METU Department of Computer Engineering Spring 2019

CENG 462—Artificial Intelligence

Homework 2 Due to: May 2, 2019
F.Polat

Implement a theorem prover for First Order Predicate Logic using *resolution refutation* technique and the *Set of Support* strategy. This program gets two lists of clauses; the list of base clauses and the list of clauses obtained from the negation of the theorem. Your program has to eliminate

- tautologies and
- subsumptions.

Your program returns whether the theorem is derivable, or not. If derivable, it has to print the resolutions that contribute to the proof of the theorem. Variables, predicate names and function names starts with a lower case letter, while constants with an upper case letter.

Input:
The first line of the input file (`input.txt`) contains two positive integers n and m corresponding to the number of clauses in the base set B and the number of clauses in the set G of clauses obtained from the negation of the theorem, respectively. In each of the following n/m lines (line 2 to line $n+1$/line $n+2$ to line $n+m+1$), there is a single clause from B/G which contains literals (atomic predicate or its negation) separated by commas.

```
3 1
p(A,f(t))
q(z),~p(z,f(B))
~q(y),r(y)
~r(A)
```

Output:
The first line of the output file (`output.txt`) contains "yes", or "no" corresponding to derivable or not derivable. Note that `empty_clause` symbolizes empty clause. In each of the following lines, there are 3 clauses separated by "$" where the first two are parent clauses, the third is the resolvent

```
yes
~r(A)$~q(y),r(y)$~q(A)
~q(A)$q(z),~p(z,f(B))$~p(A,f(B))
~p(A,f(B))$p(A,f(t))$empty_clause
```

Submission Policy

- You are going to submit a single file called `hw2.py`.
- Your codes will be graded on inek machines. Make sure they run on these machines correctly.
- Late submissions will not be accepted.
- All work must be done individually and in compliance with the course policy.