Tutorial for week 9 (week of 16 Nov)

1. Find two different unifiers U_1, U_2 for each of the following pairs of terms such that U_1 is most general (does the least amount of instantiation necessary), and such that U_2 is not most general.

   (a) `f(X,h(Z)), f(g(W),U)`

   (b) `f(X,h(Z)), f(g(W),h(W))`

2. Suppose that we work with a language with a set of constants $C = \{ c_1, c_2, c_3, \ldots \}$ and no function symbols.

   This means that Herbrand structures have as their domain C; to define an interpretation, we give e.g. for each two-place predicate p a subset $I(p) \subseteq C \times C$,[1] indicating that

   $$M \models p(c_i, c_j) \text{ means by definition } (c_i, c_j) \in I(p).$$

   For two Herbrand structures $\mathcal{M}_1, \mathcal{M}_2$ for the same set of predicates $p_1, p_2, \ldots$ we say that $\mathcal{M}_1$ is *smaller than* $\mathcal{M}_2$ ($\mathcal{M}_1 \leq \mathcal{M}_2$) just when for *every* predicate p_i and terms c_i, c_j:

   $$\text{if } \mathcal{M}_1 \models p_1(c_i, c_j) \text{ then } \mathcal{M}_2 \models p_1(c_i, c_j)$$

   Check that the following statements are equivalent
   (this is just a question of following the definitions).

   (a) if $\mathcal{M}_1 \models p(c_i, c_j)$ then $\mathcal{M}_2 \models p(c_i, c_j)$

   (b) (Using I_1, I_2 for interpretation in $\mathcal{M}_1$, $\mathcal{M}_2$), $I_1(p) \subseteq I_2(p)$

3. Consider the following program.

```
flies(X) :- bird(X).
flies(X) :- bee(X).
bird(X) :- parrot(X).
bird(X) :- skua(X).
bee(X) :- bumbleBee(X).
parrot(polly).
skua(sam).
bumbleBee(bert).
penguin(pete).
```

[1] the set $A \times B$ is just the set of pairs (a, b) with $a \in A, b \in B$.

(a) What are the predicates and constants?

(b) Find two different Herbrand models by assigning to each predicate a subset of the constants so that each program clause is true.

(c) The fixed-point characterisation of the least Herbrand model works by building bigger and bigger interpretations for the predicates until a fixed point is reached. The minimal Herbrand model is a model $\mathcal{M}$ of the given program, such that for *any* other model $\mathcal{M}'$ of the program, $\mathcal{M} \leq \mathcal{M}'$. What is the minimal Herbrand model here?

(d) Check that your minimal Herbrand model is indeed smaller than (or the same as) the models you found in part ??.

4. There is a standard philosophical distinction between the *use* of a word in the normal way, and the *mention* of a word (when the word is talked about). Syntax from an object logic which is *used* in that language can be *mentioned* in a meta-language.

   Which words in the following are being *used* and which are *mentioned*?

   > The artist formerly known as the artist formerly known as prince is now known as the artist.

5. The predicate `var/1` checks the argument when called, and succeeds if the argument is a variable, otherwise it fails.

   Here are two calls at the sicstus top level, with no program loaded:

```
| ?- var(X), X=2.
X = 2 ?
yes
| ?- X=2, var(X).
no
```

   Explain why this behaviour is incompatible with the declarative reading of Prolog programs.

6. Prolog can freely mix meta-predicates with standard predicates. Can you see a difference of *use* versus *mention* in this program? What operation does it compute, in the mode `flatten(?,-)` ?

```
flatten(X,[X]) :- var(X),!.
flatten([],[]) :- !.
flatten([H|T],L3) :-
        !,
        flatten(H,L1),
        flatten(T,L2),
        append(L1,L2,L3).
flatten(X,[X]).
```