Knowledge Representation in Artificial Intelligence and Expert Systems Using Inference Rule

***Edward E. Ogheneovo**, Dept. of Computer Science, University of Port Harcourt, Nigeria.

Promise A. Nlerum, Dept. of Computer Science and Informatics, Federal University, Otuoke, Nigeria.

*edward.ogheneovo@uniport.edu.ng, nlerumpa@fuotuoke.du.ng

Abstract

Knowledge representation is a very important concept in expert systems and artificial intelligence (AI) in general. It involves the consideration of intelligent (expert) systems and how it presents knowledge. It is best understood in term of the roles it plays based on the task at hand. A knowledge representation involves reasoning about the world rather than taking action in it. It is a set of rules, i.e., an answer to the question and a medium for efficient computation, that is, the computational environment in which thinking is accomplished. In this paper, we discussed knowledge representation using inference rule and forward chaining. The paper demonstrates the use of inference rule in explaining forward chaining using an admission process using some premises or antecedents to derive the conclusion. Some propositions or atomic sentences consisting of logical operators AND and OR are also used to infer conclusions based on some truth of additional proposition symbols.

Keywords: Artificial intelligence, expert systems, knowledge representation, inference rule, forward chaining.

1.0 Introduction

Knowledge representation is one of the fundamental concepts in expert systems and artificial intelligence (AI) [1] [2]. The field of knowledge representation involves considering intelligent (expert) systems and how it presents knowledge. Knowledge representation can best be understood in term of the roles it plays based on the task at hand. A knowledge representation is most fundamentally surrogate, a substitute for the thing itself that is used to enable an entity to determine consequences by thinking rather than acting, i.e., by reasoning about the world rather than taking action in it. It is a set of ontological commitments [3]. That is, it provides an answer to question bordering on the world around us. For instance, it answers question such as "In what terms should I think about the world?" It is part of the theory of intelligent reasoning expressed in terms of three components: 1) the representation's fundamental conception of intelligent reasoning, 2) the set of inferences that the representation sanctions, and 3) the set of inference that it recommends [4]. Knowledge representation is a medium for pragmatically efficient competition, i.e., the computational environment in which thinking is accomplished and human expression based on the things about the world [5] [6] [7]. Pragmatically, a representation provides for organizing information to facilitate making the recommended inferences and taking necessary decisions based on the outcome of such inferences [8].

Knowledge representation is a medium of understanding the roles individuals play in society and acknowledging their diversities [9]. As a field artificial intelligence (AI) and expert systems, knowledge representation has several useful consequences. First, each role requires something slightly different from a representation; which eventually leads to an interesting and different set of properties that we want a representation to have. Secondly, we believe that roles provide a framework that is useful for characterizing a wide variety of representations [10] [11]. Basically, the fundamental aspect of a representation can be captured by understanding how it views each of the roles and this will help reveal essential similarities and differences.

Knowledge representation is used to formalize and organize knowledge. One of the most commonly used representation is the production rule, or simply rule, which contains the knowledge base [12] [13] [14]. However, the term knowledge-base is a collection of rules or other information structures derived from the human expert. These rules consist of a condition or premise followed by an action or conclusion. Thus a rule consists of an IF – THEN parts. The IF part is called the condition or the antecedent and the THEN part is the action or consequence. The IF part lists a set of conditions in some logical combinations. The piece of knowledge represented by the production rule is relevant and must be in line with the reasoning being developed. If the IF part of the rule is satisfied; consequently, the THEN part can be concluded, or its problem-solving action is then taken. Expert systems whose knowledge is represented in rule form is called rule-based systems. Thus the problem–solving model, or paradigm, organizes and controls the steps taken to solve the problem [15] [16].

One common but powerful paradigm involves chaining of IF – THEN rules to form a line of reasoning. If the chaining starts from a set of conditions and move towards some conclusion, the method is called forward chaining. On the other hand, if the conclusion is known but the path to that conclusion is unknown, then reasoning backwards is used. This process is called backward chaining. These problem–solving methods are built into program modules engines or inference procedures or functions that manipulate and use knowledge in the knowledge-base to form a line of reasoning [17] [18]. The knowledge-base an expert use is what he learned at school, from colleagues, and from years of experience through practice. Therefore, we can infer that the more experience an expert has, the larger his store of knowledge. Knowledge allows him to interpret the information in his database for diagnosis, design, and analysis. Knowledge is almost always incomplete and uncertain. Thus a rule may have associated fact(s) with a confidence factor or a weight. The set of methods for using uncertain knowledge in combination with uncertain data in reasoning is called reasoning with uncertainty [19] [20] This paper discusses the knowledge representation using inference rule which particular reference to forward chaining.

2.0 Rule-Based Expert Systems

In humans and artificial intelligence (AI) problem-solving it is important to know how knowledge is represented in order to solving a problem knowledge representation deals with the question of how human knowledge can be encoded into a for that can be handled by computer algorithms and heuristics. Knowledge representations [21] are developed using different languages to ensure completeness, consistency, expressive and extensible for humans to comprehend and for computers to be able to solve such problems based on the symbols and syntax of the language. Usually, knowledge representations are encoded either by using declarative or procedural programming principles or both. Also, in most cases, knowledge representation is a mixture of explicit and implicit knowledge available to users or computers via inference process and formalisms such as symbols, frames, semantic networks, conceptual graphs, inference rules and sub-symbolic patterns [22]. However, in this paper, we only applied inference rules.

In early 1970s, Newell and Simon from Carnegie–Mellon University proposed a production system model, the foundation of the modern rule–based expert systems. The production model is based on the idea that humans solve problems by applying their knowledge (expressed as production rules) to a given problem represented by problem–specific information. The production rules are stored in the long–term memory and the problem–specific information or facts in the short–term memory. The basic structure of an expert system is shown in figure 1 It contains the following components or modules: knowledge base, database, the inference engine, explanation facilities, user interface and user. Knowledge base: The knowledge base (KB) contains the domain knowledge useful for problem solving. In rule–based expert system, the knowledge is represented as a set of rules. Each rule specifies a relation, recommendation, directive, strategy or heuristic and has the IF (condition) THEN (action) structure. The IF part is the consequent. Whenever the condition part of a rule is satisfied, the rule is said to fire and the action part is executed. Figure 1 shows the structure of a rule–based expert system [23]. As seen in the figure, there are several components in it. These components include: databases, inference engines, expert systems (which consists of knowledge base, and rule), explanation facilities, and user interface.

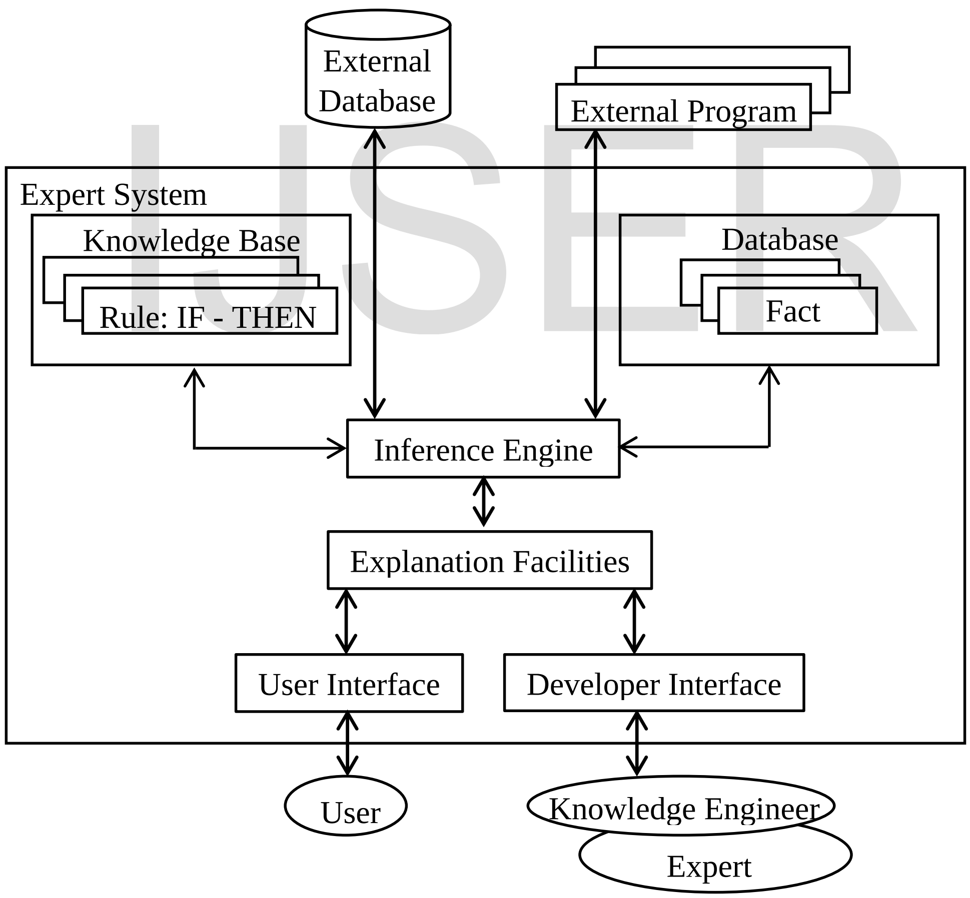


Fig. 1: Structure of a rule–based expert system

Database: The database includes a set of facts used to match against the IF (condition) parts of the rules stored in the knowledge base.

Inference Engine: This is a control mechanism for navigating through and manipulating knowledge and deducing results in an organized manner. It applies the axiomatic (self – evident) knowledge base to the task–specific data to arrive at some conclusion. Thus the inference engine carries out the reasoning which the expert system deduce the solution. The inference engine links the rules given in the knowledge base with the facts provided in the database.

The Explanation Facilities: The explanation facilities help the user to ask the expert system how a particular conclusion is reached and why a specific fact is needed. An expert system must be able to explain its reasoning and justify its advice, analysis or conclusion.

The User Interface: The user interface is the medium through which a user communicates with the expert system. It is through the user interface that a user seeking a solution to the problem communicates with the expert system.

2.1 Inference Engine

An inference engine is a software that performs the inference reasoning tasks. It uses the knowledge in the knowledge base and information provided by the user to infer new knowledge. The inference engine is often based on the use of rules called inference rules. The inference engine usually interacts with the knowledge base (i.e., IF - - - THEN - - - ELSE Statements), which contains information about how to solve problems within the problem domain. This is the global memory where the knowledge base system is records information relating to a specific problem that it is trying to solve [24]. Much of the information comes from the user but the memory is also used by the inference engine to record its own conclusions and to remember its chain of reasoning. By comparing what it knows about the problem domain in general with what it knows about the specific problem, the inference engine tries to proceed logically towards a better solution. It does this by using a mechanism that matches information in the knowledge database with pertinent action rules in the knowledge base, and if several rules apply, it selects the most appropriate one. It then implements the selected action by using chaining, either as forward chaining or backward chaining to arrive at a conclusion [25].

Rule–based systems are used as a means of storing and manipulating knowledge to interpret information in a useful way [26]. The term is often used in systems involving human related rule sets. Rule – based system is often used in artificial intelligence and research problems. In rule – based systems, much of the knowledge is represented as rules, i.e., as conditional sentences relating statements of facts with one another. Rule based systems are used to represent knowledge because human mental process is internal and therefore too complex to be represented as an algorithm. This is why most expert systems express their knowledge in the form of rules for problem solving. In rule – based expert systems, the knowledge representation method is a systematic way of "encoding" what an expert knows about some domain. Although, there are numerous knowledge representation methods, the logic – based ones are essential to the theory and practice of rule–based systems and expert systems in general. In encoding rule–based systems, propositional logic can serve as a practically useful language as it makes analysis and design of these systems relatively simple. The most basic logical form of proposition rules is: P_1

$\wedge P_2 \wedge . . . \wedge P_n \rightarrow h$. This form of a rule is logically equivalent to a Horn clause, provided that all the literals are positive. A more complex rule may contain conclusion part composed of several propositions [27].

2.2 The IF – THEN Structure

The IF – THEN structure of knowledge representation in expert system is used to relate given information or facts in the IF part to some action in the THEN part. The IF part is the conditional part while the THEN part is action part that describes how a problem can be solved. That is, rule–based knowledge representation consists of the IF part (i.e., antecedent – premise or condition) and the THEN part (i.e., consequent – conclusion or action) . A rule can have multiple antecedents joined by AND (conjunction) or OR (disjunction). The antecedent of a rule incorporates the object and its value are linked by an operator. For example, the IF – THEN part structure is as follows:

IF <antecedent>

THEN <consequent>

However, for a rule with multiple antecedents joined with AND or OR or a combination of both, we can have the following structure

IF <antecedent 1> **AND**

<antecedent 2> **AND**

⋮

<antecedent n> **AND**

THEN <consequent>

IF <antecedent 1>

OR <antecedent 2>

⋮

OR <antecedent n>

THEN <consequent>

However, the consequent of a rule can also have multiple clause:

IF <antecedent n>

THEN <consequent 1>

<consequent 2>

⋮

<consequent m>

The antecedent of a rule incorporates two parts: an **object** and its **value**. As an example, let as consider road traffic light.

R1: IF you study hard
THEN you will pass your exams

R2: IF you fail to study hard
THEN be ready to fail your exams

These statements represented in the IF – THEN forms are called production rule or just rules. The term 'rule' in artificial intelligence and expert systems is defined as an IF – THEN structure that relates given information or facts in the IF part to some action in the THEN part. What a rule does in knowledge representation is to provide some description of how to solve a problem. Relatively, rules are easy to create and use.

3.0 Methodology

As an example, consider the candidate seeking for admission to do a master's programme in a university.

Rule 1:	**IF**	(Bachelor's degree certificate is available) **AND**
		(Transcript is available) **AND**
		(Degree is in chosen course) **AND**
		(CGPA < 3.0)
	THEN	(Deny admission)
Rule 2:	**IF**	(Bachelor's degree certificate is available) **AND**
		(Transcript is available) **AND**
		(Degree is in another course) **AND**
		(CGPA >= 3.0)
	THEN	(Deny admission)
Rule 3:	**IF**	(Bachelor's degree certificate is available) **AND**
		(Transcript not available) **AND**
		(Degree is in another course) **AND**
	THEN	(Deny admission)
Rule 4:	**IF**	(Bachelor's degree certificate is available) **AND**
		(Transcript is available) **AND**
		(Degree is in another course) **AND**
		(PGD is available)
		(CGPA < 4.0)
	THEN	(Deny admission)
Rule 5:	**IF**	(Bachelor's degree certificate is available) **AND**

		(Transcript is available) **AND**
		(CGPA >= 3.0) **AND**
	THEN	(Recommend admission)
Rule 6:	**IF**	(Bachelor's degree certificate is available) **AND**
		(Transcript is available) **AND**
		(Degree is in another course) **AND**
		(PGD is available)
		(CGPA >= 4.0)
	THEN	(Recommend admission)

Based on these rules, a candidate seeking for admission for a master's programme will be recommended for admission if he met the criteria or denied admission if otherwise.

Fig. 2:

4.0 Representing Knowledge Using Forward Chaining

The solution to some problems naturally starts rom the collection of information. In this process, reason is applied tis information to obtain logical conclusions. To apply reason to gather the information needed to solve a problem, logical rule is applied. This logical rule is called chaining. Chaining is the process of obtaining the output of one rule by activating another rule. Chaining technique is used to break the task (problem) into small procedures and then inform each procedure within the sequence by itself. Two types of chaining exist. They are: forward chaining and backward chaining. In forward chaining, first the rules for matching facts are tested, and then the action is executed. In the next stage, the working memory is updated with new facts and the matching process starts all over. This process continues until no more rules are left, or until the goal is reached. Forward chaining is a data-driven reasoning approach that starts from the known facts and tries to match the rules with these facts. Sometimes, there is a possibility that all the rules match the information (condition). Forward chaining is useful when a lot of information is available and can be implemented if there are infinite number of potential solutions like configuration problems and planning. Forward chaining uses bottom-up computational approach to problem solving. It starts with a set of known facts and applies rules to generate new facts whose premises match the known facts and continue this process until it reaches a predetermined goal or until no further facts can be derived whose premises match the known facts. It checks the facts against the query or predeterm9ned goal and indicates that the inference moves forward from the facts towards the goal [28].

Backward chaining is goal-driven reasoning method. It starts from the goal (i.e., from the end), which is a hypothetical solution and the inference engine tries to find the matching evidence. When it is found, the condition becomes sub-goal, and then rules are searched to prove these sub-goals. It simply matches the right-hand-side (RHS) of the goal. This process continues until all the sub-goals are proved, and it backtracks to the previous step where a rule was chosen. If there is no rule to be established in an individua sub-goal, another rule is chosen. Backward chaining is good for situations where there are not so much facts and the information (facts) should be generated by the user. Backward chaining reasoning is also effective for application in the diagnostic tasks. Backward chaining is similar to hypothesis testing in human problem-solving.

This type of reasoning process is modelled in expert systems using a goal-driven search. It is a top-down computational approach to problem solving and it starts with a goal or hypothesis. It attempts matching the variables that lead to valid facts in the data and indicates that the inference moves backward from the intended goal to determine facts that would satisfy that goal.

This paper discusses knowledge representation using inference rue and forward chaining. In this form of chaining, the inference engine starts with facts and matches them to the conditions of a rule. If the condition is satisfied, the rule's conclusions are used to prove additional or further rules. This process continues until sufficient rules and facts are established to make a conclusion. Therefore, forward chaining is an expert system strategy to answer the question "what happens next?" It follows the chain of conditions and derivations and finally deduces the outcome. It considers all the facts and rules, and sort them before drawing a conclusion on the solution [29]. Figure 3 shows the forward chaining technique. Forward chaining technique is used to determine the conclusion and result of the rules in figure 2.

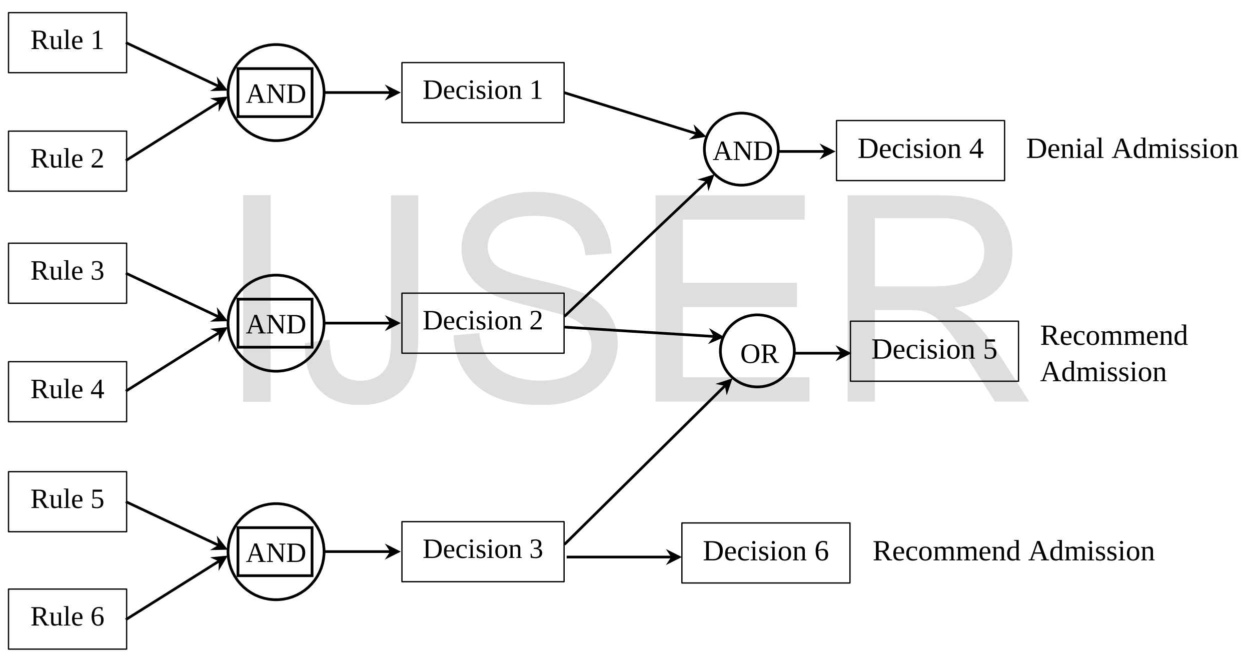


Fig. 3: Forward Chaining for figure 2.

In forward chaining, the basic idea is to fire any rule whose premises are satisfied in the knowledge and continuously make its conclusion to the knowledge base until query is found. Figure 2 is an example of forward chaining. In the figure, we have four facts: Fact 1, …, Fact 4 grouped into two. Facts 1 and 2, and Facts 3 and 4 using the AND logic operator to form decision 1 and 2 respectively. These decisions are further joined using an AND operator to arrive at decision 4 which now form the conclusion. In forward–chaining, for example, rules are applied by checking if their preconditions are satisfied. When a rule is executed (i.e., fired), its conclusion is added to the current knowledge base [30] [31].

As an example, consider the propositions in figure 3 for an atomic sentence using the logical operators AND and OR to infer some conclusions based on some premises.

Given

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$

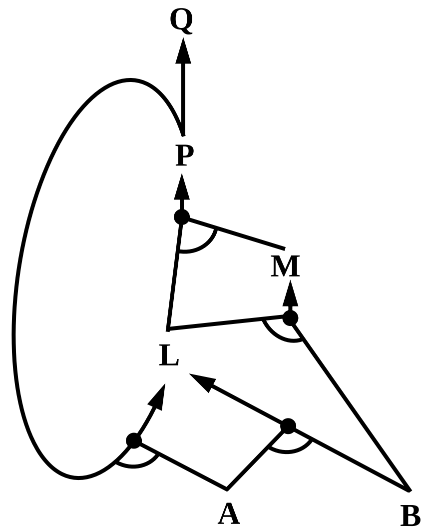
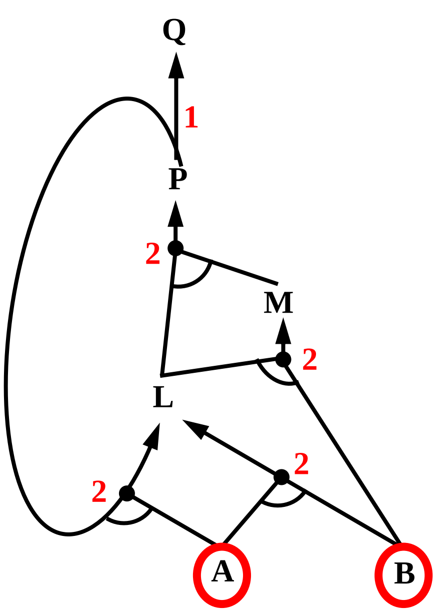


Fig. 3: Proposition symbols for forward chaining

In forward chaining, basically, we start with given proposition symbols (i.e. atomic sentence), for example, A and B as seen above. Iteratively, we then try to infer truth of additional proposition symbols, e.g., $A \wedge B \Rightarrow L$, hence, we establish L as true. We continue to infer in this process until there is no more inference that can be carried out or until we have reached to goal. Initially, taking A and B as our agenda, and annotate horn clauses with number of remises, which in this case 2, we infer nothing, i.e., ∅. At this stage, the number of premises in each combination is noted. That is:



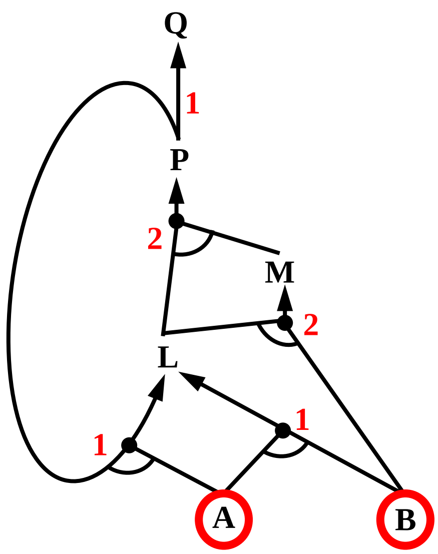
Premise	Conclusion	Premise Count
P	Q	1
$L \wedge M$	P	2
$B \wedge L$	M	2
$A \wedge P$	L	2
$A \wedge B$	L	2

Agenda: A, B

Inferred: ∅

International Journal of Scientific & Engineering Research Volume 11, Issue 4, April-2020
ISSN 2229-5518

We then start by processing agenda item A. that is, taking A and decreasing count for horn clauses in which A is a premise, we have:



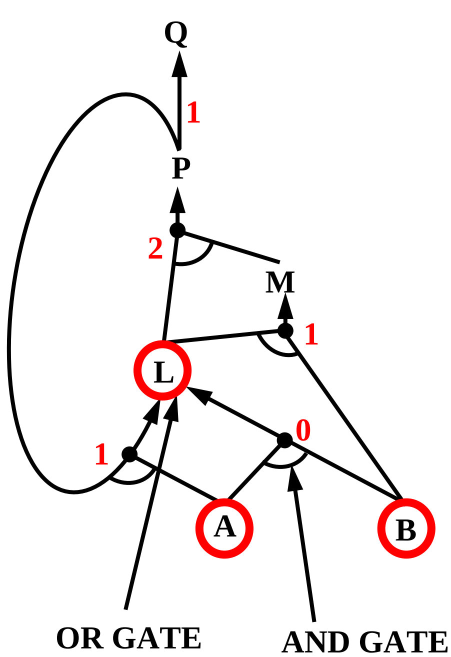
Premise	Conclusion	Premise Count
P	Q	1
L ∧ M	P	2
B ∧ L	M	2
A ∧ P	L	1
A ∧ B	L	1

Agenda: B

Inferred: A

Notice that the number of premise count in premise A ∧ P and A ∧ B have both reduced to one (1) in each case because agenda item A has been taken and we decrease count for horn clauses in which A is a premise. In those cases, we say, we pop A or inferred A. We then process agenda item B again by decreasing count for horn clauses in which B is a premise which in this case are A ∧ B and B ∧ L.

We then add L to the agenda and A and B are inferred. Then the table look like this.



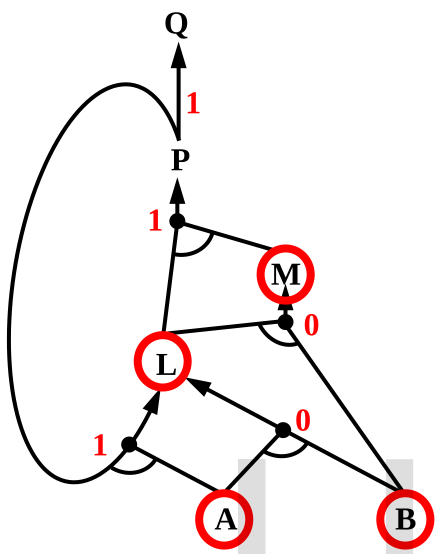
Premise	Conclusion	Premise Count
P	Q	1
L ∧ M	P	2
B ∧ L	M	1
A ∧ P	L	1
A ∧ B	L	0

Agenda: L

Inferred: A, B

Further, we process agenda item L by decreasing count for horn clauses in which L appears as a premise, which of course are L ∧ M and B ∧ L. B ∧ L ⇒ M now has its premise fulfilled and we then add M to agenda.

The new table is:

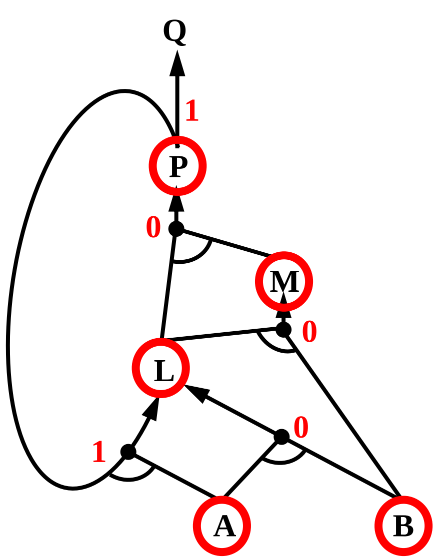


Premise	Conclusion	Premise Count
P	Q	1
L ∧ M	P	1
B ∧ L	M	0
A ∧ P	L	1
A ∧ B	L	0

Agenda: L

Inferred: A, B, L

At this stage, we add P to the agenda since we have inferred A, B, L, M. Agenda item P is then processed by decreasing count for horn clauses in which P is premise. In this case, we have P ⇒ Q and A ∧ P, at which point P has now fulfilled premise and Q is then added to the agenda as shown in fig. However, since L is already inferred, (It will not be inferred the second time) we process agenda item Q and finally q is inferred.

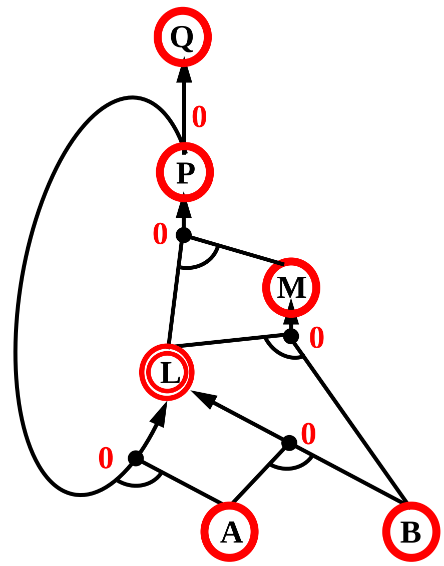


Premise	Conclusion	Premise Count
P	Q	1
L ∧ M	P	0
B ∧ L	M	0
A ∧ P	L	1
A ∧ B	L	0

Agenda: M

Inferred A, B, L, M

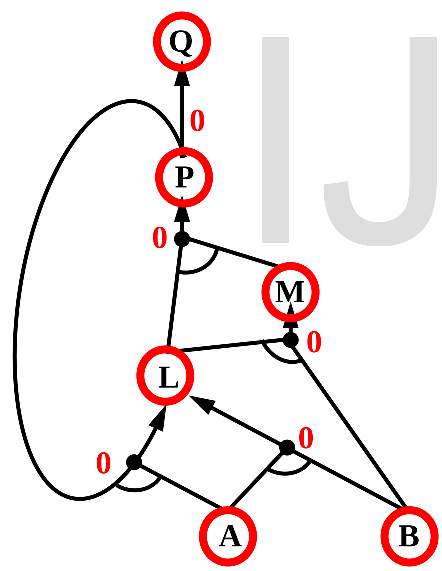
International Journal of Scientific & Engineering Research Volume 11, Issue 4, April-2020
ISSN 2229-5518



Premise	Conclusion	Premise Count
P	Q	0
L ∧ M	P	0
B ∧ L	M	0
A ∧ P	L	0
A ∧ B	L	0

Agenda: P

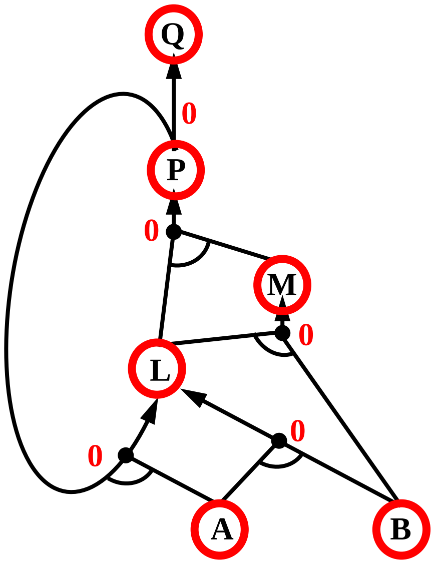
Inferred: A, B, L, M, P



Premise	Conclusion	Premise Count
P	Q	0
L ∧ M	P	0
B ∧ L	M	0
A ∧ P	L	0
A ∧ B	L	0

Agenda: P

Inferred: A, B, L, M, P, L already inferred and will not be inferred the second time



Premise	Conclusion	Premise Count
P	Q	0
L ∧ M	P	0
B ∧ L	M	0
A ∧ P	L	0
A ∧ B	L	0

Agenda: Q

Inferred: A, B, L, M, P, Q

5.0 Conclusion

Knowledge representation is a very important concept in expert systems and artificial intelligence (AI) in general. It involves considering intelligent (expert) systems and how it presents knowledge. Knowledge representation can best be understood in term of the roles it plays based on the task at hand. Knowledge representation is a medium for pragmatically efficient competition, that is, the computational environment in which thinking is accomplished. Pragmatically, a representation provides for organizing information to facilitate making the recommended inferences. Knowledge representation is a medium of human expression, that is, a language in which are say things about the world. In this paper, we discussed knowledge representation using inference rule and forward chaining. The paper demonstrates the use of inference rule in explaining forward chaining using an admission process based on some premises or antecedents to derive the conclusion. Some propositions or atomic sentences consisting of logical operators AND and OR are also used to infer conclusions based on some truth of additional proposition symbols.

References

1. Stanojevic, M. and Vranes, S. (2007). Knowledge Representation with SOUL, Expert Systems with Application, Vol. 33, pp. 122-134.
2. Davis, R., H. Shrobe, and P. Szolovits (2002). What is a Knowledge Representation? AI Magazine, pp. 17 – 33.
3. Nevo, D., I. Benbasat, and Y. Wand (2003). Exploring Meta – Knowledge for Knowledge Management Systems: A Delphi Study. In Proceedings of the 24th Int'l Conference on Information Systems. (ICIS'03), Paper 37, pp. 439 – 449.

4. Owoided, H. H. and M. M. Qasem (2010). Developing Rule-Case-Based Shell Expert System. In Proceedings of the Int'l Multiconference of Engineers and Computer Scentists, Hong Kong, March 17-19, 2010.
5. Streitberg, G. S. T., P T. Bwititi, L. Angel, A. Kenneth and K. A. Sikeris (2009). Automation and Expert Systems in a Core Clinical Chemistry Laboratory, The Association for Laboratory Automation.
6. Islam, R. and K. Anderson (2015). A Web-Based Belief Rule-Based Support System to Predict Flood, WAS2015, Brussels, Belgium, Dec. 11-13, 2015.
7. Baral, C. and S. Liang (2012). From Knowledge Represented in Frame – Based Languages to Declarative Representation and Reasoning via ASP. In Proceedings of the 13th Int'l Conference on Principles of Knowledge Representation and Reasoning, pp. 413 – 423.
8. Verhodubs, O. and J. Grundspenkis (2012). Towards the Semantic Web Expert System, Application Computing System, Vol. 44, No. 1, pp. 116-123.
9. Sowa, J. E. (2001). Knowledge Representation: Logical, Philosophical and Computational Foundations. Computational Linguistics, Vol. 27, No. 2, pp. 286-294, Shapiro, S. C. (ed.) (2001).
10. Hogo, M. A., K. Foused and F. Mousa (2009). Web-Based Expert System for Civil Service Regulations: RCSES. Int'l Journal of Computer and Information Security, Vol. 6, No. 3, pp. 7-16.
11. Claudiu-Leonardo, S. (2013). A Study Regarding the Use of Expert Systems in Economic Field. Procedia Economics and Finance, Vol. 6, pp. 385-391.
12. Tanwar, P., T. V. Prasad and K Datta (2012). Hybrid Technologies for Effective Knowledge Representation and a Comparative Study. Int'l Journal of Computer Science & Engineering Survey, Vol. 3, No. 4, pp. 43-57.
13. Patel, A and S. Jain (2017). Formalisms of Representing Knowledge. In Proceedings of the 6th Int'l Conference on Smart Computing and Communications (ICSCC'17), 7-8 December, National Institute of Technology, Kurukshetra, 136119, Haryana, India.
14. Sagheb-Tehrani, M. (2005). Expert Systems Development: Some Issues of Design Process. ACM SIGSOFT Software Engineering Notes, Vol. 30, No. 2, pp. 1-5.
15. Buchanan, B. (2005). A Brief History of Artificial Intelligence. AI Magazine, Vol. 26, No. 4, pp. 53-60.
16. Bahrami, M. (2008). A New Method for Knowledge Representation in Expert System's (XMLKR). In Proceedings of IEEE First Int'l Conference on Emerging Trends in Engineering and Technology, pp. 326-331.
17. Jose, A. and A Prasad (2011). Design and Development of a Rule Based Expert System for AACR: A Study o the Application o Artificial Intelligence Techniques in Library and Information Field, VDM Verlag, Saarbrȕcken, Germany.
18. Nalepa, G. J (2016). Diversity of Rule-Based Approaches: Classical Systems and Recent Applications, AVANT, Vol. VI, No. 2, pp.104-116.

19. Hu-Chen, L., Y. Jian-Xiu, L. Z. Wu and I. Guaagdong (207). Fuzzy Petri Nets for Knowledge Representation and easoning: A Literarture Review. Emerging Applications of Artificial Intelligence, pp. 45-56.
20. Napela, G. J., K. Khuza and K. Kaczor (2013). Proposal of an Information Engine Architecture for Business Rules and Processes. Artificial Intelligence and Soft Computing. In Proceedings of the 12th Int'l Conference, ICAISC 2013, Zakopane, Poland, June 9-13, 2013, Lecture Notes in Artificial Intelligence, Vol 7895, pp. 453-464.
21. Stevens, R., C. A. Gable and S. Bechhafer (2000). Ontology-Case Knowledge Representation for bioinformatics. Briefings in Bioinformatics, Vol. 1, No. 4, pp/ 398-414.
22. Luger, G. (2009). Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 4th ed., Pearson Education.
23. Mandis, D. (2012). Artificial Intelligence in Supervised Learning. In Proceedings of the 11th WSEAS Int'l Conference on Artificial Intelligence, pp. 213-218.
24. Rusell, S. and P. Norwig (2009). Artificial Intelligence: A Modern Approach, 3rd ed., Prentice-Hall.
25. Al-Ajulan, A. (2015). The Comparison between Forward and Backward Chaining. Int'l Journal of Machine Learning and Computing, Vol. 5, No.2, pp. 106-113.
26. Stephan, G., H. Pascal, and A Andres (2007). Knowledge Representation and Ontologies. Semntic Web Services, Springer, Berlin, Heiddberg, pp. 51-105.
27. George, F. (2008). Artificial Intelligence: Structures and Strategies for Complex Problem Solving. Addison-Wesley Publishing Company Inc.
28. Baget, J. (2004). Improving the Forward Chaining Algorithm for Conceptual Graphs Rules: Principle of Knowledge Representation and Reasoning. In Proceedings of the 9th Int'l Conference (KR2004), pp. 407-414.
29. Cabrera, M. M. and E. O. Edye (2010). Integration of Rule-Based Expert Systems and Case-Based Reasoning in an Acute Bacterial Meningitis Clinical Decision Support System
30. Kesarwani, P. and A. K. Misra (203). Selecting Integrated Approach of Knowledge Representation by Comparative Study of Knowledge Representation Scheme. Int'l Journal of Scientific and Research Publications, Vol. 3, No. 2, pp. 3150-3153.
31. Napela, G. J., S. Bobak, A. Ligeza and K. Koczor (2011). Formalization and Modeling of Rules Using the XTT2 Method. Int'l Journal of Artificial Intelligence Tools. World Scientific, Vol. 20, No. 6, pp. 1107-1125.