Intelligent Systems



Rule Based Systems

[30077548@students.southwales.ac.uk](mailto:30077548@students.southwales.ac.uk?subject=Rule%20Based%20Systems%20Report)

Sebastian Ha

30077548

Table of Contents

[Introduction 3](#_Toc210416843)

[Evaluation of the System 4](#_Toc210416844)

[AI Context 4](#_Toc210416845)

[Knowledge acquisition and Learning 4](#_Toc210416846)

[Adjustments needed 4](#_Toc210416847)

[Conclusion 4](#_Toc210416848)

[Appendices 4](#_Toc210416849)

# Introduction

Expert Systems are computer programs that use artificial intelligence to mimic decision making and problem-solving abilities of human experts in a certain field or topic to answer a presented problem domain.

Expert systems are computer programs that utilise knowledge and inference mechanisms to provide advice, solutions or recommendations in specialised domains. They are designed to capture and emulate the expertise of human specialists to solve complex problems and make informed decisions.

<https://www.youtube.com/watch?v=O-FZ4Q8RXds>

The problem domain I was assigned was the Wait for Table decision problem. This covers the everyday decision of a restaurant customer of choosing whether they should wait for a table by considering many factors such as how busy the restaurant is, waiting times for a table, whether there is another restaurant they can go to and other types of factors that can affect their resulting decision. For this problem I was allocated a decision tree to work off to help create the expert system.

The purpose of expert systems in this assignment is to encode the decision process behind the wait for table decision tree. By having rules applied to the facts provided by the customer the system will be able to advise whether a customer should wait for a table. Further to this, expert systems are designed to capture specialist knowledge and apply it systematically to individual cases, supporting decision making in a transparent way.

# Evaluation of the System

**Design Decision – Encapsulation of Knowledge Stores**

In this implementation, both the **WorkingMemory** (facts) and **RuleBase** (rules) are stored as private members of the **InferenceEngine**. This design choice ensures that only the inference engine has the ability to read or update these structures. The **ApplicationManager** interacts with the inference engine rather than modifying the knowledge directly.

This encapsulation reflects the theoretical model of an expert system, where the inference engine acts as the “reasoning mechanism” and is solely responsible for comparing facts against rules and updating the working memory. By preventing external components from directly manipulating the knowledge base or working memory, the system maintains integrity and consistency in its reasoning process.

Conflict res

The inference engine implements conflict resolution by selecting the most specific rule (the one with the greatest number of conditions). This improves the naturalness of the consultation by ensuring that the next question asked is maximally informative. The traceStep function then identifies the first missing condition from that chosen rule, thereby determining the next fact to elicit from the user.

When multiple rules are consistent, the engine applies **conflict resolution by specificity**: it selects the rule with the greatest number of conditions (i.e., the most specific rule). This tends to ask more informative questions and reduces unnecessary dialogue. We do not use recency, refractory inhibition, or rule priorities in this version.

# AI Context

# Knowledge acquisition and Learning

# Adjustments needed

# Conclusion

# Appendices