An Approach to Strengthen Expert System Shell with Knowledge Illustration Established on Peak of the Fuzzy Logic

Alauddin Alomary[1] and Mohammad Jamil[2]
[1]Department of Computer Engineering, University of Bahrain, Bahrain
[2]Department of Math and Computer, Qatar University, Qatar

Abstract: *Several tools are used to develop expert systems with the help of either programming or developing a special shell. A good expert system shell is projected to hold uncertainty properly for use in attractive domains of applications that deal with imprecise information. In this paper, we have implemented and tested a shell using new inference methods with the support of fuzzy logic to achieve our goal.*

Keywords: *Fuzzy expert system, uncertainty factor, possibility theory, fuzzy operators.*

Received November 29, 2004; accepted May 24, 2005

1. Introduction

Several expert systems have been developed and are being used by various applications. These expert systems have been developed either using programming logic or shell. Expert system shells deeply simplify and accelerate the improvement of expert systems [9]. Many expert system shells are now available ranging from simple rule-based shells such as Rule Master to more sophisticated hybrid tools such as KEE [28]. Much of the human knowledge is imprecise [30] and often, human thinking and reasoning involve inexact information [26]. To express such inexact information or uncertainty in an expert system is significant. However, the management of uncertainty, in expert system shells is still a challenging design issue. A good expert system shell is expected to handle uncertainty properly for use in interesting domains of applications that deal with imprecise information.

1.1. Significant Works

In designing any expert system, one of the key trouble is to supervise facts of different forms. Lacking any unique theory to manage all the forms as a whole, different approaches have been proposed with their own zones of applicability. The principle ones are:

- Stanford Certainty Factor (CF) calculus [36].
- Bayesian probability.
- Dempstar-Shafer (DS) theory of evidence [16, 35].
- Fuzzy logic [42].

Fuzzy logic is used to get approximate answer when no exact answer is possible. Fuzzy Set represents the basis of Theory of Possibility. Several approaches based on probability theory, Dempster-Shafer theory and possibility theory [10, 42] for handling uncertainty have been pursued [6, 10]. Several probability theory based expert systems have also been developed [14, 19]. However, many of such systems can handle only the uncomplicated problems. This is due to numerous probability assessments required to construct a functioning knowledge base. Further knowledge acquisition and encoding for such systems are difficult. Nonetheless, probability theory has a well-formalized methodology and it is universally applicable provided the dubious assumption about the underlying distribution holds.

Dempster-Shafer theory provides for expressing ignorance explicitly and does not restrict belief in the negation of a hypothesis. However, the complexity of the implementation of Dempster-Shafer theory has restricted its application in expert systems. Consequently, very few Dempster-Shafer based expert systems have been developed.

In addition to the systems described in [6, 10, 14, 22], three main fuzzy shell were implemented and used widely. These shells are: FLOPS [37], FuzzyCLIPS [31] and Fril [5]. FLOPS, a shell that comes in both fuzzy and non-fuzzy flavor and was written by William Siler [37]. FLOPS was derived from the OPS family, created by Charles Forgy of the Carnegie Mellon AI Department. FuzzyCLIPS is based on NASA's CLIPS expert system [31] language; and Fril is based on PROLOG, the dominant AI language in Europe. All three of these are rule-based and data-driven (non-procedural). FLOPS and Fuzzy CLIPS employ IF-THEN rules and forward chaining; Fril employs Horn clauses and backward chaining. Although these tools are known and used widely, however, they have some limitations. For example, FLOPS, need a variety of unconditional commands. Also, operational modes in FLOPS are more complex; procedural command mode versus data-driven run mode, and parallel versus sequential rule-firing modes. Finally, both ambiguities and contradictions can occur when working with FLOPS. Resolution of contradictions requires an appropriate theory of possibility and necessity that differs from conventional theory. The FuzzyCLIPS, need some enhancements in calculating the certainty factors, and in the way used to defuzzify the fuzzy facts as well as it needs to make fuzzy values as a standard types [31].

Many other commercial/research knowledge based shells have also incorporated fuzzy reasoning, for example LEONARDO [24], CUBICALC [15], TILSHELL [38], SYSTEM Z-II [26], CADIAG-2 [1, 2], RUM [11], FESS II [12], REVEAL [23], FESP [39], Fault [40] and FuzzyShell [32]. Most of these shells were designed for a small-scale expert system and were built on top of traditional non-fuzzy systems [32].

1.2. Thought and Plan in our Approach

Our expert system shell uses possibility theory that affords to articulate the imprecise and qualitative human knowledge to construct knowledge base for intelligent program. It allows to express the vagueness in a natural style with linguistic constructs similar to likely, unlikely and more or less etc. These linguistic constructs are used by human experts to illustrate occurrences of events but appending this with the probabilities does not protect fuzziness and it vanishes the accuracy of the description of a concept. The theory of possibility developed by Zadeh [42] grants to express such vague conditions with accuracy and construct partial matching of facts possible through compositional inference. Therefore we used theory of possibility as an erective tool to control ambiguity in our expert system shell. In the several application like diagnosis, assessment, selection and risk analysis, the expert systems is capable of manipulating uncertainties.

Our developed shell utilizes the following:

- Several old and new inference methods on the support of fuzzy logic [19].
- Consumes rules as the knowledge representation scheme.
- Symbolize and control fuzzy terms and operators (i. e., modifiers).

Our shell has been utilized to extend several expert systems in a variety of applications with remarkable performance and it is capable of concluding imprecise information. Currently, the shell is being used to develop an image processing in the area of Geographic Information System.

2. Theoretical Aspects

Zadeh [42] has given the concepts of fuzzy set theory and we have used in our shell development which is discussed here in brief.

2.1. Fuzzy Set Theory

A fuzzy set A is formed by taking elements from U with a membership function μA defined as:
μ_A: U ----> [0, 1]. The fuzzy set is denoted as $[\mu_1/x_1, \mu_2/x_2, ..., \mu_n/x_n]$ where x_i e U and μ_i is the membership grade of x_i [12].

The characteristic function of a fuzzy set represents the grade of membership of its elements that is indicated by a membership value lying in the interval [0, 1]. Fuzzy set theoretic concepts due to Zadeh [42] used in the shell development is briefly reviewed below. Concepts such as tall, strong, smart etc. can be represented as fuzzy sets. The characteristic function of a fuzzy set, represents the grade of membership of its elements which is indicated by a membership value lying in the interval [0, 1].

Example 1: Concepts such as tall, strong, smart, etc. can be represented as fuzzy sets.
e. g., concept of tall = [0.0/140, 0.1/150, 0.3/160, 0.5/170, ..]. The value 0.5/170 means person is 50% tall if the height is 170 cm. Here, a value says 0.3/160 may be interpreted as a person is 30% tall if his/her height is 160 cm. Here, "tall" is a fuzzy term and linguistic construct "very" is a fuzzy operator (modifier).

In the rule-based expert system, a shell uses rules. A rule in the form *IF X THEN Y* means this will be triggered by matching X to infer Y. When the rules are expressed with fuzzy terms, the fuzzy conditional inference [1, 12, 33, 38] shown below can be assumed:

- Rule: IF m is X THEN n is Y.
- Fact: M is X'
- Concl: N is Y'

Where m and n are the names of objects, and X, X', Y, and Y' are fuzzy terms.

Example 2: Let's talk about people and "youthness" [18]. In this case, the set S (the universe of discourse) is the set of people. A fuzzy subset YOUNG is also defined, which answers the question "to what degree is person x young?" To each person in the universe of discourse, we have to assign a degree of membership in the fuzzy subset YOUNG. The easiest way to do this is with a membership function based on the person's age.

young (x) = { 1, if age (x) <= 20, (30 - age (x)) / 10, if 20 < age (x) <= 30, 0, if age (x) > 30}.
A graph of this is shown in Figure 1.

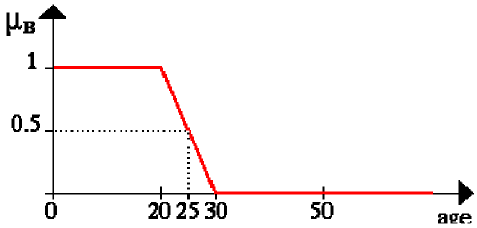

Figure 1. Membership function based on person's age.

Given this definition, Table 1 shows some example values:

Table 1. Example 2 values.

Person	Age	Degree of Youth
Johan	10	1.00
Edwin	21	0.90
Parthiban	25	0.50
Arosha	26	0.40
Chin Wei	28	0.20
Rajkumar	83	0.00

So, given this definition, we'd say that the degree of truth of the statement "Parthiban is YOUNG" is 0.50. Note: Membership functions almost never have as simple a shape as age (x). They will at least tend to be triangles pointing up, and they can be much more complex than that. Furthermore, membership functions so far are discussed as if they always are based on a single criterion, but this isn't always the case, although it is the most common case. One could, for example, want to have the membership function for YOUNG depend on both a person's age and their height (Arosha's short for his age). This is perfectly legitimate, and occasionally used in practice. It's referred to as a two-dimensional membership function. It's also possible to have even more criteria, or to have the membership function depend on elements from two completely different universes of discourse.

2.2. Inference Methods

Four inference methods will be discussed [1] then a new inference method in our expert system shell will be developed and utilized. The inference methods which have been discussed: Rs, Rg, Rsg, and Rgg are described using fuzzy relations [16].

1. The *Rs* method is based on the definition of *Rs* relation defined as:

$$Rs = \begin{cases} 1 \;\ldots\ldots\; \text{if } \mu_A(u) <= \mu_B(v) \\ 0 \;\ldots\ldots\; \text{if } \mu_A(u) > \mu_B(v) \end{cases}$$

The conclusion *B′* (in the fuzzy conditional inference) is obtained by: *B′* = *A′ o* Rs where *o* is the composition relation.

2. The *Rg* method uses the following relation:

$$Rg = \begin{cases} 1 \;\ldots\ldots\; \text{if } \mu_A(u) <= \mu_B(v) \\ \mu_B(v) \;\ldots\; \text{if } \mu_A(u) > \mu_B(v) \end{cases}$$

The conclusion *B′* is obtained by: *B′* = *A′* o *Rg*.

3. The *Rsg* method is based on the relation Rsg defined as:

Rsg [i, j] = Min (*Rs*[i, j],*Rg′*[i, j]),Where Min stand for the minimum, Rs is as given above and *Rg′* is constructed from R_g using 'not A' and 'not *B′* in place of A and B resp.
Here conclusion B' is obtained by: B' = A' o Rsg

4. The *Rgg* method uses the relation:

Rgg[i, j] = Min(*Rg*[i, j],*Rg′*[i, j]), B' is obtained as: *B′* = *A*' o *Rgg*.
We have referred our inference method Rmy and which is defined as:

$$Rmy = \begin{cases} \mu_B(v) \ldots \text{if } \mu_A(u) <= \mu_B(v) \\ 0 \;\ldots\; \text{if } \mu_A(u) > \mu_B(v) \end{cases}$$

The conclusion *B′* is obtained as: *B′* = *A′* o *Rmy*.

3. Overview of System Architecture

The proposed system architecture has four major components, as shown in Figure 2. The names of the components are:

- User Interface.
- Inference Engine [36].
- Knowledge Base (KB) [11, 13].
- Database (DB) [33, 34].

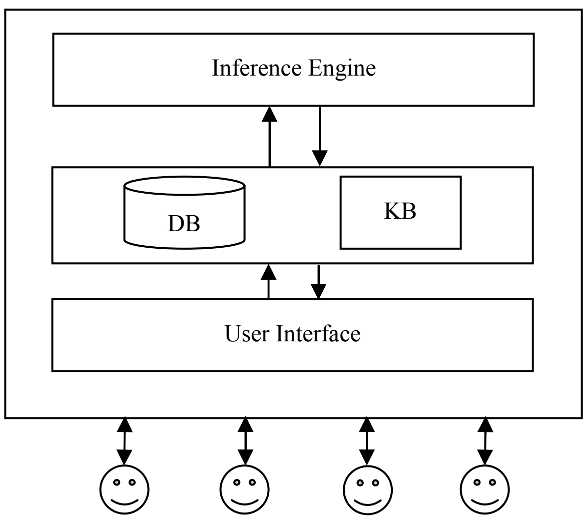

Figure 2. System architecture.

3.1. User Interface Component

The user interface reveals the superiority and forbearance of the system to acquire collection of acceptability. Therefore, a great deal of deliberation has been put in developing effectual and pleasant user surroundings to maintain client rest. We have lots of GUIs facilities and menus to keep client user-friendly. Figure 3 shows the main GUI menu. In this Figure, several groups are categorized and converted into functions. These functions are:

1. Knowledge BaseF unctions (KBF).
2. Database Functions (DBF).
3. Rules Functions (RUF).
4. Facts Functions (FAF).
5. Reasoning Functions (REF).
6. Concepts Functions (COF).
7. Fuzzy Operators (quantifiers) Functions (FOF).
8. Changing the Disk Drive Function (CDF).
9. Exit.

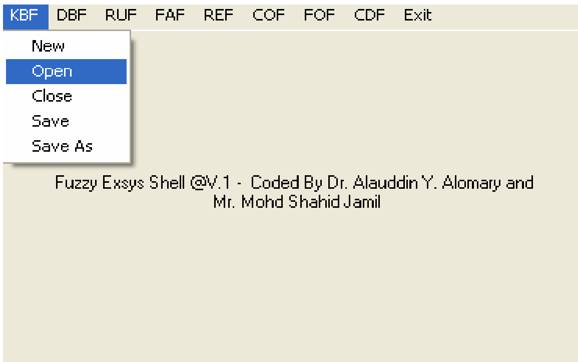

Figure 3. The fuzzy expert system GUI.

3.2. Database and Knowledge BaseComponent

A database deals with facts, fuzzy terms (concepts) and fuzzy operators. Facts symbolize the early data required to trigger rules and fuzzy terms are applied to signify concepts in a particular application where as fuzzy operators (modifiers) are concerned to direct fuzzy terms.

Our system intends for representing rules as the knowledge representation scheme [10, 12]. The rule has two parts, antecedent part and consequent part. The rule grammar is defined by the following syntax:

<Rule> ::= *IF <Antecedent_Part> THEN*
<Consequent_Part >
<Antecedent_Part > ::= <Exp>
<Consequent_Part> ::=
<Consequent> |
<Consequent_Part> &&
<Consequent_Part >
<Exp> ::= (<Exp>) | <Antecedent
> | <Exp >
<Operator> <Exp>
<Operator> ::= && | ||
<Antecedent> ::= object value
<Consequent>:: = object value

While designing a class or structure to manage huge application, we have considered many factors which can affect the system. These are some categories of factors:

- How to reduce access time?.
- How to use less space to save storage?.
- How to divide the knowledge into small logical component to develop easier programming logic?.
- How to preserve the class or structure of knowledge in reliable approaches?.

First and last factors give very important effect in structure. Special class structure is used to represent rules internally. For example, the antecedent of a rule is represented by class like:

class Antecedent {

Rule_No	*int*
Operation	*|| , && ,~*
Counter	*int*
Direction	*next, pre*

}

Field *Rule_No*is used to connect the antecedent with the consequent of the same rule. The *Operation* field indicates the type of the operation. The *counter* and *direction* fields are used to determine the order of the computation of the constituents of an antecedent. To clarify this aspect, consider the rule:

If (((A1 && (A2 || A3)) || A4) && (A5 || A6)) {C}

For A1: Operation = &&, Counter = 2, Direction = next. This shows that A1 is involved in the computation at the second around with the result of what comes to its right. Similarly, A2 and A3 will have counter = 1. This means that A2 and A3 are involved in the computation first with the operation '||', the result of which is ANDED (&&) with A1. This is a modified version of AO* algorithm [20].

3.3. The Inference Engine Component

The inference engine interrelate with knowledge and database to act further. It interacts with database to acquire data, then it searches for suitable rules in the Knowledge Base (KB) to achieve reasoning and then it stores the outcome reverse into the database to utilize further. Forward chaining [26] and Backward chaining [27] control strategies can be used in the inference engines. In our system, backward chaining has been implemented. A searching and matching function, selection and execution function, and entry and exit function have been also built. A selection function assists to select appropriate rule from set of applicable rules which guarantee a competent solution to the problem and avoid in-depth search. It also provides high degree of assurance to increase the strength of the conclusions and also help in making decision in case of conflict situations.

When a data is obtained then it will be either matched fully or partially by a fact from the database or has to be inferred. When it is inferred, the data either exists in the knowledgebase and database or a dialogue is initiated with the user to acquire missing data. Such data will either be fuzzy or crispy. For non fuzzy object, the system immediately requests the user regarding the antecedent in its accurate form. If the user answers, this will make it easy to the system to enter the certainty factor for it. For fuzzy object, the user may enter a concept such as "Tall" or a value known to him then the computation takes place depending on the value and the corresponding fuzzy set representing the concept tall. We can exemplify with this.

Example: Let extend the example of the section 2, suppose the required fact be "X is Tall" and corresponding fuzzy set (F) = [0/140, 0.1/150, 0.3/160, 0.5/170, ... , 1/220].

If the user puts the value 170 then the corresponding membership grade, 0.5, is used to carry the computation and easily can be evaluated. When the user enter 165 which not present in the fuzzy set but it is between 160 and 170 then its membership grade can be accomplished using following technique. Obtain the first value L such that it is the largest value less than v, and assign the value next of L to H. For v =160, L = 160, H = 170 and the membership grades corresponding to them, $\mu(L) = \mu(160) = 0.3$ and $\mu(H) = \mu(170) = 0.5$, Certainty Factor (CF) of the fuzzy fact is calculated like.

$CF = \mu(L) + [((v - L) / (H - L)) * (\mu(H) - \mu(L))] = 0.3 + [((165 - 160) / (170 - 160)) * (0.5 - 0.3)] = 0.4$. Thus, the obtained fuzzy fact is: X is Tall with CF = 0.4. For v equal to or less than 140, CF is $\mu(140)$. For v equal to or greater than 230, CF is $\mu(230)$. As we have mentioned earlier, the system supports a backward chaining mechanism. The algorithm which implements this mechanism is as follows:

```
Algorithm Used:
    Deduce(Rule_No, Chain_length,Goal)
     Begin{4}
       Insert_Into_Examined_list (Rule_No)
       Read_Rule_From_KB (Rule_No)
       While (antecedents are not all checked)
       Do
        Begin {3}
         A := Next_Antecedent
         If (NOT Is_It_Fact (A)) Then
         If ( Chain_length < Max ) Then
          Begin{2}
           I := 0
           Do
           I := Is_It_Deduced (A,I,Rule_N)
           Until (I = -1) OR
            (Not_Examined_Yet (Rule_N, A))
           If ( I <> -1 ) Then
             Begin {1}
              Do
                Deduce (Rule_N,Chain_length
                      + 1,A)
                Do
                 I:= Is_It_Deduced (A, I,
                                 Rule_N)
                 Until (I = -1) OR
                  (Not_Examined_Yet (Rule_N,
                                   A))
                 Until (I = -1)
             End {1}
             Else
             Ask_User (A)
          End {2}
          Else
          Ask_User (A)
        End {3}
        Try_To_Derive_Conclusion
     End {4}
```

The meaning of the procedures or methods used in the algorithm is as follow: *Deduce ()* procedure works to realize rules, goals and the level from goal. *Insert_Into_Examined_list ()* procedure works to add rules into the temporary set of database check list. *Read_Rule_From_KB ()* procedure developed for fetching rules from knowledgebase to match with the nearest goal. *Ask_User ()* procedure requests client to contribute extra information regarding the rules and *Try_To_Derive_Conclusion ()* procedure produces the most appropriate outcome equivalent with the rules and also recommend finale.

3.4. Algorithm Description and Analysis

As we know that AO* Algorithm [20] supports the principle of Problem Reduction. This uses the backward technique to solve a problem and divide the problem into sub-problem to sub-problem until it reaches to the trivial primitive problems. The division and reduction of the problem can be represented easily using AND-OR graph/ tree. In this the root node will be the original problem and the leaf node match to primitive problem descriptions. The AND node symbolizes the sub-goal that must be accomplished and the OR node advises the option for the goal. Each node assign a numeric value h' to guesstimate cost of the path from itself to a solution node. As the successor of a node generated, h' values is calculated and it propagates backwards. Whenever it is a leaf or terminal node then label of the node is achieved. The value h' will be zero (0). At the visit of each node it search for the lower h' value to locate the current best estimated path.

This way, the above process is repeated till either the root node is labeled as solved or its h' value exceeds a futility limit in which case the process is terminated. The algorithm has been traced and h' values have also been mentioned to observe good estimated path to that node and the arrows indicate the current best paths. In our problem for the simplicity we have sustained an identical cost procedure so that each arc between a node and its successor has a cost of 1 as well as each and each AND arc with multiple successors has a cost of 1 for each of its components as shown in Figure 4.

The application of AO* algorithm to the fuzzy set theoretic expert system should produce or verify conclusions with a high degree of assurance in the strength of the conclusions. The h' value of a node in the AND-OR graph then recommends the degree of confidence in that node. When an event occurs, then it will investigate the antecedent and compare facts regarding the relationship among their constituents and then determine whether or not a rule should be triggered. Also it observes the estimate value to achieve ultimate target with optimal solution. To investigate antecedent part of the rules, it is clear that the relationships among their constituents can be captured by constructing an AND/OR tree. The aim will then be to seek for a result in the problem space represented as an AND/OR tree. To handle this we have used AO* algorithm. The following example illustrates the process. Suppose we have a Rule:

```
If (((A1 && (A2 || A3)) || A4) && A5)
    {
      C
    }
```

Then the AO* tree is shown in Table 2:

Table 2. AO * tree.

	Left Operand					Right Operand					Operators	Result	
	Type	No	CF	F'	F	Type	No	CF	F'	F		CF	FR
R1	A	2	...	...	...	A	3	...	...	...	OR	...	...
R2	A	1	...	...	...	R	1	...	...	...	AND	...	...
R3	R	2	...	...	...	A	4	...	...	...	OR	...	...
R4	R	3	...	...	...	A	5	...	...	...	AND	...	...

The first row of Table 2 represents a sub-tree whose node is an OR node with two leaves A2 and A3. Similarly for the second and third rows which represent an AND sub-tree and an OR- sub-tree respectively. The last row represents the root tree. It is observed, therefore, that the above structure instantly concludes the rule to be triggered. The result of each row is calculated as explained above. The result of the last row constitutes the result of the entire rule.

4. Application Exploited

Our system has been used in various application areas. Knowledgebase is created and stored through the function of the shell. A user will interact with the inference engine to acquire desired analysis and to achieve conclusion. The system will search and trigger the several rules in the sequence to achieve goal. Also the engine asks the user to add new rule or facts to find another optimal cost else the reasoning process will end. The proposed system was tested using knowledgebase of the students record. A student records are analyzed to construct a knowledgebase then the proposed expert system shell was used to select a particular specialization for the student. The knowledgebase consists from several rules with their certainty factor (CF). A sample of the knowledge base is shown below:

Rules:

1. If (English Score is GOOD AND Math Score is GOOD) THEN Management Requirement is SATISFIED (0.90) AND Engineering Requirement is SATISFIED (0.70).
2. If (Chemistry Score is GOOD AND Biology Score is GOOD) THEN Medicine Requirement is SATISFIED (0.90) AND Pharmacy Requirement is SATISFIED (0.80).
3. If (Chemistry Score is GOOD AND Biology Score is GOOD AND Math Score is REASONABLE AND Physics Score is REASONABLE) THEN Scientific Medicine Requirement is SATISFIED (1 .0).
4. lf (Math Score is GOOD AND Physics Score is GOOD AND Chemistry Score is REASONABLE) THEN Engineering Requirement is SATISFIED (0.9).
5. If (Math Score is GOOD AND Physics Score is REASONABLE) THEN Computer Requirement is SATISFIED (0.9).
6. If (INTERESTED in Biology AND Chemistry is NICE) THEN You Are INTERESTED in Medicine (0.90).
7. If (INTERESTED in Chemistry AND Biology is NICE) THEN You Are INTERESTED in Pharmacy (0.90).
8. If (INTERESTED in Physics OR INTERESTED in Chemistry) AND Math is NICE AND Biology Is NICE THEN You Are INTERESTED In Scientist Medicine (0.90).
9. If (INTERESTED in Physics AND INTERESTED in Math AND Chemistry Is NICE) THEN You Are INTERESTED In Engineering Department(0.90)
10. If (INTERESTED in Math AND Physics Is NICE) THEN You Are INTERESTED In Computer Department (0.90).
11. If (Medicine Requirement Is SATISFIED OR You

Are INTERESTED In Medicine) THEN Medicine Department Is SUITABLE (1.00).

12. If (Pharmacy Requirement Is SATISFIED OR You Are INTERESTED In Pharmacy) THEN Pharmacy Department Is SUITABLE (1.00).
13. If (Scientific Medicine Requirement is SATISFIED OR You Are INTERESTED In Scientific Medicine) THEN Scientific Medicine Department Is SUITABLE (1.00).
14. If (Engineering Requirement Is SATISFIED OR You Are INTERESTED In Engineering) THEN Engineering Department is SUITABLE (1.00).
15. If (Department Requirement Is SATISFIED OR You Are INTERESTED In Computer) THEN Computer Department is SUITABLE (1.00).

Results:

We have seen these results of a student's record from our system.

1. Management score is Good.
2. Chemistry score is Good.
3. Biology score is More-or-Less Good.
4. Not Interested in Biology.
5. Chemistry is Nice.
6. More-or-Less Interested in Chemistry.
7. Biology is Not Nice.
8. Math score is Plus Reasonable.
9. Physics score is Very Reasonable.
10. Interested in Physics.
11. Math is Very Nice.
12. Math score is Very Good.
13. Physics score is Good.
14. Chemistry score is Very Reasonable.
15. Very Interested in Math.
16. Physics Is Very Nice.

The result of each inference methods mentioned in section 2 (*Rs*, *Rg* , *Rsg*, *Rgg*, and *Rmy*) using the facts given in knowledgebase are given in Table 3 with the certainty factor.

5. Conclusion

Expert system shell is implemented and tested to correct stream and to inform student for appropriate choice of department according to their wish and grade. Our contribution includes the proposing and implementation of a new inference method (Rmy). Rmy was compared with the exiting four inference methods (Rs, Rg, Rsg, Rgg) to obtain the certainty factor of the consequent proposition with the help of fuzzy theory. The new inference method has reached to a clearer and straight forward conclusion and we find that the "Computer is very suitable" choice is the only uppermost precedence so that it will be effortless for the student to just decide on right preference. Therefore, the result drawn by the new inference method is more accurate with respect to the other methods.

The developed shell was implemented using Java language under Windows environment and is being used in GIS for the processing of image and some of the outcome is already achieved. It has been an effortless assignment for the developer of expert systems to apply the shell to its bursting potential.

Table 3. Inference results.

Method	Derived Conclusion
Rs	Medicine is suitable (0.81) Pharmacy is more or less suitable (0.77) Medical sciences is more or less suitable (0.77) Engineering is very suitable (0.81) Computer is very suitable (0.81)
Rg	Medicine is suitable (0.81) Pharmacy is more or less suitable (0.77) Medical sciences is more or less suitable (0.77) Engineering is suitable (0.81) Computer is suitable (0.81)
Rsg	Medicine is not suitable (0.9) Pharmacy is more or less suitable (0.77) Medical sciences is more or less suitable (0.77) Engineering is suitable (0.81) Computer is suitable (0.81)
Rgg	Medicine is not suitable (0.9) Pharmacy is more or less suitable (0.77) Medical sciences is more or less suitable (0.77) Engineering is suitable (0.81) Computer is suitable (0.81)
Rmy	Medicine is not suitable (0.81) Pharmacy is suitable (0.77) Medical sciences is suitable (0.81) Engineering is suitable (0.81) Computer is very suitable (0.81)

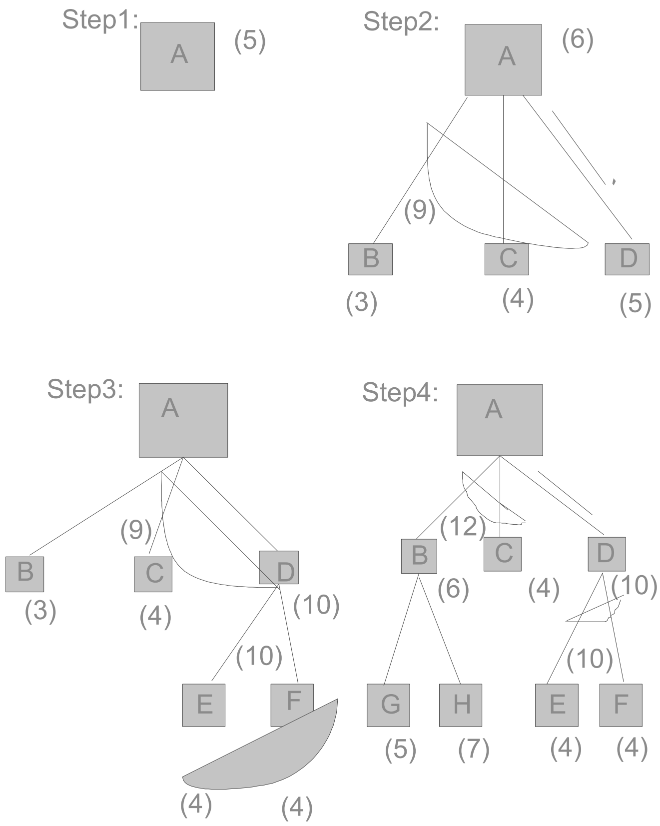

Figure 4. Operation of AO* algorithm, steps 1–4.

References

[1] Adlassnig K. P., "Fuzzy Set Theory in Medical Diagnosis," *IEEE Transactions System, Man and Cybernetics Archive*, vol. 16, no. 2, pp. 260-265, 1986.

[2] Adlassnig K. P. and Kolarz G., "Representation and Semiautomatic Acquisition of Medical Knowledge in Cadiag–1 and Cadiag-2," *Computer and Biomedical Result*, vol. 19, no. 1, pp. 63-79, 1988.

[3] Artificial Intelligence Section, *CLIPS Version 6.0 Reference Guide*, Johnson Space Center, vol. 1, 1993.

[4] Baldwin J., "Fuzzy Sets and Expert Systems," *Information Sciences*, vol. 36, no. 2, pp. 123-140, 1985.

[5] Baldwin J. F., Martin T. P., and Pilsworth B. W., *Fuzzy and Evidential Reasoning in Artificial Intelligence*, Research Studies Press Ltd., England, 1995.

[6] Bhatnagar R. K. and Kanal N. L., "Handling Uncertain Information: A Review of Numeric Non-Numeric Methods," *in Proceedings of the 1st Annual Conference on Uncertainty in Artificial Intelligence (UAI-85)*, Elsevier Science Publishing Company, Inc, pp. 3-26, 1986.

[7] Bhavani T., "From Rules to Frames and Frames to Rules," *AI Expert*, vol. 2, no. 10, pp. 31-38, October, 1989.

[8] Boley H., "Expert System Shells: Very-High-Level Languages for Artificial Intelligence," *Expert Systems*, vol. 7, no. 1, pp. 2-8, February 1990.

[9] Bonissone P. P. and Tong R. M., "Reasoning with Uncertainty in Expert Systems," *International Journal of Man-Machine Studies*, vol. 22, no. 3, pp. 241-250, 1985.

[10] Bonissone P. P., Gans S. S., and Decker K, "RUM: A Layered Architecture for Reasoning with Uncertainty," *in Proceedings of the 10th International Joint Conference on Artificial Intelligence*, vol. 1, pp. 891-898, 1987.

[11] Buckley J. and Tucker D., "Second Generation Fuzzy Expert System," *Fuzzy Set System*, vol. 31, no. 1, pp. 271-284, 1989.

[12] Correlius L., *Fuzzy Logic and Expert Systems Applications*, Academic Press, 1998.

[13] Cubicalc Hyperlogic Escondido, California, USA, 1990.

[14] Dempster A. P., "Upper and Lower Probabilities Induced by Multivalued Mapping," *Mathematical Statistics*, vol. 38, pp. 325-339, 1967.

[15] Dombal F. T., Leaper D. J., Staniland J. R., McCann A. P. and Horrocks J. C., "Computer-Aided Diagnosis of Acute Abdominal Pain," *British Medical Journal*, vol. 2, pp. 9-13, 1972.

[16] Fuzzy Logic Tutorial at FuzzyNet Online, available at: http://www.aptronix.com/FuzzyNet, May 2005.

[17] George J. K., and Tina A. F., *Fuzzy Sets, Uncertainty, and Information*, Prentice Hall, Englewood Cliffs, N J, 1988.

[18] Giarratano J. and Riley G., *Expert System: Principles and Programming,* Bouston, MA, PWS, 1994.

[19] Hall L., Szabo S., and Kandel A., "On Derivation of Memberships for Fuzzy Sets in Expert Systems," *Information Sciences,* vol. 40, no. 2, pp. 39-44, 1986.

[20] Harmons P. and Sawyer B., *Creating Expert Systems*, John Wiely, 1990.

[21] Horrocks J. C. and DeDombal F. T., "Computer-Aided Diagnosis of 'Dyspepsia'," *The America: Journal of Digestive Diseases*, vol. 20, no. 5, pp. 397-406, 1975.

[22] Jones P. K., "Reveal: An Expert Systems Support Environment," in Forsythe R. (Ed), *Expert Systems: Principals and Case Studies*, Chapman Hall, London, 1984.

[23] Leonardo Creative Logic Ltd., Uxbridge, UK, 1987.

[24] Leondes C., *Fuzzy Logic and Expert Systems Applicatio*ns, Academic Press, 1998.

[25] Leung K. S. and Lam W., "Fuzzy Concepts in Expert Systems," *IEEE Computer*, vol. 21, no. 9, pp. 43-56, 1988.

[26] Leung K. S., Wong W. S., and Lam W., "Applications of a Novel Fuzzy Expert System Shell," *Experts Systems*, vol. 6, no.1, pp. 2-10, 1989.

[27] Mathkour H., "Expert System Development Tools: A Case Study," *in Proceedings of the 1st International Conference on AI Applications*, Cairo, Egypt, 1992.

[28] Mizumato M., Fukami S. and Tanaka K., "Some Methods of Fuzzy Reasoning," in Gupta. M. M., Remade R. K., and Yager R. R. (Eds), *Advances in Fuzzy Set Theory and Applications*, North-Holland, pp. 117-136, 1979.

[29] O'Neil J., "Plausible Reasoning," *Australian ComputerJournal* , vol. 19, no.1, pp. 2-15, 1987.

[30] Orchard R. A., "FuzzyClips Version 6.04 User's Guide," FuzzyClips Web Site, Institute of Information Technology, National Research Council, Canada, available at: http://www.iit.nrc.ca/IR_public/fuzzy/fuzzyClips/fuzzyCLIPSIndex.html, June 2005.

[31] Pan J., DeSouza N., and Kak A., "FuzzyShell: A Large-Scale Expert System Shell Using Fuzzy Logic for Uncertainty Reasoning," *IEEE*

Transactions on Fuzzy Systems, vol. 6, no. 4, pp. 563-581, 1998.
[32] Payne E. C. and McArthur R. C., *Developing Expert Systems*, John Wiley, 1990.
[33] Rich E., *Artificial Intelligence*, McGraw-Hill, 1983.
[34] Shafer G. A., *A Mathematical Theory of Evidence*, Princeton University Press, 1976.
[35] Shortliffe E. G. and Buchanan B. G., "A Mathematical Model of Inexact Reasoning in Medicine," *Mathematical Biosciences*, vol. 23, pp. 351-379, 1975.
[36] Siler W. and Buckly J., *Fuzzy Expert System and Fuzzy Reasoning*, John Wiely, 2004.
[37] TILSHELL Togai Infralogic, Irvine, California, USA, 1990.
[38] Turksen I. B. and Berg M., "A Fuzzy Expert System for a Service Center of Spare Parts," *International Journal of Expert Systems with Application*, vol. 5, pp. 447-464, 1992.
[39] Whalen T., Schott B., and Ganoe F., "Fault Diagnosis in Fuzzy Networks," *in Proceedings of the International Conference Cybernetic Society*, Seattle, WA, pp. 35-46, 1982.
[40] Zadeh L. A., *Calculus of Fuzzy Restrictions in Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, Academic Press, New York, 1975.
[41] Zadeh, L. A., "Fuzzy Set as a Basis of Theory of Possibility," *Fuzzy Sets and Systems*, vol. 1, no. 1, pp. 3-28, 1978.
[42] Zadeh, L. A., "Fuzzy Set," *Information and Control*, vol. 8, no. 1, pp. 338-353, 1965.

Alauddin Alomary received his BSc from the University of Mosul, Iraq, in electronic and communication engineering, in 1980, his Master degree in communication engineering from University of Technology, Baghdad, Iraq in 1986, and his PhD in system and information engineering from Toyohashi University, Japan in 1994. Currently, he is an associate professor at the Department of Computer Engineering, College of Information Technology, University of Bahrain. His research interests include artificial intelligence, web intelligence, computer networks and ASIC systems design using VHDL. He has been actively involved in many research projects and published more than 25 papers. He is a member of IEEE and the Japanese Information Processing Society.

Mohammad Jamil received his MSc from Aligarh Muslim University with first class position, then he received another Master in computer science and application with first class position from Aligarh Muslim University, India, and was awarded Gold Medal from the president of India for securing first position top rank. He is working toward his PhD in artificial intelligence at VKS University. Currently, he is working as a senior lecturer in the Department of Computer and Math, Foundation Department, University of Qatar. He has over 16 years of diversified experience in computer science and information technology and published more than 17 papers.