20CS6033/20CS5133 AI – I

Fall2016

Instructor: Anca Ralescu

**Homework Assignment #4**

**Assigned on October 18, 2016**

**Due on November 1, 2016**

**40 points**

In this program you are to implement in FRIL a small theorem prover for propositional logic. You will need to implement a predicate called **roi** (from **r**ule **o**f **i**nference) for each of the rules of inference for propositional logic. These rules are to be used conjunction with a knowledge base **in at most two inference steps**.

To begin with, you must decide how to represent the logic operators: **and, or, ~** and **imp**. You may select any convention you want provided that you make that explicit, so that a user can enter his/her arguments in the correct format.

Next you must adopt various conventions on how to use these symbols in conjunction with the propositional letters:

For example, one could use **(~ p)** to denote **~p**.  In any case, you must be very careful in your choices so that the rules of evaluating logic expressions will be applied unambiguously.

For example, the De Morgan law should be recognized to evaluate

**(~ (p and q))**as (**(~ p) or (~ q))**

The use of parentheses must be very judicious, as one hand they simplify a lot the  implementation of rules, on the other hand their use puts a lot of burden on the user to enter many parentheses, and to cope reading a result that contains many parentheses.

You may use the knowledge base either in **conjunctive normal form** or in the **implicative form** but you need to specify which one you use so that the user can enter the knowledge base accordingly.

In the light of the above, your program should then do the following:

1. Be able to perform one step inference with each rule implemented
2. Accept a knowledge base and two specified rules of inference, and output all the results from the knowledge base using these two rules (Forward Chaining).
3. Accept a knowledge base and a conclusion and verify that this conclusion can be obtained from the knowledge base through at most two steps of inference (Backward Chaining).

Notice that you program may also need to recognize the priority of operations.

Some ideas on how to proceed:

Your program should have a knowledge base containing the basic rules of inference described in a predicate **roi** of the following format:

(**(roi NAME  TYPE PREMISES CONCLUSION))** which holds if the rule of inference with name NAME and premises PREMISES written in TYPE (cnf or inf) can be used to infer CONCLUSION.  For example (you will find these in small\_prover.frl) :

**((roi UnitResolution  cnf ((A or B) (~ B))  A))**   
**((roi UnitResolution  cnf ((A or B) (~ A))  B))**

or, in implicative form:

**((roi UnitResolution  inf  ( ( (~ A) imp B) (~ B))  A))**

**((roi AndElimination \_ ((A and | T)) A))**   
**((roi AndElimination \_ ((A and | T )) B)(member B T))**

**((roi AndIntroduction \_ () ()))**   
**((roi AndIntroduction \_  (A | T) ( A and | T1) )**   
**(negg member\_op OP T)**   
**(roi AndIntroduction \_ T T1))**

**((operators (and or imp ~)))** % this lists the logical operators known to the prover  
**((member\_op L)(operators OPS )(member OP OPS))**

De Morgan's Rules:   
**((deMorgan () ()))**   
**((deMorgan ((~ (A and | T) )) ((~ A) or T1))(deMorgan T T1))**   
**((deMorgan ((~ (A or | T) )) ((~ A) and T1))(deMorgan T T1))**

You need to define a predicate **small\_prover**

**((small\_prover  TYPE PREMISES CONCLUSION STEPS))**

which holds when CONCLUSION can be obtained from PREMISES by applying the sequence of nontrivial STEPS (no more than two).  Please notice that a trivial step is that in which the conclusion is actually one of the premises.

You will need to use list processing FRIL predicates some of which you can find in the file list\_pro.frl . Include those predicates in a separate section of your program file preceded by the comment /\* list processing predicates \*/

I am also providing you with some help in the files attached to this homework. Please include any predicates used from these files in a separate section preceded by the comment /\* from <fill in the file name> \*/

Please write your solutions neatly. Please turn in only one copy per team and include, clearly in your homework the names of EACH team member. Your assignment should contain only one file.