German University in Cairo
Department of Computer Science
Dr. Haythem O. Ismail

Introduction to Artificial Intelligence, Winter Term 2014
Project 2: Unification and Clause Form

Due: November 27, by 23:59

1. **Project Description:** In this project you will implement two functions/methods in the language of your choice. The first, `Unify`, takes two FOL terms or atomic sentences and returns a most general unifier, if one exists. (You do *not* need to implement the particular unification algorithm we discussed in class, you may do some research and implement a better algorithm among those proposed in the literature.) The second, `ClauseForm`, takes a well-formed sentence of FOL and returns an equivalent one in clause form. Your functions should allow a *trace* mode in which partial results are pretty-printed. By implementing these two functions, you would have implemented two major components of a resolution-based reasoning system.

2. **Test Cases:** Make sure you test `Unify` on the following examples.

   a) $P(x, g(x), g(f(a)))$ and $P(f(u), v, v)$

   b) $P(a, y, f(y))$ and $P(z, z, u)$

   c) $f(x, g(x), x)$ and $f(g(u), g(g(z)), z)$

   Also check `ClauseForm` on the following inputs.

   a) $\exists x[P(x) \land \forall x[Q(x) \Rightarrow \neg P(x)]]$

   b) $\forall x[P(x) \Leftrightarrow (Q(x) \land \exists y[Q(y) \land R(y, x)])]$[1]

3. **Groups:** You may work in groups of at most three.

4. **Deliverables**

   a) Source Code

   - You should implement the function `Unify`, described above.
   - You should implement the function `ClauseForm`, described above.
   - Both functions should be tested on the above test cases.
   - Both functions should be runnable in a *trace* mode, in which partial results are pretty-printed.
   - Part of the grade will be on how readable your code is. Use explanatory comments whenever possible.
   - If you use code available in library or internet references, make sure you comment *each line* of the code.

[1] Example is due to Stuart Shapiro

b) Project Report, including the following.

- A brief discussion of unification and clause form FOL.
- A discussion of how your functions represent FOL expressions.
- A discussion of your implementation of `Unify`.
- A discussion of your implementation of `ClauseForm`.
- A sample run, showing the results of applying both functions on the above test cases.
- Instructions on how to run (in regular and *trace* mode) and exit the functions.
- Proper citation of any sources you might have consulted in the course of completing the project.
- If you use code available in library or internet references, make sure you fully explain how the code works.
- If your program does not run, your report should include a discussion of what you think the problem is and any suggestions you might have for solving it.

5. Important Dates

Source code. On-line submission by November 27 at 23:59. Directions for on-line submission are available on the course web site.

Project Report. A hard-copy should be submitted. You have two options:

a) November 27, by 16:00, or

b) November 29, by 16:00, provided that an *identical* on-line version is submitted with the code (by November 27 at 23:59).

Brainstorming Session. In tutorials.