

# CENG 462

## Introduction to Artificial Intelligence

Fall '2016-2017

### Programming Homework 2

---

Due date: 23 December 2016, Friday, 23:55

## AUTONOMOUS THEOREM PROVING

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
- 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 an integer stating the number of tasks to be solved. Then for the following lines, 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  $n + m + 1$  with respect to the line containing  $n$  and  $m$  for that task), there is a single clause from  $B/G$  which contains literals (atomic predicate or its negation) separated by commas.

```
1
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. The same pattern goes on for the rest of the tasks. 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*

$\sim r(A) \$ \sim q(y), r(y) \$ \sim q(A)$

$\sim q(A) \$ q(z), \sim p(z, f(B)) \$ \sim p(A, f(B))$

$\sim p(A, f(B)) \$ p(A, f(t)) \$ \text{empty\_clause}$

## 1 Regulations

1. Your code should be in Python.
2. **Late Submission: Not Allowed**
3. **Cheating: We have zero tolerance policy for cheating.** People involved in cheating will be punished according to the university regulations.
4. **Newsgroup:** You must follow the newsgroup ([news.ceng.metu.edu.tr](http://news.ceng.metu.edu.tr)) for discussions and possible updates on a daily basis.
5. **Evaluation:** The .py file will be checked for plagiarism automatically using “black-box” technique and manually by assistants, so make sure to obey the specifications.

## 2 Submission

Submission will be done via COW. Upload your code file named ”the2.py”. Do not send any other files.