/b\} = q(a, b)$?
- What is the result of $r(f(X, Y), g(Y, Z), Z)\sigma$ where $\sigma = \{X/f(a, b), Y/g(a, b), Z/X\}$? The result is $r(f(f(a, b), g(a, b)), g(g(a, b), X), X)$.

2. (25 points) Given a KB

$$\begin{aligned} a &\leftarrow b \wedge c. \\ c &\leftarrow b. \\ d &\leftarrow \end{aligned}$$

2.1 Give an interpretation of the KB. We want to specify an interpretation $I = (D, \phi, \pi)$ of the KB.

Since the set of constants are empty, the set of individuals and the mapping between constants and individuals are not important in specifying any interpretation of the KB. Thus, we only need to specify the mapping of the predicate symbols (π) into Boolean function. This can be given by the following table:

$\pi(a)$	$\pi(b)$	$\pi(c)$	$\pi(d)$
$TRUE$	$TRUE$	$TRUE$	$TRUE$

2.2 Is your interpretation in (2.1) a model of the KB? Yes, it is. Because every clause of the KB is true in I (for every clause, the body is true and the head is true as well).

2.3 What can you say about the set of all logical consequences of the KB? Obviously, d must belong to the set of logical consequences of the KB since it is a fact.

It is easy to see, however, that d is the only logical consequence of the KB since none of a, b, c can be derived using d.

Thus, the best answer would be that the set of logical consequences of KB consists of only d.

2.4 What happens to the set of logical consequences of the KB (comparing to your answer in 2.3) if we change the second clause to the new clause

$$b \leftarrow c.$$

Nothing really changes since we cannot derive b, c, or a using the new rule.

2.5 What happens to the set of logical consequences of the KB (comparing to your answer in 2.3) if we add to the original clauses of the KB a new clause:

$$b \leftarrow c.$$

Nothing really changes since we cannot derive b, c, or a, even with the addition of the new rule.

3. (15 points) Given the KB

$$\begin{aligned} f(X) &\leftarrow g(X,Y) \wedge h(Y). \\ g(a,b) &\leftarrow \\ g(a,c) &\leftarrow \\ h(b) &\leftarrow \end{aligned}$$

Use the bottom-up proof procedure to compute all ground logical consequences of the KB.

First, we do the grounding:

$$f(a) \leftarrow g(a,a) \wedge h(a).$$

$$
\begin{aligned}
f(a) &\leftarrow g(a,b) \wedge h(b).\\
f(a) &\leftarrow g(a,c) \wedge h(c).\\
f(b) &\leftarrow g(b,a) \wedge h(a).\\
f(b) &\leftarrow g(b,b) \wedge h(b).\\
f(b) &\leftarrow g(b,c) \wedge h(c).\\
f(c) &\leftarrow g(c,a) \wedge h(a).\\
f(c) &\leftarrow g(c,b) \wedge h(b).\\
f(c) &\leftarrow g(c,c) \wedge h(c).\\
g(a,b) &\leftarrow\\
g(a,c) &\leftarrow\\
h(b) &\leftarrow
\end{aligned}
$$

Let the set of consequences be C.

We follow the steps of the bottom-up procedure:

- Step 1: $C = \emptyset$;
- Step 2: Select a clause, whose body is contained in C and whose head is not in C. In this case, we can pick either of the last three clauses. Let us take the clause $g(a,b) \leftarrow$. We have $\mathbf{C} = \mathbf{C} \cup \{\mathbf{g}(\mathbf{a},\mathbf{b})\} = \{\mathbf{g}(\mathbf{a},\mathbf{b})\}$.
- Repeat step 2: select $g(a,c) \leftarrow$. We have $\mathbf{C} = \mathbf{C} \cup \{\mathbf{g}(\mathbf{a},\mathbf{c})\} = \{\mathbf{g}(\mathbf{a},\mathbf{b}), \mathbf{g}(\mathbf{a},\mathbf{c})\}$.
- Repeat step 2: select $h(b) \leftarrow$. We have $\mathbf{C} = \mathbf{C} \cup \{\mathbf{h}(\mathbf{b})\} = \{\mathbf{g}(\mathbf{a},\mathbf{b}), \mathbf{g}(\mathbf{a},\mathbf{c}), \mathbf{h}(\mathbf{b})\}$.
- Repeat step 2: select $f(a) \leftarrow g(a,b) \wedge h(b)$. We have $\mathbf{C} = \mathbf{C} \cup \{\mathbf{f}(\mathbf{a})\} = \{\mathbf{g}(\mathbf{a},\mathbf{b}), \mathbf{g}(\mathbf{a},\mathbf{c}), \mathbf{h}(\mathbf{b}), \mathbf{f}(\mathbf{a})\}$.
- We cannot find a clause whose body is contained in C and whose head is not in C. So, we stop and answer with $\{g(a,b), g(a,c), h(b), f(a)\}$.

4. (40 points) Given the KB

$$
\begin{aligned}
father(tom, john) &\leftarrow\\
father(tom, sandra) &\leftarrow\\
father(john, mike) &\leftarrow\\
father(john, elena) &\leftarrow\\
mother(sandra, katy) &\leftarrow\\
mother(elena, jerry) &\leftarrow\\
ancestor(X,Y) &\leftarrow father(X,Y).\\
ancestor(X,Y) &\leftarrow mother(X,Y).\\
ancestor(X,Y) &\leftarrow ancestor(X,Z) \wedge ancestor(Z,Y).
\end{aligned}
$$

Use the SLD resolution to show that *tom* is an ancestor of *jerry*. That is, find a successful SLD derivation for the query

$$?ancestor(tom, jerry)$$

Shows the steps in detail.

First, we formulate the answer clause: $ac = yes \leftarrow ancestor(tom, jerry)$.

- $ac = yes \leftarrow ancestor(tom, jerry)$.
  Select the clause $ancestor(X, Y) \leftarrow ancestor(X, Z) \wedge ancestor(Z, Y)$.
  Renaming $ancestor(X1, Y1) \leftarrow ancestor(X1, Z1) \wedge ancestor(Z1, Y1)$.
  Substitution $\{X1/tom, Y1/jerry\}$.
  We get $ac = yes \leftarrow ancestor(tom, Z1) \wedge ancestor(Z1, jerry)$.

- $ac = yes \leftarrow ancestor(tom, Z1) \wedge ancestor(Z1, jerry)$.
  Select the clause $ancestor(X, Y) \leftarrow father(X, Y)$.
  Renaming $ancestor(P, Q) \leftarrow father(P, Q)$.
  Substitution $\{P/tom, Q/Z1\}$.
  We get $ac = yes \leftarrow father(tom, Z1) \wedge ancestor(Z1, jerry)$.

- $ac = yes \leftarrow father(tom, Z1) \wedge ancestor(Z1, jerry)$.
  Select the clause $father(tom, john)$. No renaming.
  Substitution $\{Z1/john\}$.
  We get $yes \leftarrow ancestor(john, jerry)$.

- $ac = yes \leftarrow ancestor(john, jerry)$.
  Select the clause $ancestor(X, Y) \leftarrow ancestor(X, Z) \wedge ancestor(Z, Y)$.
  Renaming $ancestor(X2, Y2) \leftarrow ancestor(X2, Z2) \wedge ancestor(Z2, Y2)$.
  Substitution $\{X2/john, Y2/jerry\}$.
  We get $yes \leftarrow ancestor(john, Z2) \wedge ancestor(Z2, jerry)$.

- $ac = yes \leftarrow ancestor(john, Z2) \wedge ancestor(Z2, jerry)$.
  Select the clause $ancestor(X, Y) \leftarrow father(X, Y)$.
  Renaming $ancestor(P1, Q1) \leftarrow father(P1, Q1)$.
  Substitution $\{P1/tom, Q1/Z2\}$.
  We get $ac = yes \leftarrow father(tom, Z2) \wedge ancestor(Z2, jerry)$.

- $ac = yes \leftarrow father(tom, Z2) \wedge ancestor(Z2, jerry)$.
  Select the clause $father(john, elena)$. No renaming.
  Substitution $\{Z2/elena\}$.
  We get $yes \leftarrow ancestor(elena, jerry)$.

- $ac = yes \leftarrow ancestor(elena, jerry)$.
  Select the clause $ancestor(X, Y) \leftarrow mother(X, Y)$.
  Renaming $ancestor(P3, Q3) \leftarrow mother(P3, Q3)$.
  Substitution $\{P3/elena, Q3/jerry\}$.
  We get $ac = yes \leftarrow mother(elena, jerry)$.

- $ac = yes \leftarrow mother(elena, jerry)$.
  Select the clause $mother(elena, yerr)$. No renaming.
  Substitution $\{\}$.
  We get $yes \leftarrow$.
  This is a successful SLD-derivation.