COS30019

Introduction to AI Assignment - 2 Inference Engine

Syed Omair Maqdoom Mohiuddin

102863767

COS30019_A02_T025

Table of Contents

Instructions

The program is developed in Java, a general purpose programming language that is object-oriented and class-based. The language is structured in such a way that developers can write code anywhere and run it anywhere without worrying about the underlying computer architecture. It is also referred to as write once, run anywhere (WORA). It is commonly used to build wide range of platforms. Before running a Java program it needs to be compiled. The executable file can be run using another application (IDE) or run from the command line (terminal). This project is a command line application which takes two arguments. A source fille and Algorithm to be used.

To compile a Java program, we use the below syntax**:**

javac file_name.java

If there are multiple independent files which are required to compiled together, we use the below syntax:

javac file_1.java file_2.java file_3.java OR **javac *.java (**only when Javafiles are in same directory)

After compilation the .java files gets translated into .class file(byte code). And to tun the Program use the below syntax:

java ClassName (Note that the file name specified must be .class file that contains a main method)

We have compiled all the files and provided the executable file.

We have also created a batch file which has the code to run the java program.

To run the program use the below syntax:

InferenceEngine algorithm text_file_name

Here, InferenceEngine is the .bat(batch) file which calls the program. algorithm can be replaced by **TT (for Truth Table), or FC (for Forward Chaining) or BC (for Backward chaining)** to specify the algorithm, and filename can be replaced by the actual file name.

For Instance:

InferenceEngine TT test_HornKB.txt

Introduction

An inference engine is a program that allows you to draw logical conclusions regarding knowledge assets. The inference engine is frequently mentioned as a component of a knowledge base by experts. Inference engines may be used to improve business intelligence by dealing with various types of data. An inference engine is frequently a component of a knowledge base; when used in conjunction with the knowledge base, the inference engine assists users in extracting logical insights from the vast amount of data available to them. Inference engine capability is available from a variety of suppliers for gaining meaningful insights from sales and business data. Larger corporations, like Microsoft, have their own inference engine tools.

One of the components of master frameworks is the inference engine for propositional rationale, which refers to a set of methods that use natural language to generate new realities from existing ones or respond to a variety of knowledgebase queries, whereas the information base refers to a set of sentences that depict realities about the world in some proper language (Schroyens, 2010). Induction motors usually need a search. A specialized framework is made up of three parts: the user interface, the induction motor, and the data bank. The user interface allows framework clients to collaborate with the master framework. The knowledge base contains excellent and space-specific information. The derivation motor portion is responsible for reverse and forwards tying. This is the section where coherent standards are implemented to the data base to obtain fresh data or make a decision. The surmising motor uses the retrogressive and forwards fastening operations as approaches for suggesting arrangements or completing data in the master framework.

Considering the Truth Table, Backward Chaining, and Forward Chaining algorithms, this assignment seeks to establish a derivation motor for propositional logic in programming. The program is developed in Java and tested at several stages to ensure its accuracy.

Features

The application contains various facets. It parses a knowledge base text file into horn forms and query. Command line is used to run the application. There are three algorithms that are employed in the inference method.

Forward Chaining

Forward chaining is a method of thinking in artificial consciousness in which surmising principles are used to current data to extract further data until an endpoint (goal) is reached (Grout, 2006). Before reasoning new facts, the surmising motor assesses current realities, determinations, and situations in this type of binding. The control of information in the information base allows an endpoint (goal) to be achieved. Arranging, viewing, managing, and understanding applications can all benefit from forward fastening.

The forward-chaining method PL-FC-ENTAILS?(KB,q) examines if a knowledge base of definite clauses entails a single proposition symbol q—the query. It starts with the knowledge base's known facts (positive literals). If all of an implication's premises are known, the inference is added to the list of known facts.

Properties of Forward Chaining

- Very natural algorithm for knowledge bases in Horn Form
- The interaction utilizes a down-up approach (bottom to top)

- Forward fastening starts with an underlying condition and employs realities to bring it to a conclusion
- The process is data-driven
- It is sound and complete

Advantages

- It may be used to achieve a range of inferences.
- It gives a solid foundation for drawing conclusions.
- Because the data obtained from it is not limited, it is more versatile than backwards chaining.

Disadvantages

- Forward chaining is a time-consuming procedure. Eliminating and synchronizing accessible data might take a long time.
- Unlike backwards chaining, this sort of chaining does not have a clear explanation of facts or observations. The former employs a goal-oriented approach to arrive at quick conclusions.

Backward Chaining

Backward Changing is an idea in Artificial Intelligence that includes backtracking from the objective to steps which informed the outcome (Cohen, 2017). These algorithms work backward from the goal, chaining through rules to find known facts that support the proof. We describe the basic algorithm, and then we describe how it is used in logic programming, which is the most widely used form of automated reasoning. We also see that backward chaining has some disadvantages compared with forward chaining, and we look at ways to overcome them. Finally, we look at the close connection between logic programming and constraint satisfaction problem. Individuals can also use these systems to design coherent advancements that can be used to search down additional major models. In reverse tying, it may be used for researching, diagnosing, and solving problems.

If the knowledge base contains a sentence of the type lhs goal, where lhs (left-hand side) is a list of conjuncts, FOL-BC-ASK(KB, goal) will be proved. An atomic fact such as American(West) is treated as a clause with the empty list as the lhs. A query with variables can now be verified in a variety of ways.

Properties of Forward Chaining

- Very natural algorithm for knowledge bases in Horn Form
- It is a form of goal-directed reasoning
- The cost is much less than linear in the size of the knowledge base
- It uses up-down approach
- To illustrate the truth of realities, the endpoint (goal) is divided into sub-objectives.

Advantages

- Because the outcome is already determined, conclusions are simple to form.
- Because the conclusion is known, it is a faster technique of reasoning than forwards chaining.
- Correct answers may be obtained successfully in this form of chaining if the inference engine follows pre-determined rules.

Disadvantages

- The reasoning process can only begin if the conclusion is known
- Multiple solutions or answers are not deduced
- It only gets the data it needs, therefore it's less versatile than forwards chaining.

Truth Tables

The definition of an equation as a repetition was used in truth tables, presuming that it was valid under each interpretation (Solaki, 2018). However, using truth tables to evaluate the reality of an equation quickly becomes infeasible. For larger equations, we should look for a better technique - something that is really schematic but doesn't rely on individual values anymore.

Information and reasoning are also important for fake professionals since they enable successful ways of acting that would be incredibly difficult to achieve using other methods. (Ghidini and Serafini, 2017). It's vital to remember that data on activity outcomes allows critical thinkers to function successfully in difficult situations. In any event, the knowledge about critical thinking professionals is solid and unwavering. Information-based experts can benefit from data presented in extremely wide formats by connecting and recombining data to meet their needs. This cycle is frequently far removed from the criteria that exist apart from everything else, such as when a mathematician proves a theory or a space specialist calculates the world's future.

Bugs/Missing

Truth Table:

Bugs: None Known

Forward Chaining:

Bugs: None Known

Backward Chaining

Bugs: None Known

Error Handling

- **Invalid Input Format:** If user enter more than 2 arguments in the input then he is prompted with the required format of the input.

```
Please check your input. Required Input Format: [Algorithm] [File]
Available Algorithms:
TT
FC
BC
```

Figure 1: Invalid Input Format

- **Incorrect File Name:** When a user knowingly/unknowingly inputs a file name which doesn't exist, the program throws a File not Found Exception.

```
C:\Users\7250\OneDrive - Swinburne University\Swinburne\Year 3\Semester1\Introduction to AI\assignment2>InferenceEngine TT new.txt
Error File Not Found.Please check your input
java.io.FileNotFoundException: new.txt (The system cannot find the file specified)
        at java.base/java.io.FileInputStream.open0(Native Method)
        at java.base/java.io.FileInputStream.open(FileInputStream.java:216)
        at java.base/java.io.FileInputStream.<init>(FileInputStream.java:157)
        at java.base/java.io.FileReader.<init>(FileReader.java:75)
        at Parser.readTextFile(Parser.java:15)
        at Main.ReadFile(Main.java:39)
        at Main.main(Main.java:15)

C:\Users\7250\OneDrive - Swinburne University\Swinburne\Year 3\Semester1\Introduction to AI\assignment2>
```

Figure 2: File Not found Exception

- **Case Insensitive:** If user inputs lower/upper/mixed case in any algorithm, the program interprets it and changes as the requested algorithm.

```
C:\Users\7250\OneDrive - Swinburne University\Swinburne\Year 3\Semester1\Introduction to AI\assignment2>InferenceEngine Bc test1.txt
YES p2, p3, p1, d

C:\Users\7250\OneDrive - Swinburne University\Swinburne\Year 3\Semester1\Introduction to AI\assignment2>
```

Figure 3: Case Insensitive

- **Unavailable Algorithm:** If user inputs a unavailable algorithm, then the program gives an error and outputs the available algorithms.

```
C:\Users\7250\OneDrive - Swinburne University\Swinburne\Year 3\Semester1\Introduction to AI\assignment2>InferenceEngine RB test1.txt
Invalid Algorithm
Available Algorithms:
TT
FC
BC

C:\Users\7250\OneDrive - Swinburne University\Swinburne\Year 3\Semester1\Introduction to AI\assignment2>
```

Figure 4: Unavailable Algorithm

Test Cases

A test-driven development technique was used to create the program. Test cases were built to check the validity of this program, and the outputs were analyzed.

TEST_1:

TELL

p2=> p3; p3 => p1; c => e; b&e => f; f&g => h; p1=>d; p1&p3 => c; a; b; p2;

ASK

D

FC: YES a, b, p2, p3, p1, d

BC: YES p2, p3, p1, d

TT: YES 3

TEST_2:

TELL

p=> q; l&m => p; b&l => m; a&p => l; a&b => l; a; b;

ASK

Q

FC: YES a, b, , l, q

BC: YES b, l, m, p, q

TT: YES 1

Test 3: Query Can't be proven

TELL

a=> b; b => a;

ASK

B

FC: NO

BC: NO

TT: NO

TEST 4:

TELL

a&z=> b; a => b; a;

ASK

B

FC: YES a, b

BC: YES a, b

TT: YES 3

TEST 5:

TELL

a&b => d; a; b; p2;

ASK

D

FC: YES a, b, p2, d

BC: YES b, d

TT: YES 1

Research

I have used several test cases with various inputs to check the validity of the program.

Research TEST 1:

TELL

a&b=>d; b=>c; c&a=>d; b&d=>e; c&e=>f; d=>b; a; c;

ASK

F

FC: YES a, c, d, b, d, c, e, f

BC: YES a, d, e, f

TT: YES 1

Research_TEST_2:

TELL

a => b; b =>c; c => d; a;

ASK

D

FC: YES a, b, c, d

BC: YES a, b, c, d

TT: YES 1

Research_TEST_3:

TELL

a => p3; p3 => p1; c => e; a; b; p2;

ASK

p1

FC: YES a, b, p2, p3, p1

BC: YES a, p3, p1

TT: YES 3

Research_TEST_4: Query Can't be proven

TELL

ASK

f

FC: NO

BC: NO

TT: NO

<u>Research TEST 5:</u> Query Can't be proven

TELL

a => b;

ASK

c;

FC: NO

BC: NO

TT: NO

Resources

www.youtube.com. (n.d.). *Forward and BackwardChaining*. [online] Available at: https://www.youtube.com/watch?v=EZJs6w2YFRM&ab_channel=FranciscoIacobelli

www.youtube.com. (n.d.). *Forward Chaining in Artificial Intelligence | Forward Chaining in Artificial Intelligence Example*. [online] Available at: https://www.youtube.com/watch?v=PBTSdx_C9WM&ab_channel=WellAcademy

www.youtube.com. (n.d.). *Artificial Intelligence - Forward and backward chaining method in Artificial Intelligence*. [online] Available at: https://www.youtube.com/watch?v=Rzh906gvz9I&ab_channel=DEEBAKANNAN

www.javatpoint.com. (n.d.). *Forward Chaining and backward chaining in AI - Javatpoint*. [online] Available at: https://www.javatpoint.com/forward-chaining-and-backward-chaining-in-ai.

Section. (n.d.). *Forward and Backward Chaining in Artificial Intelligence*. [online] Available at: https://www.section.io/engineering-education/forward-and-backward-chaining-in-ai/.

I have used the above resources to get better understanding about forward and backward chaining. There are also some YouTube videos which explains how the inference engine works.

Cohen, W.W., Yang, F. and Mazaitis, K.R., 2017. 'Tensorlog: Deep Learning Meets Probabilistic' DBs. arXiv preprint arXiv:1707.05390. https://arxiv.org/abs/1707.05390

Ghidini, C, & Serafini, L. 2017 'Distributed first order logic', Artificial Intelligence 253:1– 39.

Grout, I. (2006). *Integrated circuit test engineering : modern techniques*. London: Springer.

Quoc Vo, B. (2022), Logical Agents & Knowledge Representation, Introduction to AI, Learning material via canvas, Swinburne University of Technology

Lecture material provided a brief understanding of various forward and backward algorithm

Russell, S.J. and Norvig, P. (2016). *Artificial intelligence : a modern approach*. Upper Saddle River: Pearson.

The above book aided me on understanding the various algorithms, knowledge base.

Schroyens, W, 2010. A meta-analytic review of thinking about what is true, possible, and irrelevant in reasoning from or reasoning about conditional propositions. European Journal of Cognitive Psychology, 22, 897 921

I relied on this resource to advance my skills and understanding of conditional propositions. I later adopted the skills in designing and developing the project.

Solaki, A, 2018, ‘Rule-based reasoners in epistemic logic’, in J. Sikos, and E. Pacuit, (eds.), At the Intersection of Language, Logic, and Information - ESSLLI 2018 Student Session, vol. 11667 of LNCS, Springer, pp. 144–156.