

The Comparison between Forward and Backward Chaining

Ajlan Al-Ajlan

Abstract—Nowadays, more and more students all over the world need expert systems, especially in academic sectors. They need advice in order to be successful in their studies, and this advice must be provided by the academic management. There are two reasoning strategies in expert system, which have become the major practical application of artificial intelligence research: forward chaining and backward chaining. The first one starts from the available facts and attempts to draw conclusions about the goal. The second strategy starts from expectations of what the goal is, and then attempts to find evidence to support these hypotheses. The aim of this paper is to make a comparative study to identify which reasoning strategy system (forward chaining or backward chaining) is more applicable when making evaluations in expert management, especially in the academic field.

Index Terms—Artificial intelligence, expert system, forward and backward chaining, state space.

I. INTRODUCTION

In the academic field, some students need the best advice in order to improve their situation, and this advice must be provided by the academic management. The academic management should consider many important factors when providing such advice and some of these factors are the mode of study (part-time or full-time), is the student working or not, the number of courses taken last semester, the CGPA of last semester etc. In addition, as modern society has become more specialized, the need to have “expert systems” is increasing rapidly. These needs are visible in all areas of life [1].

These days, humans provide most “expert advice”. They have initiative and intelligence to adapt to a new model automatically and made decisions based on new data rapidly. However, humans also take a long time to learn things and can only perform one complete task at a time. For some of the reasons mentioned above, many companies have attempted to amass and collate the knowledge of experts about a special problem area. Some other reasons are as follows: (1) the knowledge of people who have particular expertise can be stored before they retire, so the company does not lose their knowledge and (2) the implementation of expert systems lightens the load on specialists. Such systems can be created to solve routine problems more easily and quickly [2]-[4].

The aim of this paper is to make a comparative study between forward-chaining and backward-chaining to select which of them is more applicable to the management expert

system. The comparison will be based on the adequacy of the reasoning strategy, which means to what extent the given strategy can be used to represent all knowledge required for the management expert system. The basic description of an educational management expert system is the use of expert system techniques in management to be used in the academic sector. What does academic management mean? In any academic sector there must be co-ordinator who co-ordinates specific programmes and should provide suitable advice for every student who needs it. The co-ordinator should consider various factors before he provides the advice, and neglecting any one of them might affect the advice negatively. This system will facilitate the task of the co-ordinator, and will help to provide the best advice for each case.

The importance of this study is: 1) no previous work has been done to compare forward-chaining and backward-chaining in the field of management in the academic sector, 2) to select the reasoning strategy paradigms required for a particular problem, 3) the study will contribute to the already on-going research on expert systems, namely, on managerial expert systems, 4) the research will lay a good foundation for other researchers intending to work in this area by providing a full picture of expert systems, their components, and related issues and 5) if this project is implemented, it will reduce the time and effort needed by students to meet the programme co-ordinator to obtain advice on a certain problem.

This paper is structured as follows: in the beginning, we present the related studies including a literature review of AI and expert system in Section II and Section III. Section IV highlights on the GAES. An implementation of GAES using the backward and forward-chaining techniques is presented in Section IV. An analysis and evaluation of GAES of this paper is presented in Section V. Finally, the conclusion is in Section VI.

II. ARTIFICIAL INTELLIGENCE

Artificial Intelligence (AI) began before the arrival of electronics; it began when philosophers and mathematicians such as George Boole (rightly regarded as one of the founding fathers of computing and information technology (1815–1864)) and others established the principles, which came to be used as the basis for AI logic. In 1943, AI started with the beginning of the invention of the computer [5].

AI is a part of computer science and provides machines with the ability to find the best solution for complex problems in a more human-like fashion [6], [7]. In other words, AI is concerned with programming computers to achieve tasks which are at present done better by humans, because they

Manuscript received August 4, 2014; revised December 8, 2014 This work was supported by the Qassim University.

Ajlan Al-Ajlan is with the Department of Information System Management, Qassim University, Saudi Arabia (e-mail: aajlan@qu.edu.com).

include much higher mental processes such as memory association, cognitive learning, and critical reasoning. Therefore, writing a program to perform a complex arithmetical calculation would not be seen as an AI action, whereas, writing a program to test a theory would [8], [9]. AI has search strategies that can be divided into two kinds of search as follows [3], [10], [11]:

A. Uninformed Search Strategies (Blind Search)

This explores the alternatives and events of a problem space one at a time. It is an algorithm approach and is able to distinguish a goal from other solutions, but the operators must be available to make decisions about any given goal. There are many uninformed search strategies and we will cover some of them, specifically, the three below [12]-[14]:

Breadth-First Search (BFS): This begins to search across the first stage of the problem space and it uses a number of arbitrary rules, for example, left to right (a search technique that looks for a solution along all of the nodes on one stage of a problem space before considering nodes at the next lower stage). If it cannot find any solution within this stage, it will drop down to the next stage and search in the same way as in Fig. 1. It will always discover the most direct way in first, and it guarantees a solution. Nevertheless, BFS is an inflexible algorithm that blindly types the problem space looking for a goal. It is easy to miss a solution [14]-[16].

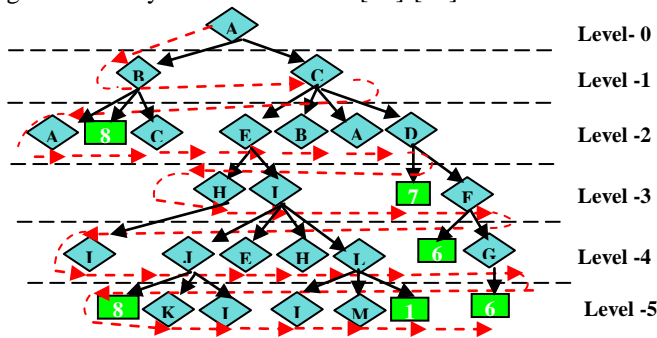


Fig. 1. Breadth-first search in graduate admission expert system.

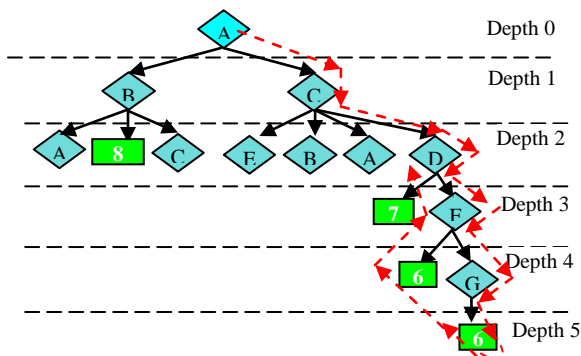


Fig. 2. Depth-first search in graduate admission expert system.

Depth-First Search (DFS): This is a systematic and intuitive strategy. It explores one path to its end (goal) before examining any other path. The process continues until it reaches a goal or dead-ends where the problem does not have a solution as in Fig. 2. DFS gives a guaranteed solution. It quickly searches deeply into a problem space, and uses less memory space. However, DFS is unsuitable for problems with a large search space and may not reach a goal in a reasonable

time. It is a complex type and is not optimal [13], [14], [17].

Uniform Cost Search (UCS): According to [3], UCS is similar to BFS, except that it accumulates costs to select the best node to expand. It stops when a goal is reached and when no other node expansion can provide a lesser amount of costly solution. UCS strategy discovers the solution with the least path cost, whereas BFS discovers the goal with the least depth as in Fig. 3 [17], [18].

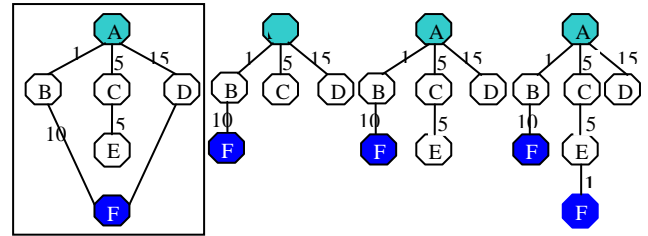


Fig. 3. A route finding problem. (1) The state space, showing the cost of each operator. (2) Progression of search.

B. Informed Search Strategies (Heuristic Search)

In Ref. [18], heuristic search was defined as “the study of the methods and rules of discovering and invention”. It involves trying all possible search paths to reach the nearer goal.

Best-First Search (BeFS): It guides the search towards the solution nodes of the problem space. It should be understood that the word "best" may not be close to the interpretation of the designer and it usually boils down to selecting the most efficacious way to meet the needs of the problem. BeFS is informed and it uses knowledge to guide the search to know where best to begin and how best to continue in search of a goal [14], [19].

Hill Climbing Search (HCS): This is a kind of DFS and always moves towards the goal. The solution space in this search is explored in DFS, following the path that appears to shorten the remaining distance to the conclusion. It uses heuristics to find which direction will take it closest to the goal. However, the constraint of this search is that if we arrive at a dead end, we will not be able to go backward [20].

A* Search: A search technique that finds minimal cost solutions, and is also directed towards goal states, is called an A* (A-star) search. It combines a DFS and BFR by cleverly backtracking a path when an obstruction to the straightest. If we can calculate the cost of each estimate in the above search model, then A* is the best choice. Such estimations signal the closeness to the goal. If estimation is accurate, then it always follows the lowest cost path [17].

III. EXPERT SYSTEMS

Over the past decade, Expert Systems (ES) have become the major practical application of AI research. Today, there are many useful systems in nearly every discipline in operation throughout the world. There are many application areas which have been successful for ES development. Many implementations of ES use a variety of tool, from hardware platforms to small private computers, using powerful Lots of Irritating Superfluous Parentheses machine workstations

[21]-[23]. ES are automated advisory programs that attempt to emulate the reasoning processes and knowledge of experts to reach the goal of particular problem. ES are of great importance to institution and commercial organisations due to the difficulties of gaining experts [24], [25].

There are many benefits of using ES that are [9]: (1) ES is available, 24 hours a day, never retires, and does not forget or make silly mistakes, (2) ES is faster and more available than human experts in responding to some applications, especially real-time systems, and (3) it has the ability to solve complex problems. Fig. 4 shows ES components and we will consider each component with a brief description [26]:

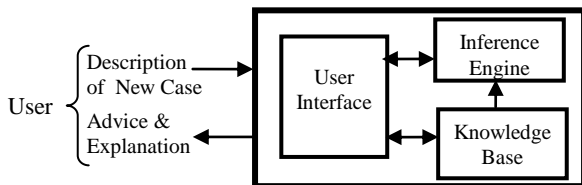


Fig. 4. Major parts of an expert system.

Knowledge Base (KB): KB contains the facts and knowledge to specific assignment, and rules for applying those facts. ES maintains expert's domain knowledge as KB.

Working Memory (WM): it is a part of an ES that contains the problem facts that are discovered during the session. WM contains two kinds of information which are: (1) supplied by the user and (2) inferred by the system.

Inference Engine (IE): it allows the ES to contact the knowledge stored in the KB and to run the system, as it draws an inference by relating user-supplied facts to KB rules. IE includes the following points: (1) works with facts, (2) searches the rules for a match and (3) adds the conclusion from the rules to the WM.

A Rule: An IF... THEN structure that logically relates information in the IF part to the information in the THEN part.

Example: IF (The code is SYS-MC24)
THEN (Course Is Data Mining).

The User Interface: It supports communication between the IE and the system user, and any interaction between the ES and the user with natural language [27].

A. Forward-Chaining

The solution to some problems naturally starts with the collection of information. Reasoning is applied to this information to obtain logical conclusions. For example, a doctor usually begins to diagnose a patient by asking him about the symptoms he or she suffers from, such as high blood pressure, temperature, headache, sore throat, coughing ...etc. Then the doctor uses this information to draw a reasonable conclusion or to establish a hypothesis to explore further. This way of reasoning is called in an expert data driven system, forward-chaining [11], [12].

Forward-chaining is a bottom-up computational model. It starts with a set of known facts and applies rules to generate new facts whose premises match the known facts, and continues 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 predetermined goal, and indicates that the inference moves forward from the facts toward the goal [28].

Forward-chaining is also used for improving and developing ES and modelling the human brain in AI.

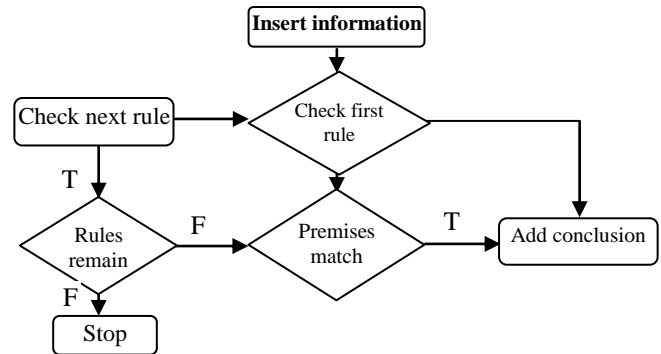


Fig. 5. Forward-chaining inference process.

The application of forward-chaining in a rule-based expert system is as in Fig. 5 and they are: 1) the user provides the system with information on the problem, 2) it seats the information in the working memory, 3) the inference engine scans the rules in some predefined sequence looking for one whose premises match the contents in the working memory. If it finds a rule, it adds the rule's conclusion to the working memory, 4) the system checks the rule again looking for new matches, 5) on the new cycle, rules that were previously fired are ignored, 6) this procedure continues until no match is found and 7) the working memory contains information provided by the user and inferred by the system.

Example: The following rules are samples from the admission system which will be used to compare the forward and backward reasoning.

Rule 1: IF (Bachelor's degree is British degree) AND (Final transcript available) AND (Prior graduate credit available) AND (Graduate GPR < 3)
THEN (Deny normal admission).

Rule 2: IF (Bachelor's degree is not British degree) AND (TOEFL > 5) AND (Final transcript available) AND (Prior graduate credit available) AND (Graduate GPR > 3) AND (GPR last 60 UG hours > 3) AND (Honour grade Valedictorian).
THEN (Admit full status and special decision)

Rule 3: IF (Bachelor's degree is not British degree) AND (TOEFL > 5) AND (Final transcript available) AND (Prior graduate credit available) AND (Graduate GPR > 3) AND (GPR last 60 UG hours > 3) AND (Not honour grade Valedictorian) AND (GRE < 1000) AND (GRE < 850)
THEN (Deny normal admission).

Assume that we have student who has applied for admission and his details are as follows: he/she has a British Bachelor degree, the final transcript is available, prior graduate credit is available and the Graduate Grade Point Ratio (GPR) is less than 3.

Firstly, the student information is inserted into the system, and then the student information and the premises of the first rule are compared. The system will find that there is no match between the rule premises and the student information

because the student has a Graduate GPR less than 3. Therefore, the conclusion (Deny normal admission) of the first rule will not be considered. Accordingly, checking the next rule, ‘Rule 2’, we find that there is a match between the R2 premises and the student’s information, so the conclusion (Admit full status and special decision) of R2 is added to the working memory. Now R3 is checked and compared with the student information; again we find that there is no match between the student information and the rule premises because the student (Not honour grade Valedictorian) has Graduate Record Exam (GRE) <1000 and has a GRE < 850. The system will then check all the rules until no rule remains to be checked. The advice will be stored in the working memory, which is “admit full status and special decision”

B. Backward-Chaining

The backward-chaining process is similar to hypothesis testing in human problem-solving. For example, a doctor may suspect some problems with a patient, which he then attempts to prove by looking for certain symptoms. This style of reasoning is modelled in an ES using a goal-driven search, and is called backward-chaining [11], [12]. it is a top-down computational design and starts with a goal or hypothesis and looks for rules to support the 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 [28].

The application of backward-chaining in a rule-based expert system is as follows: 1) the system checks the memory to see if the goal has been added. Another knowledge base may have already proved the goal, and therefore this step is necessary. The system searches its rules, and if the goal not has been previously proved, it looks for one or more that contains the goal in its THEN part. This kind of rule is called a goal rule, 2) the system then checks to see if the goal rule’s premises are listed in the memory. If the premises are not listed, they become new goals called sub-goals to be proved, which may be supported by other rules and 3) the system continues in this recursive manner until it finds a premise that is not provided by any rule; a “primitive”. When a primitive is found, the system asks the user for information about it. The system then uses this information to help prove both the sub-goals and the original goal.

Example: In the three rules above, the backward-chaining strategy starts with a list of decisions, generates all the facts for each decision, and asks the student about each fact. If the student wants to take “Admit full status and special decision”, then according to what we have explained above about the application of backward-chaining, the system checks the memory to see whether or not the goal has been added. If it has, then as a first step, the system checks whether or not the goal has been proved, and if not, it compares the student information (Admit full status and special decision) with the goal (after THEN part) for all rules. The system continues in a recursive manner until it finds a premise that is not provided by any rule (a primitive) when a primitive is found, the system asks the user for information about it and uses it, to help prove both the sub-goals and the original goal. In our example the primitives are the premises of R2: Bachelor's degree is not British degree and TOEFL>5.

IV. GRADUATE ADMISSION EXPERT SYSTEM (GAES)

Because only a few candidate applications require human consideration, the others being relatively straightforward accept/reject cases. Therefore a screening system would reduce the admissions tutor’s workload. An experienced admissions tutor is about to retire and we may wish to make his experience available to other novice admissions tutors.

The GAES baseis represented by a collection of facts and rules. The set of facts can be divided into domain facts (which are true for the whole problem domain) and case facts (which hold for the specific problem instance under scrutiny).

Facts:

- (F1) Language must met=550 TOEFL or US UG.
 - (F2) Transcript availability
 - (F3) Met language requirement and Honour student and etc.
 - (F4) Applicant 1 has grades US BSc and transcript >3.3
 - (F5) Applicant 2 TOEFL <550 and GER>1100.
- Domain Facts**
Case Facts

User starts with the given facts and rules and manipulates these to derive new facts and rules until a desired fact or rule is established. For example, the expert may decide to offer applicant 1 a place because of the following deductive process:

- 1) *Facts F2 and F3:* we conclude from these facts that the applicant has transcript and language met requirements.
- 2) *Rule R1:* Since the applicant has transcript and language met with Honours, he or she is checked to decide for conditional admittance.

To know how the facts will be used in order to produce a decision, Fig. 6 illustrates the relationship between the facts/inputs and the goals/outputs [29].

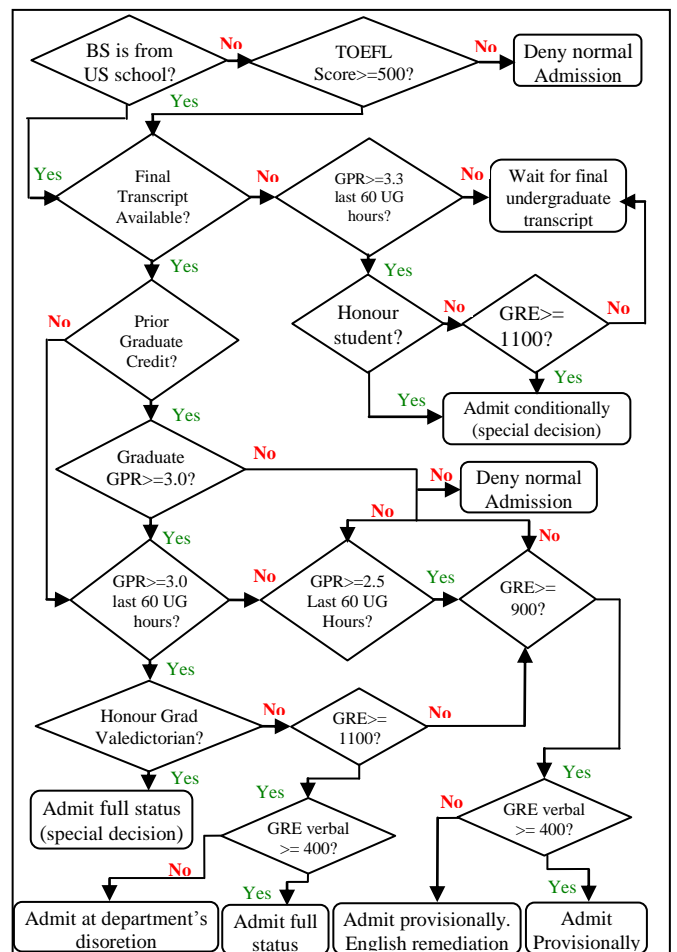


Fig. 6. Graduate admission expert system.

We can notice from the decisions in Table I that 1-5 are applicable for students who have provided a final undergraduate transcript. Decisions 6-7 are applicable for students who have not provided a final undergraduate transcript. Decision 8 is possible whether or not the undergraduate degree or language has been completed.

TABLE I: THE DECISIONS OF RULES IN GAES

No	Rule	Decision
1.	Admit with full status	applicable for students who have provided a final undergraduate transcript
2.	Admit with full status (special decision)	applicable for students who have provided a final undergraduate transcript
3.	Admit provisionally	applicable for students who have provided a final undergraduate transcript
4.	Admit provisionally with recommended remedial English coursework	applicable for students who have provided a final undergraduate transcript
5.	Refer to academic department for decision	applicable for students who have provided a final undergraduate transcript
6.	Admit conditionally (special decision)	applicable for students who have not provided a final undergraduate transcript
7.	Wait for transcript before making decision	applicable for students who have not provided a final undergraduate transcript
8.	Deny normal admission (could be reviewed)	possible whether or not the undergraduate degree or language has been completed

Again if we look at the recommendations and the decision tree above, the main factors for the process to proceed is the language requirement and the availability of the transcript, and this is based on the inputs section below. Hence, the tree could be constructed based on these main facts, and then one can proceed logically upon each decision requirement. Here is a grouped list of the attributes, identified from the flow chart, from which inputs can be written:

1. English language requirement.
 - Bachelor's degree from US institution (BS).
 - Test of English as a Foreign Language (TOEFL).
2. Completion of undergraduate program.
 - Final undergraduate transcript availability.
3. Performance as an undergraduate.
 - Grade Point Ratio (GPR) last 60 undergraduate (UG).
 - Honour student.
 - Honour graduate.
 - Valedictorian.
 - Salutatorian.
4. Performance in prior graduate work (if any taken).
 - Grade Point Ratio (GPR) for completed graduate work.
5. Performance on graduate admission tests.
 - Graduate Record Exam (GRE) overall score.
 - Graduate Record Exam (GRE) verbal score.

Therefore, the main decisions are based upon the language requirement to decide upon the conditional or full decision.

V. APPLYING THE BACKWARD AND FORWARD REASONING

In this section, we apply the backward & forward-chaining techniques to GAES, and will use Luger's rules. In this case, 12 rules will apply the backward and forward-chaining techniques. In GAES, there are many rules and the following 12 rules are a sample and will be used in this section.

Rule 1: IF (The applicant's UG is receiving a bachelor's degree from US) **OR**

- (The applicant's TOEFL score > 499)
THEN (The language requirement met)
Rule 2: IF (The language requirement met) **AND**
 (The applicant's final undergraduate transcript is available)
THEN (Full decision)
Rule 3: IF (Conditional decision) **AND**
 (The applicant's GPR on last 60 undergraduate hours > 3.29) **AND**
 (The applicant is an honour student)
THEN (Admit conditionally)
Rule 4: IF (The undergraduate GPR should be considered = true) **AND**
 (Applicant's GPR on last 60 undergraduate hours>2.49) **AND**
 (The applicant is not the recipient of special honours) **AND**
 (The applicant's GRE score > 899) **AND**
 (The applicant's GRE score < 1100) **AND**
 (The applicant's GRE verbal score < 400)
THEN (Admit provisionally- recommend English remediation)
Rule 5: IF (The applicant is not prior graduate work completed)
THEN (The undergraduate GPR should be considered = true)
Rule 6: IF (Conditional decision) **AND**
 (The applicant is not the recipient of special honours) **AND**
 (The applicant's GRE score < 1100)
THEN (Wait for transcript)
Rule 7: IF (The language requirement met) **AND**
 (Applicant's final undergraduate transcript is not available)
THEN (Conditional decision)
Rule 8: IF (Conditional decision) **AND**
 (The applicant's GPR on last 60 undergraduate hours < 3.3)
THEN (Wait for transcript)
Rule 9: IF (Conditional decision) **AND**
 (The applicant is not the recipient of special honours) **AND**
 (The applicant's GRE score > 1099)
THEN (Admit conditionally)
Rule 10: IF (Full decision) **AND**
 (The undergraduate GPR should be considered = true) **AND**
 (applicant's GPR on last 60 undergraduate hours>2.99) **AND**
 (The applicant is not the recipient of special honours) **AND**
 (The applicant's GRE score > 1099) **AND**
 (The applicant's GRE verbal score > 399)
THEN (Admit with full status)
Rule 11: IF (Language requirement met = no)
THEN (Deny normal admission)
Rule 12: IF (The undergraduate GPR should be considered=true) **AND**
 (The applicant's GPR on last 60 undergradat hours < 2.5)
THEN (Deny normal admission)

A. Backward-Chaining

The above rules are part of the knowledge base of the graduate admission system. Here we will perform the backward-chaining to the given knowledge base. We will place the top-level goal in the working memory, i.e. Admit full status. Three rules match the expression in working memory: R1, R3 and R8. We resolve conflicts in favour of the lowest numbered rule, and then R1 will fire. This causes the value of "Language requirements are met" and the premises of R1 to be placed in the working memory. The production system at the start of a consultation in GAES as in Fig. 7.

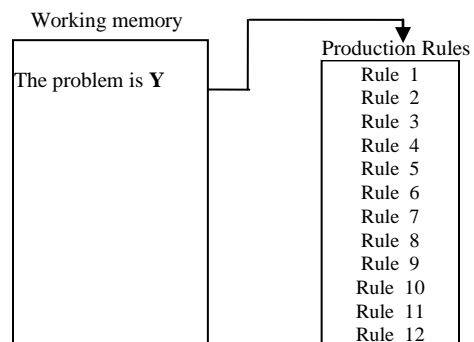


Fig. 7. The production system at the start of a consultation in GAES.

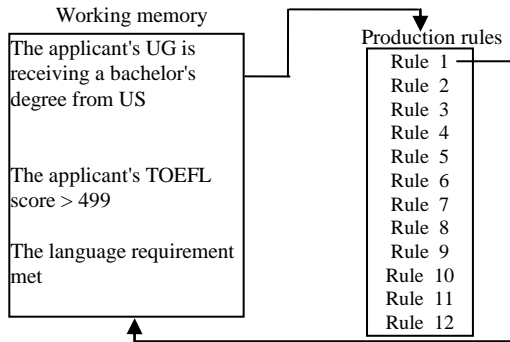


Fig. 8. The production system after R1 has fired.

There are two premises in R1 as in Fig. 8; one of them must be satisfied to prove the goal true. These are OR branches of the search graph, representing a decomposition of the problem “The language requirement met” into two premises: “The applicant's UG is receiving a bachelor's degree from US” and “The applicant's TOEFL score > 499)”. None of these two premises match with any rule conclusions, so in this situation, query the user directly about these premises.

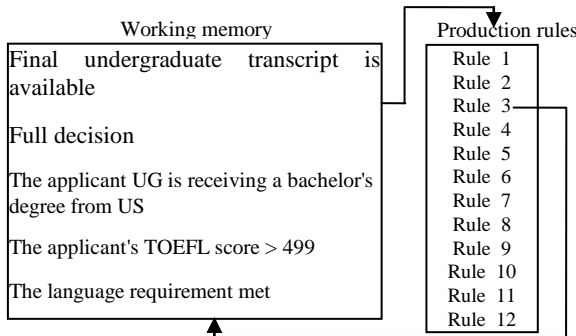


Fig. 9. The production system after R3 has fired.

Until this point, the system failed to determine the solution, so we try the next rule which is R3. R3 has two premises: “The language requirement met” and “Final undergraduate transcript is available”. These two premises will be placed in WM as shown in Fig. 9.

The first premise of R3 matches the goal of R1, which is stored with their values in the WM. The second premise of R3 does not match with any rule conclusions, and thus we must query the user. We will try the next rule, which is R8, because we failed to find the solution by firing R 1 and 3.

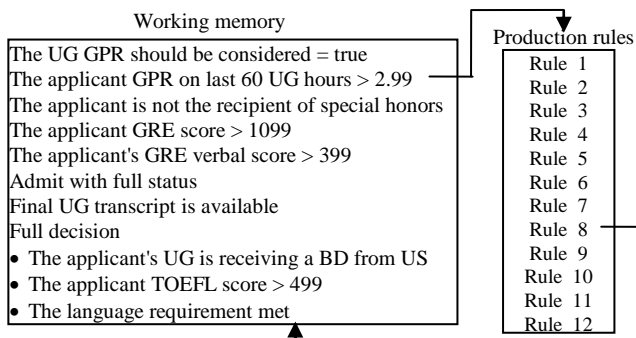


Fig. 10. The production system after R8 has fired.

Rule 8 has six premises which will be placed in the WM as in Fig. 10 above. The last four premises (The applicant's GPR on last 60 undergraduate hours > 2.99, The applicant is not the recipient of special honours, The applicant's GRE score >

1099, and The applicant's GRE verbal score > 399) do not match any of the rule conclusions so we will query the user. The first premise “Full decision” matches with the R3 conclusion, and we know this conclusion is proved. The second premise “The undergraduate GPR should be considered = true” matches the conclusion of R9, so the premise of R9 “The applicant is not prior graduate work completed” will be placed in the WM. This premise does not match any rule conclusions so we will query the user.

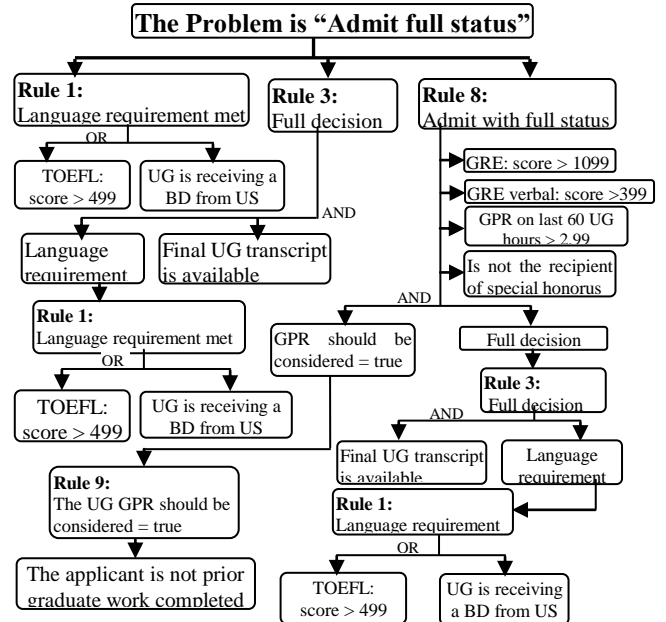


Fig. 11. The and / or graph searched in GAES with goals of rules 1, 3 and 8 matching the premises (The language requirement met) of rules (3) and the premise (Full admission) of rules (8).

Note that the conclusion of R8 is the same as solution we are looking for. Therefore, if the user confirms all these premises as true, the expert system will have successfully found the solution as in Fig. 11.

B. Forward-Chaining

In forward-chaining, the breadth-first search is more common. The algorithm for this compares the data of WM with the conditions of each rule in the rule base, using the ordering of that rule base. If the information in WM provides a rule's firing, the outcome is placed in WM and the procedure continues to the next rule. Assume that there is one student with the following details applying for admission, and that these details are:

- 1) The applicant's UG is receiving a bachelor's degree from US.
- 2) The applicant's final undergraduate transcript is available.
- 3) The applicant's GPR on last 60 undergraduate hours is 3.
- 4) The applicant is not the recipient of special honours.
- 5) The applicant's GRE score is 1100.
- 6) The applicant's GRE verbal score is 400.
- 7) The applicant is not prior graduate work completed.

In forward-chaining reasoning, we will start with examining the premises of all rules in order to see what information is “askable”. Askable is a piece of information that makes up (part of) the premise of a rule, but is not the conclusion of some other rule. In this example, we performed

a conflict resolution by taking the first rule found in the knowledge base, and then followed the results of that rule. The forward-chaining starts with an empty WM.

Thus, we examine if the premise of R1 “The applicant's UG is receiving a bachelor's degree from US” is askable, and the answer of this query is “true”, according to the applicant's details. Then “The applicant's UG is receiving a bachelor's degree from US” is placed in the WM, as is the conclusion of R1 “The language requirement met” because one of the two OR premises is true as in Fig. 12.

Working memory

The applicant's UG is receiving a bachelor's degree from US
The language requirement met

Fig. 12. After R1 is placed in WM.

Now the control moves to R2. The premise “The language requirement met” is not askable, so rule 2 fails and the control moves to R3. The premise of R3 “The language requirement met” is also not askable, so R3 fails and the control moves to R4. The premise of R4 “Conditional decision” is also not askable, so R4 fails and the control moves to R5. The premise of R5 “Conditional decision” is also not askable, so R5 fails and the control moves to the next rule. In Rules 6 to 8, the premises are not askable so they also fail and the control moves to R9. The premise of R9 “The applicant is not prior graduate work completed” is askable and according to the applicant's details the answer of this query is “true”, so “The applicant is not prior graduate work completed” and the conclusion “The undergraduate GPR should be considered = true” are placed in the WM as in Fig. 13.

Working memory

The applicant's UG is receiving a bachelor's degree from US
The language requirement met
The applicant is not prior graduate work completed
The undergraduate GPR should be considered = true

Fig. 13. After R9 is placed in WM.

The control now moves to next rule. The premises of R10, 11 and 12 are not askable so they fail. By the end of the first pass we find that two rules are proved: R1 and R9.

At this point all rules have been considered, so the search now returns with the new contents of WM in order to consider the rules in their correct order a second time.

Since R1 is proved, the second pass will start with R2, and then compare the premises of R2 with the WM. According to the content of WM, the premise “The language requirement met” is “true”, and the second premise “The applicant's final undergraduate transcript is not available” is askable, but its value is “false” because the applicant's final undergraduate transcript is available, according to the applicant's details. Therefore R2 has failed and the control moves to R3. The premise “The language requirement met” is “true” based on the content of the WM. The second premise “The applicant's final undergraduate transcript is available” is askable, and its value is “true” based on applicant's details, so this premise and the conclusion “Full decision” will be placed in the WM, and the control moves to the next rule. The R4, 5, 6 and 7 have the premise “Conditional Decision” which is the conclusion of R2, and R2 failed to match the applicant's details. This means that the 4 rules also will also fail because of one AND

premise, and the control moves to R8. The first two premises of R8 match the content of the WM, and the other 4 premises are askable. Based on the applicant's details, all four premises are true, so the goal of R8 is placed in the WM, and the control then moves to R10 because R9 was proved in the first pass. The premise of R10 does not match the content of WM, so it fails and the control moves to R11. The first premise of R11 “The undergraduate GPR should be considered = true” matches the content of the WM. The other premises of this rule are askable. Based on the applicant's details, the premise “The applicant's GPR on last 60 undergraduate hours >2.49” is “true”, the premise “The applicant is not recipient of special honors” is “true”, the premise “The applicant's GRE score>899” is “true”, the premise “The applicant's GRE score< 1100” is “false”, and the premise “The applicant's GRE verbal score < 400” is “false”. Since two AND premises are false, the rule fails and the control moves to R12. The premise of R12 “The undergraduate GPR should be considered = true” matches the content of WM. The second premise is askable but it does not match with the applicant's details, so this rule also fails and the second pass is completed.

By the end of pass two, all rules are fired and their premises are checked. We find that R8 matches the details of the applicant, so its goal is the best decision for this applicant, which is “Admit with full Status”.

VI. RESULT

The main points needed for a comparison of forward-chaining and backward-chaining are summarised in the Table II.

TABLE II: THE COMPARISON BETWEEN FORWARD-CHAINING AND BACKWARD-CHAINING

Forward-chaining	Backward-chaining
Starts with the initial facts.	Starts with some hypothesis or goal.
Asks many questions.	Asks few questions.
Tests all the rules.	Tests some rules.
Slow, because it tests all the rules.	Fast, because it tests fewer rules.
Provides a huge amount of information from just a small amount of data.	Provides a small amount of information from just a small amount of data.
Attempts to infer everything possible from the available information.	Searches only that part of the knowledge base that is relevant to the current problem.
Primarily data-driven	Goal-driven
Uses input; searches rules for answer	Begins with a hypothesis; seeks information until the hypothesis is accepted or rejected.
Top-down reasoning	Bottom-up reasoning
Works forward to find conclusions from facts	Works backward to find facts that support the hypothesis
Tends to be breadth-first	Tends to be depth-first
Suitable for problems that start from data collection, e.g. planning, monitoring, control	Suitable for problems that start from a hypothesis, e.g. diagnosis
Non-focused because it infers all conclusions, may answer unrelated questions	Focused; questions all focused to prove the goal and search as only the part of KB that is related to the problem
Explanation not facilitated	Explanation facilitated
All data is available	Data must be acquired interactively (i.e. on demand)
A small number of initial states but a high number of conclusions	A small number of initial goals and a large number of rules match the facts
Forming a goal is difficult	Easy to form a goal

VII. CONCLUSION

To conclude which of the reasoning strategies is better for the GAES, we will consider the recommendations of [18] for choosing either the forward or backward-chaining, and we will apply each of these recommendations to the GAES.

A. A Goal-Driven Search (Backward-Chaining) Is Appropriate for Problems in Which

A goal or hypothesis is given in the problem statement or can easily be formulated: In GAES, the goal or hypothesis is given in only one case, which is a student query about a specific goal before he applies for admission. This case is not the main purpose of the graduate system. The main purpose of the system is to obtain the information from the applicant, and to try to find a goal according to the applicant's details. In this case, the goal is not given and it is difficult to be formed, so this point is not applicable with the graduate system.

There are a large number of rules that match the facts of the problem, and which produce an increasing number of conclusions or goals: In GAES there are a large number of potential goals, but there are only a few ways to use the facts and given information of a particular problem instance.

Problem data is not given but must be acquired by the problem solver: In GAES, usually the data is given, refolding the details of the applicant, so this point is not applicable with the graduate system.

B. A Data-Driven Search (Forward-Chaining) Is Appropriate for Problems in Which

All or most of the data is given in the initial problem statement: As mentioned above in the third point of the goal-driven search, the data is given in the graduate system, which is the detailed information of the applicant. This data reflects the details of the applicant who wants admission, so this point is applicable with our system.

There are a large number of potential goals, but there are only a few ways to use the facts and given information of a particular problem instance: Yes, there are indeed a large number of potential goals, but few ways to use the facts and given information of this particular problem instance. For example, there are four rules whose goal is "Admit conditionally", but all four rules have different premises. Therefore, this point is applicable with the system.

It is difficult to form a goal or hypothesis: In GAES, the goal is difficult to be formed because the process of making the decision is complicated. Therefore, this point is applicable with the system.

From the above recommendations, we will conclude that the best approach for GAES is with forward-chaining.

REFERENCES

[1] B. McLaren *et al.* (January 2010). Supporting collaborative learning and E-discussions using artificial intelligence techniques. *International Journal of Artificial Intelligence in Education*. [Online]. 20(1). pp. 1-46. Available: <http://dl.acm.org/citation.cfm?id=1898078.1898079>

[2] F. George, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, Addison Wesley Pub Co Inc, 2008, ch. 5.

[3] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed., Prentice-Hall, 2009, ch. 1, pp. 2-18.

[4] D. Mandic, "Artificial intelligence in supervised learning," in *Proc. the 11th WSEAS International Conference on Artificial Intelligence*, 2012, pp. 213-218.

[5] H. Simon. (2010). An Introduction to the Science of Artificial Intelligence. *Oracle ThinkQuest*. Available: <http://library.thinkquest.org>

[6] A. Champandard. (2010). Artificial Intelligence Introduction. [Online]. Available: <http://ai-depot.com>

[7] M. Kolp. (2005). Knowledge systems, organizational knowledge management expert systems. *Université catholique de Louvain*. [Online]. Available: <http://www.cs.toronto.edu/~mkolp/lis2103/expsys2.pdf>

[8] P. Jackson, *Introduction to Expert System*, 3rd edition, Addison-Wesley, 1999, ch. 4, pp. 16-28.

[9] F. Moisiadis. (2004). Artificial Intelligence and Expert Systems. [Online]. Available: <http://www.comp.mq.edu.au>

[10] D. Kaminaris *et al.*, "PTME: A new expert system application for power transformer troubleshooting and maintenance," in *Proc. the 6th Conference on Artificial Intelligence*, Wisconsin, 2007, pp. 52-57.

[11] A. Jose and A. Prasad, *Design and Development of a Rule Based Expert System for AACR: A Study of the Application of Artificial Intelligence Techniques in Library and Information Field*, VDM Verlag, Saarbrücken, Germany, 2011.

[12] M. Sagheb-Tehrani, "Expert systems development: Some issues of design process," *ACM SIGSOFT Software Engineering Notes*, vol. 30, no. 2, pp. 1-5, 2005.

[13] V. Vishnevsky *et al.*, "Scalable blind search and broadcasting over Distributed Hash Tables," *Computer Communication*, vol. 31, no. 2, pp. 292-303, 2008.

[14] A. Cawsey. (2003). Introductory AI Course, Computers in Teaching and Learning. [Online]. Available: <http://www.cce.hw.ac.uk>

[15] R. Zhou and E. Hansen, "Breadth-first heuristic search," *Artificial Intelligence*, vol. 4-5, no. 170, pp. 385-408, 2006.

[16] C. Crespelle and P. Gambette, "Unrestricted and complete Breadth-First Search of trapezoid graphs in on-time," *Information Processing Letters*, vol. 110, no. 13, pp. 497-502, 2010

[17] T. Mori, "Japanese question-answering system using a* search and its improvement," *ACM Transactions on Asian Language Information Processing*, vol. 3, no. 4, pp. 280-304, 2005

[18] G. Luger, *Artificial intelligence: Structure and Strategies for Complex Problem Solving*, 4th edition, Pearson Education, 2009.

[19] A. Kishimoto *et al.*, "Evaluation of a simple, scalable, parallel best-first search strategy," *Artificial Intelligence*, vol. 195, pp. 222-248, 2013.

[20] W. Yan *et al.*, "An investigation into dynamic yard crane deployment and comparisons between hill-climbing and best-first-search algorithms," *International Journal of Computer Applications in Technology*, vol. 25, no. 3, pp. 254-264, 2008.

[21] L. Medsker and J. Liebowitz, *Design and Development of Expert Systems and Neural Networks*, 1st ed., Prentice Hall, 1993.

[22] J. Stannard, *Building Expert Systems in Prolog*, 1st ed., Springer, 2000, ch. 1, pp. 1-8.

[23] I. Dokas *et al.*, "Fault tree analysis and fuzzy expert systems: Early warning and emergency response of landfill operations," *Environmental Modelling*, vol. 24, no. 1, pp. 8-25, 2009.

[24] F. Bobillo *et al.*, "A semantic fuzzy expert system for a fuzzy balanced scorecard," *Expert System. Appreciation*, vol. 36, pp. 423-433, 2009.

[25] J. Giarratano and G. Riley, *Systems: Principles and Programming*, 4th edition, Course Technology, 2004.

[26] T. S. Kuhn, "The structure of scientific revolutions," *International Encyclopedia of Unified Science*, vol. 2, no. 2, pp. 274-276, 2006.

[27] R. McDuffie and P. Eugene, "Tax expert systems and future development," *The CPA Journal*, vol. 64, no. 1, 1994.

[28] I. Gaag and V. Lida, *Principles of Expert Systems*, Addison-Wesley, 1991, ch. 3, pp. 131-139.

[29] EXPertise2Go. (2009). Building and Using Expert Systems: a Mini-Course Introducing the e2gLite Expert System Shell, Cornell University Online Classes in Product and Service Design.



Ajan S. Al-Ajlan is an assistant professor in the Department of Information System Management at Qassim University-Saudi Arabia. He received his Ph.D degree in computer science, from De Montfort University - Britain (UK) in 2009. His research interests include e-learning systems & watermarking technology.