A priority Queue for a simplified agenda manager in a rule-based expert system shell

Assignment - 3

Geoffrey Rathinapandi

Fall 2017

Table Of Contents

1.Objective :

To implement a simplified agenda manager in a rule base expert system shell using priority queue implementation. The basic elements of rule base system are a knowledge based and inference engine.

2.Program Description :

The program gets the input file name from the user. Once the user provides the file-name, file is obtained & started process each line of the input file. The first line will be read & parses the names of the rule & priority value. After the whole line is read, a heap will be built using buildQ() & heapify() methods. After building the heap, the higher priority rule will be found out & displayed as output through ExtractMax() method. The displayed element will be removed from the heap.

Similarly rules present in further lines of input file will be added into the heap built. At the end of adding all elements in each line of input file, the highest priority rule will be displayed. After the input file is fully read, the rules will be displayed as per the highest priority until all the rules are processed .The agenda manager will half after 30 cycles or if there is no more rules to execute.

3.Executing the Program :

- Download the File
- Save the Text file in the same location
- Running through terminal :
    - Browser through the directory where the program is.
    - Use python3 *ProgramName.py* to execute the program
    - User will be prompted to enter the input file

Program Logic :

• Consider a text file that contains a set of rules. Each activated rule has a corresponding priority.

• Agenda Manager reads a line and builds priority list based on the priority of the rules and builds the heap

• Then it executes the most prioritized rule from the heap.

• The agenda manager first deletes from the agenda the previously executed rule that was executed in a previous cycle.

• Agenda manager updates list of rules by adding new rules and deleting rules that was executed in a previous cycle.

• Agenda manager then determines the rule with highest priority for the inference engine to execute in this cycle. Only one rule can be executed in each cycle. .

• Then the cycle continues and the Agenda Terminates after it has finished 30 cycles

4. Test Results

The Agenda Manager is tested with three sample input files and the results of those is below

4.1 Input File 1

The Program has been tested with the first input file *test1.txt* and the result has been uploaded in GitHub below is the link.
https://github.com/Ge0f3/AgendaManager/blob/master/Results/test1Res.txt

4.2 Input File 2

The Program has been tested with the first input file test1.txt and the result has been uploaded in GitHub below is the link.
https://github.com/Ge0f3/AgendaManager/blob/master/Results/test2Res.txt

4.3 Input File 3

The Program has been tested with the first input file test1.txt and the result has been uploaded in GitHub below is the link.
https://github.com/Ge0f3/AgendaManager/blob/master/Results/test3res.txt

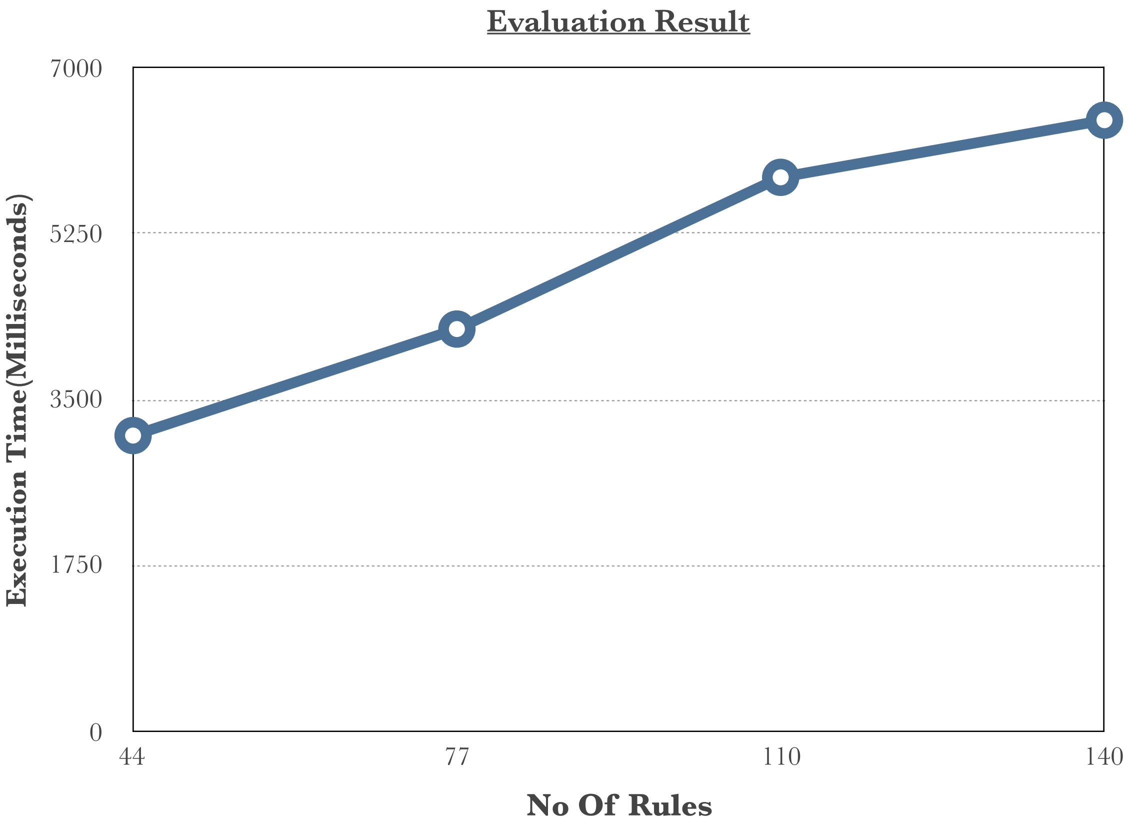
5 .Performance Evaluation

The performance of the program is evaluated using various input size and the result is plated in the graph below.

Input Size (Rules)	Running Time (Milliseconds)
44	3121.31
77	4244.01
110	5843.32
140	6445.55

5.1 Result & Graph

The graph reflects the worst-case complexity of the priority queue which is O(nlgn)



6.Conclusion

A priority queue for a simplified agenda manager in a rule-based expert system shell can be implemented in such way.