Logic Programming, Fall 2014
Tutorial 7: sample answers

1. (a) Constants: polly, sam, bert, pete. There are no function symbols. Predicates: flies, bird, bee, parrot, skua, bumbleBee, penguin.

   (b) The ground terms are: polly, sam, bert, pete.

   (c) There are 28 ground atomic formulas, each has the form $p(c)$ where p is one of the 7 predicate symbols and c is one of the 4 constants.

   (d)

$$\begin{aligned} f(Y) \;=\; & \{parrot(polly), skua(sam), bumbleBee(bert), penguin(pete)\} \cup \\ & \{flies(c) \mid (bird(c) \in Y) \vee (bee(c) \in Y), c \text{ constant}\} \cup \\ & \{bird(c) \mid (parrot(c) \in Y) \vee (skua(c) \in Y), c \text{ constant}\} \cup \\ & \{bee(c) \mid bumbleBee(c) \in Y, c \text{ constant}\} \cup Y \end{aligned}$$

   (e)

$$\begin{aligned} f(\{\}) &= \{parrot(polly), skua(sam), bumbleBee(bert), penguin(pete)\} \\ f^2(\{\}) &= f(\{\}) \cup \{bird(polly), bird(sam), bee(burt)\} \\ f^3(\{\}) &= f^2(\{\}) \cup \{flies(polly), flies(sam), flies(burt)\} \\ f^4(\{\}) &= f^3(\{\}) \end{aligned}$$

   (f) The Herbrand universe is $\{polly, sam, bert, pete\}$. Constants are interpreted as themselves, e.g., $polly^H = polly$, etc. For each predicate p, define p^H to be function mapping c to true if $p(c)$ is in the least fixed point of f, so:

$$\begin{aligned} parrot^H(c) = true &\Leftrightarrow c = polly \\ skua^H(c) = true &\Leftrightarrow c = sam \\ bumbleBee^H(c) = true &\Leftrightarrow c = bert \\ penguin^H(c) = true &\Leftrightarrow c = pete \\ bird^H(c) = true &\Leftrightarrow c \in \{polly, sam\} \\ bee^H(c) = true &\Leftrightarrow c = bert \\ flies^H(c) = true &\Leftrightarrow c \in \{polly, sam, burt\} \end{aligned}$$

2. (a) $CWA(T) = T \cup \{\neg rich(karen), \neg poor(keith), \neg happy(keith)\}$

   (b) Consider $T' = T \cup \{happy(keith)\}$. Then $T \subseteq T'$, and $CWA(T) \models \neg happy(keith)$, but it is not the case that $CWA(T') \models \neg happy(keith)$.

   (c) Yes, for this example $CWA(T)$ does predict the behaviour of Prolog negation by failure since Prolog answers yes to

```
\+A
```

   where A is a ground atomic formula, only for exactly the 3 cases

$$A = rich(karen), poor(keith), happy(keith)$$

   that $CWA(T)$ negates.

(d) The Clark completion is:

$$\forall X.happy(X) \leftrightarrow poor(X)$$
$$\forall X.poor(X) \leftrightarrow X = karen$$
$$\forall X.rich(X) \leftrightarrow X = keith$$
$$\neg(karen = keith)$$

(e) Yes, Clark completion does exhibit non-monotonicity. Consider T' as in (b), then $Comp(T')$ is:

$$\forall X.happy(X) \leftrightarrow (poor(X) \vee X = keith)$$
$$\forall X.poor(X) \leftrightarrow X = karen$$
$$\forall X.rich(X) \leftrightarrow X = keith$$
$$\neg(karen = keith)$$

So $T \subseteq T'$, and $Comp(T) \models \neg happy(keith)$, but it is not the case that $Comp(T') \models \neg happy(keith)$.

3. CWA gives:

$$happy(tweedledum) \vee happy(tweedledee)$$
$$\neg happy(tweedledum)$$
$$\neg happy(tweedledee)$$

The problem is that this is inconsistent!

4. When there are no function symbols, the set of ground atomic formulas (for the signature of predicates and constants appearing in program and query) is finite. One can then explicitly construct the minimum Herbrand model by computing the least fixed-point of the function corresponding to the program, as in Question 1. The validity of the query is then decided merely by checking if there is a tuple of constants, instantiating the existential quantifiers, for which the resulting conjunction of ground atomic formulas are all true in the minimum Herbrand model (i.e., contained in the least fixed-point set).

Note: The set of ground atomic formulas becomes infinite once a single unary function symbol is present. Decidability for the function-symbol-free fragment is specifically a property of definite clause logic. General predicate logic for a single binary predicate (no function symbols, no constants) is undecidable.