CSC520 Fall 2023 Assignment 3

Due October 12^{th} at 11:59pm

This assignment consists of three questions which involve written answers and code. In order to complete the assignment you must submit a written report in pdf form detailing your answers to the questions as well as your code. As discussed in class all work *must* be your own. You may not use third party libraries or example code to complete the assignment with the exception of csv file loaders. All reports must be clear and well written. All code must be clear, readable, and well-commented. Upload your report as a file called `<unityid>-Assign3.pdf` and your sourcecode in a zip called `<unityid>-Assign3.zip`.

Question 1 First-Order Predicate Logic (10 points)

Consider the following English sentences.

- All lions are animals.
- The head of a lion is the head of an animal.

1. (5 pts) Convert the sentences into first-order predicate logic. Be extremely careful about quantification; because not everything in the universe is an animal, or a lion, or a head, you will need both kinds of quantification. Use the following lexicon:

   lion(X) X is a lion.

   animal(X) X is an animal.

   head(H,X) H is the head of X.

2. (5 pts) Following the steps on text book to convert the logic statements into CNF.

3. (5 pts) Using resolution prove using FOPL resolution that the second of the original sentences follows from the first. Number your clauses, and indicate explicitly step-by-step what resolves together, under what substitution.

Question 2 Problem Solving (70 pts)

You have been given a text file representing a basic knowledge base. Each line of the file represents a single statement in conjunctive normal form composed of individual variables, represented as lower-case letters, with basic punctuation and negations. Thus:

```
(a | -b | -c | d) & (c | q)
```

Is equivalent to:

$$(a \vee \neg b \vee \neg c \vee d) \wedge (c \vee q)$$

Your task in this assignment is to implement code that loads the specified knowledge base into memory, generating an internal representation of the prepositional logic, and then performs resolution theorem proving to solve a given query. Your code should be called as follows:

`python query.py <HEU> <QUERY> <KB>` or `java -jar query.jar <HEU> <QUERY> <KB>`

where `<HEU>` is either `BRUTE` or `TARGET` and represents a heuristic used in search; `<QUERY>` is a CNF statement representing the rule we are seeking to prove true (e.g. `"(-c | d)"`); and `<KB>` is the knowledge base file we will use. You make make use of regular expression libraries and standard data structures for this task.

On execution your code must print out each intermediate step showing the combinations made. Once the execution is complete it will print out the result of the query (true or false) and it will print out the final state of the knowledge base.

As discussed in class resolution theorem proving can be done in a brute-force manner, where we combine all possible pairings until no new knowledge can be created. Or it can be done in a heuristically-guided manner where we sort the possible rule candidates by some factor such as holding the query variable or complexity. Your task in this assignment is to implement both a brute-force selection and one heuristic guide *of your choice* and then to use it when evaluating both the current KB and the one you define below.

For your final report you must:

1. Define a set of test queries, that are compatible with this KB and the one you define below.
2. Execute the queries with your knowledge bases and report on their relative performance under the two algorithms.
3. Evaluate the impact of your chosen heuristic and suggest potential improvements.

Question 3 Propositional Logic Problem (20 points)

Your task in this problem is to identify a new propositional logic problem suitable for use with your solver above. For this task you must define a *novel* problem that is suitable for propositional logic, define a knowledge base for it, and show how that KB will be used to address real-world queries. This must be an original problem and you may *not* copy it from published materials. For this question you will submit:

1. A brief description of your problem and how a knowledge-based agent built on it will be used in real applications.
2. A KB covering at least 8 variables and 8 unique rules including 2 implications, represented in propositional logic.
3. A sample query with the steps for backward-chaining resolution.
4. A representation of the KB suitable for use with your code above.