MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

Memo No. 381

October 14, 1976

Wumpus Advisor I. A first implementation of a program that tutors logical and probabilistic reasoning skills

by
James L. Stansfield
Brian P. Carr
and
Ira P. Goldstein

Abstract

The Wumpus Advisor program offers advice to a player involved in choosing the best move in a game for which competence in dealing with incomplete and uncertain knowledge is required. The design and implementation of the advisor explores a new paradigm in Computer Assisted Instruction, in which the performance of computer-based tutors is greatly improved through the application of Artificial Intelligence techniques. This report describes the design of the Advisor and outlines directions for further work. Our experience with the tutor is informal and psychological experimentation remains to be done.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. It was supported in part by the National Science Foundation under grant C40708X, in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75-C-0643 and in part by the Division for Study and Research in Education, Massachusetts Institute of Technology.

CONTENTS

ACKNOWLEDGEMENTS

The authors would like to thank John Avgoustis, William Clinger, Virginia Grammar, Robin Gross, Fred Knoll, Beth Levin and Neil Rowe who accomplished the preliminary implementation of the Wumpus Advisor while taking part in the Educational Technology course 6.860/SRE 220 at M.I.T

I. Introduction

The Wumpus Advisor grew out of a course we gave in Educational Technology to a small group of graduate and undergraduate students at M.I.T. Our goal was to explore a new paradigm in Computer Aided Instruction, in which the competence of computer-based tutors is greatly improved by applying Artificial Intelligence techniques to their design. We particularly wished to study the structure of Intelligent Computer Aided Instruction (ICAI) programs that incorporate an Expert module which allows the tutor to compare the student's response to those generated by the expert. In using the term ICAI and exploring the consequences for a tutorial program of the availability of an expert module, we follow the lead of John Brown, (Brown and Burton 1975), who has shown in his design of sophisticated instructional environments for electronics, the promise of this approach.

In order to experiment with this paradigm, an ICAI program for a simple game was implemented as a course project. The program serves as an Advisor to a player, offering advice and analysis at appropriate times. We chose Wumpus, a maze-exploration game, because it represented the next step in complexity beyond the tutor designed by Burton & Brown for West, a simple game on the Plato system for exercising arithmetic skills (Burton 1976). Wumpus is motivating and requires a variety of skills covering planning, plausible reasoning, decision theory and incomplete and uncertain knowledge.

The Wumpus Advisor was successfully implemented by the students in the course under Stansfield's supervision. The program was later improved and extended by Carr, who is continuing to work on the project. This paper describes the current state of the program which gives appropriate advice in English about the logic involved in choosing a best move. Four different levels of student are catered for but other than this broad distinction there is little student modelling. This aspect of the research is currently being developed.

By studying simple teaching situations and modelling them with programs that teach we gain insight into the processes underlying learning and teaching. The rich metaphors of computer programming help us to describe teaching and learning precisely and in detail while the discipline imposed by requiring a working program weeds out impractical ideas and points the way to better ones.

CAI programs need models of situations and students if they are to understand what is going on and act appropriately. We must provide them with practical procedures for making decisions about teaching and give them a precisely formulated knowledge of their subject matter so that they can interpret, model and act in a variety of teaching situations. They also need an expressive means of communication such as natural language, display screens and tablets for both interpreting the students behaviour and making effective responses.

Many early teaching programs and some current ones were "fact dispensing" machines. They used the "empty bucket" theory of learning, a trivial one in which the learner is simply a receptacle to be filled with facts. Although this theory may be decorated with extra rules to present facts in special orders or in clusters, it is very naive and hardly says anything at all about real learning. The key computing concept which it excludes is that of a process. The student should above all else be learning how to do something and should be participating in various activities toward that end. He is programming himself with the teachers assistance. By changing the paradigm from facts to procedures the whole enterprise is greatly enriched.

From this viewpoint we are forced to analyse the student's learning task and compare this with his behaviour. It becomes important to notice and correct the things he does wrong, forgets to do, does unnecessarily or does in the wrong order. Many ideas from Computer Science are of great significance to this. The student's task can be modularly decomposed into subtasks with individual goals. These subtasks can be organized as processes, coroutines or steps in a procedure. The vocabulary of Computer Science is rich in precise concepts for describing this. Similarly, his organization of information and methods must be examined and debugged. There are sufficient partially-formulated concepts in AI that deal with perception, natural reasoning, organising knowledge, planning and so on, for new descriptions to be made of the learning and teaching process.

The Wumpus Advisor develops the application of computers in education. It is the first version of a program which helps a student to learn a simple game called Wumpus (Yob 1975). Acting as an interface between the student and the game, it intervenes whenever the student's moves show that he needs advice. Advice is given as English discourse explaining in full the merits and faults of particular moves. Wumpus is played in a network of tunnels whose connections are initially unknown to the player. He must search this network avoiding dangers and trying to find and kill the dangerous and deadly Wumpus. Throughout play the advisor gives the student information about his immediate locality and evidence about nearby dangers. From this information it is possible to make plausible inferences and judgements which aid in avoiding dangers. The game is highly motivating to children and exercises several types of reasoning skill.

The game paradigm for advisors has also been researched by Burton using the game West (Burton and Brown 1975). Wumpus is a more complex game and is a natural next step. In general, games form excellent subject matter for advice giving. They are varied, provide motivation, and exist at many degrees of difficulty. Some, such as chess, have large bodies of advice associated with them in the literature. Games are often models of real-world situations and develop abilities that are useful in everyday life. Many of the strategies involved in the game of Go are of this nature.

There are five good reasons for using a simple game as the domain of an advice-giving program.

1. Closure

The rules are clearly defined. Since it is easy to describe what constitutes a legal move the student can always be expected to play within the rules even if he plays badly. This means that the advisor will be able to make sense of his inputs. With a less bounded domain it is easy for breaks in communication to occur because the program cannot understand the student.

2. Expertise

We can easily design an expert player for many simple but interesting games. An expert gives a precise procedural theory of the domain which we aim to teach.

3. Homogeneity

For simple games the same theory of good play applies at each move. The rules that the expert uses are good at all stages of the game. This gives generality to the teaching situation. A skill is being taught which is exemplified in different ways throughout the game.

4. Simplicity

It is easy to find simple examples of games well within programming capability.

5. Motivation

The student is motivated by a game when he may not be by traditional curricular domains.

These properties make it easy to sustain an interaction between the student and the teacher. Even with no advice-giving at all, the game scenario provides a continuing exchange. In a sense this is cheating for it makes it easy to write a "toy" program but the important point is that we can start from such a position and enhance the advice giving step by step. This is the way people learn games in any case, beginning with the rules and accumulating strategies which cover progressively more situations.

Our general methodology was to find a domain which the computer can deal with easily, which requires only simple inputs but which has a large set of states. Games fit this well. Electronics does too as Sophie, the electronics advising program (Brown and Burton 1975), shows. Sophie helps a student learn how to repair a faulty electronic circuit. A faulty circuit can be simulated. Moves correspond to measurements or alterations and, though there are only a few move types, the possible hypotheses that can be made about a faulty circuit are numerous and varied. Domains like geography or history are hard to use in a CAI program. They are very knowledge-oriented and tend not to be closed. Limited and well-structured aspects of them must be used if the domain is not to expand continually or the student is not to overreach the program's knowledge (see Collins 1975 for promising work in this direction).

A simple game like Wumpus makes the task of writing an advisor manageable but does not exclude important features of the teaching process. Models of the student, ways of using them to provide relevant advice, questions of motivation and of not overadvising, can all be studied even for a simple game. We have not programmed any student modelling facility yet in our advisor though the work we have completed is a preparatory step.

The student is doing several things when he plays Wumpus with the advisor. First, he is learning how to play Wumpus. An adaptation of the program could also teach him variations and perhaps entirely different types of game. By learning Wumpus he learns certain reasoning and planning methods. These are of various types which we summarize shortly. At a more general level, the student is learning how to approach new games and what methods are appropriate for unravelling the consequences of a given set of rules. This is not restricted to games. There are more general situations with logical properties and rules and he might be developing a skill in producing effective procedures for acting in these situations. When first in a new situation one must direct the most resources towards an understanding of the situation. As skill accumulates, fewer resources are needed and eventually tuning up and debugging is only done rarely. This is a general property of skill aquisition. (See Sussman ,1973, for a computer model of this kind of learning.)

The corresponding aim of an advisor is to help the student learn how to do all this. Our current Wumpus Advisor only advises on particular points of play so the student will only build up general skills indirectly. Later, we describe an approach that can be taken to improve the Wumpus advisor and consider decision making skills in more general terms showing how the advisor might teach these.

There appear to be several different styles of playing and thinking about Wumpus. People bring a variety of attitudes to the game. Some play very safely while others play with abandon for the fun of taking risks. Those who approach the game from the point of view of its logical structure are more likely to learn efficient play in a shorter time than those who neglect this structure. On the basis of informal observations, they appear to quickly absorb and benefit from the current program's style of advice. Players who see the game from other viewpoints might also benefit from our advisor's analytic approach which can be generalized widely to other domains. However, the current advisor does not give the gradual and sensitive advice about logical rules which must be provided for a student whose manner of play is different from its own. Again, on the basis of informal observations, we find that such subjects ignore long technical advice because it spoils the fun of the game. A more appropriate advisor would understand their motivations and treat the logical aspect as only one of several. This is an area which deserves considerable research.

1.2 Analytical and Synthetic approaches to learning games.

When a student is given the rules of Wumpus he must first analyse them to determine their implications. There are several ways he can do this. Firstly he can experiment, playing a variety of possibly risky moves until he empirically determines the regularities. In complex situations experimentation is combined with induction to generate and test hypotheses. A more direct method of analysis uses logic to infer properties of the game so that strategies can be developed to take advantage of these properties. This is very clearly illustrated in Wumpus. The player knows some but not all of the state of the board at any time. He can analyse the laws of the game and can develop about one dozen precise rules of inference that he can use to help locate the Wumpus and avoid dangers. He must embody these rules in a procedure for analysing a board situation and must use synthetic principles to do this. The Advisor contains an expert Wumpus player which has all of these rules already available to it. When relevant, it points out examples of the rules to help the player make his move. The player is made to consider the corresponding rule and incorporate it into his play.

Techniques of synthesis are used to construct programs and plans. Goldstein (Goldstein and Miller, 1976) describes a classification scheme for plans in the context of Logo program writing. Typical examples are linear plan, recursive plan and parallel plan. Acquiring skill at Wumpus can be seen as synthesizing a set of programs, so different synthesis techniques lead to different Wumpus playing strategies. Many problems are encountered when assembling separate pieces of advice into a coherent strategy. Some rules have preconditions and may only be invoked in certain situations. A strategy which only applies in certain circumstances will otherwise give rise to bad play. It is useful to explain errors in the student's model of play in terms of debugging and recognisable bug types. The student may then learn to recognise bug types himself and gradually build up a repertoire of repair techniques.

1.3 Methods appropriate to Wumpus

Besides general techniques of synthesis and analysis there are those which are associated with particular domains. Wumpus includes two types of knowledge omitted from previous teaching programs. These are incomplete and uncertain knowledge. A Wumpus player usually knows only a portion of the board and must develop procedures which can act effectively under these conditions. Three general methods; decision theory, probability theory, and planning are useful techniques for this type of situation.

1. Planning.

To play a game well one has to plan and should learn to avoid certain planning bugs such as planning too far ahead or too unevenly. There are often good reasons for choosing a few candidate moves and restricting lookahead only to these. AI has a considerable body of knowledge about planning in various domains and these principles should be taught by a good advisor.

2. Decision Theory.

Because Wumpus involves uncertainty and most moves have a combination of valuable and dangerous outcomes we can well apply the decision theory paradigm which is useful in many more general situations. This theory shows how to assign values and costs to properties of outcomes and gives a way of comparing these utilities when the outcomes occur with calculable probabilities. It incorporates a back-up algorithm that combines planning with evaluating particular states.

3. Probability.

In any uncertain situation probabilistic heuristics may be used to advantage. Estimating the probabilities of death at each move is crucial to good Wumpus play and our program uses qualitative probabilistic reasoning in its expert player and for giving advice.

1.4 The rules of Wumpus.

Wumpus is played by one player, a Wumpus hunter, in a world consisting of a number of caves connected by tunnels. The player moves around this warren trying to avoid dangers and with the goal of finding and shooting the Wumpus. Initially the hunter only knows the structure of the warren immediately around him. He knows the number of the cave he is in and of all caves directly connected to him by tunnels. Every time he makes a move, which must be into a neighboring cave, he is told the cave-numbers neighboring his new cave. The dangers of the warren are pits, bats and the Wumpus which, like the player, are initially located at random in the warren. Any move into a cave containing a pit or the Wumpus results in instant death. If the player moves into a bat cave he is carried away by the bats and dropped into a random cave which may of course contain danger. Bats are not fast enough to save the player from pits or the Wumpus if he inadvertently wanders into a cave containing both bats and one of these hazards. They do carry the player away before he gets a chance to see what the neighbors of the bat cave are though. There are clues which help in avoiding the hazards. The player hears squeaking if he is one cave away from a bat and he can feel a breeze if he is one away from a pit. He can also smell the stench of the Wumpus from up to two caves away but cannot tell the distance directly. None of this evidence tells the player the direction of a hazard. The hunter has a bow and five arrows which he can fire at any time into a neighboring cave. The arrow will ricochet at random through the warren for up to a distance of five caves and will kill the Wumpus if he is hit. It is possible that the arrow will by chance find its way back and kill the hunter. A typical warren will contain 20 caves 3 bats, 3 pits, the player and the Wumpus.

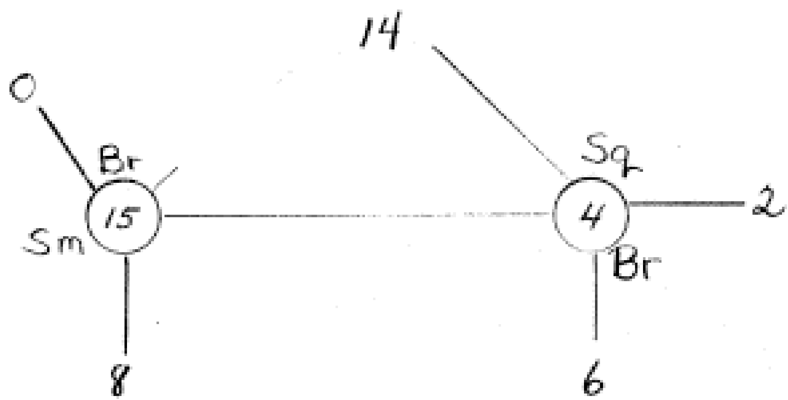
1.5 A Wumpus Scenario

The student's input is in bold type the computer's output is in italics. For the sake of readability we have taken the liberty of editing the advisor's output by, for example, using pronouns to refer to caves that have already been mentioned. We could easily update the program to effect most of these changes itself. In the case of pronoun reference, Winograd's program (Winograd 1973) provides methods that are sufficient for the limited examples in this scenario.

Hello Ira. I see you are a beginning player. You have had 5 wins and 3 losses. Let's begin.

You are at cave 15 with neighbors 4, 14, 0, and 8. Brrrr! I feel a draft. You are near a pit. What a stench. I smell the Wumpus! What now?

4



I wish you luck. You are at cave 4 with neighbors caves 14, 6, 2, and 15. Brrrr! Squeak! A bat is near. What now?

14

This is a dangerous move. You have been in two neighbors of 14 and you felt a draft in both. This is double evidence so it is probable that a pit is there. What now?

Advice

Cave 8 is a better cave to explore. What now?

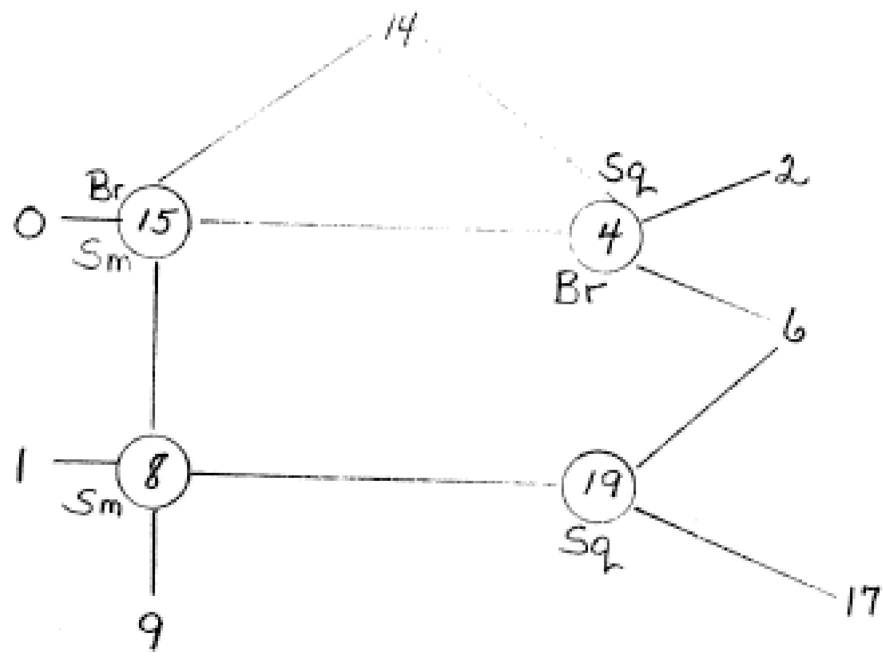
Why

There is no danger from the Wumpus at 14 or 8. It is possible that 8 contains a pit but it is more likely that there is one at 14. What now?

8

You are at cave 8 with neighbors 19, 1, 15 and 9. What a stench! What now?

19

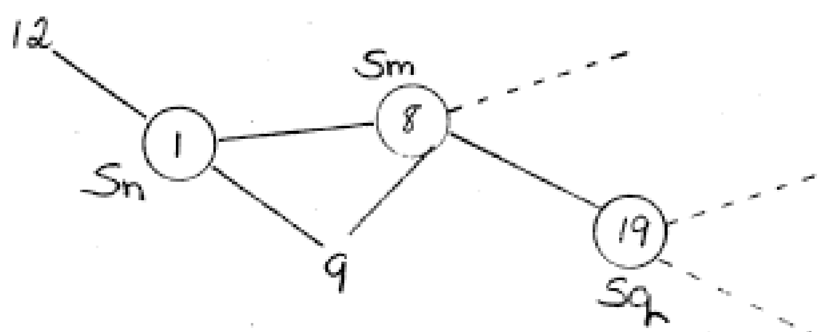


You are at cave 19 with neighbors 17, 8, and 6. Squeak! What now?

8

This is a good move, Ira, since it is on its way to the safe caves 1 and 9. You are at cave 8 with neighbors 19, 1, 15 and 9. What a stench! What now?

1



You are at cave 1 with neighbors 12, 8, and 9. What a stench! What now?

12

That is a risky move. The Wumpus could be in 12 because you haven't found any caves that are within two caves of 12 that were smell free. What now?

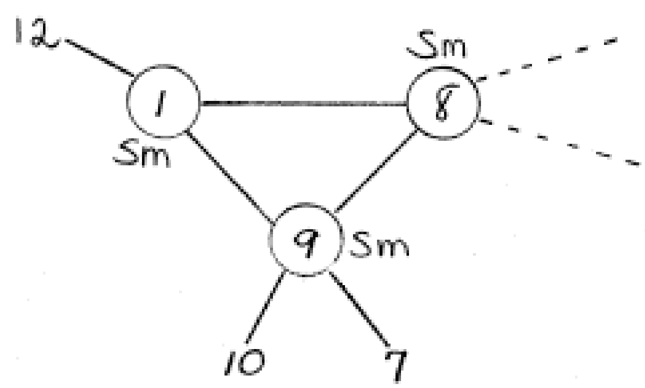
Advice

You should visit all the caves which you know are safe before you start taking risks. Cave 9 is a good cave to explore. Like 12 it is free from bats and pits; however, the Wumpus can not be at cave 9. What now?

More

If the Wumpus were at cave 9, then cave 8 would be next to him and you would smell him at cave 19. But, you didn't smell him at cave 19, so 9 is safe. What now?

9



You are at cave 9 with neighbors 8, 7, 10, and 1. What a stench! What now?

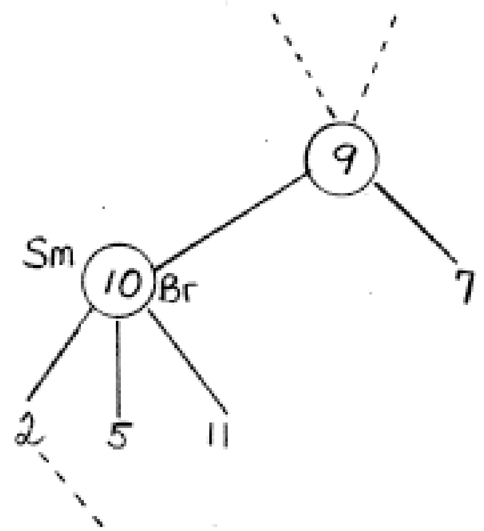
Help

I can give you information about where you are, where you have been, and what you have learned at each cave that you have visited. I can also give you advice. What now?

Advice

Cave 10 is a good cave to explore. It is possible that the Wumpus is in cave 10, but that is a necessary risk. There are no safer caves. What now?

10



You are at cave 10 with neighbors 11, 9, 5, and 2. Brrrr! What a stench! What now?

11

It is possible that cave 11 contains a pit. Would you like to go ahead with that move?

No

What now?

7

You can not go directly to cave 7 from cave 10. Would you like a route to cave 7?

Yes

Go to 9 and then to 7. Shall I make the moves?

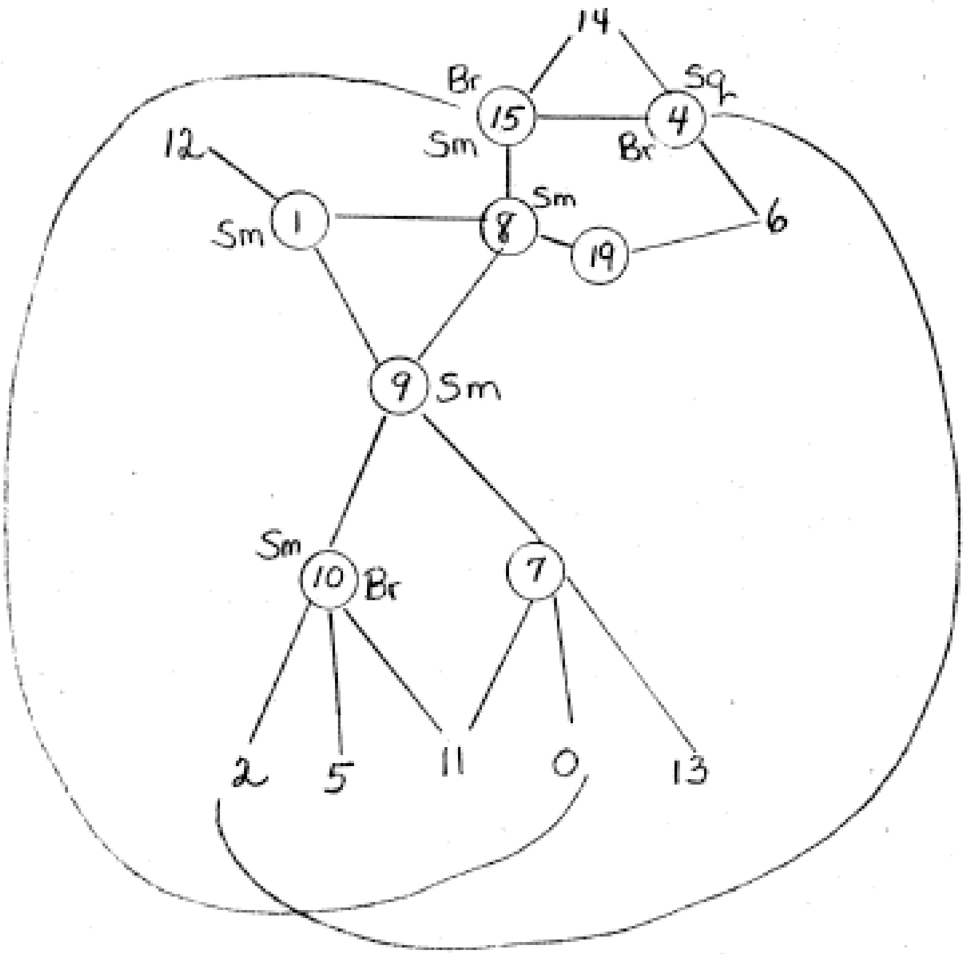
Yes

You are at cave 7 with neighbors 13, 0, 9, and 11. What now?

9

You are at cave 9 with neighbors 8, 7, 10, and 1. What a stench! What now?

1



You are at cave 1 with neighbors 12, 8, and 9. What a stench! What now?

12

That is a fatal move. Would you like advice?

Yes

The Wumpus is there by the following argument:

Cave 8 must be exactly two away from the Wumpus since it smells and cave 19 doesn't. This means that either cave 1, 15 or 9 must be next to him.

Cave 15 is two away since it is next to 4 which doesn't smell. Cave 9 isn't one away since you visited cave 7 and that didn't smell. Therefore, by elimination, cave 1 is one away. Hence, one of its neighbors must be the Wumpus.

The neighbors of 1 are 8, 9 and 12. You have visited 8 and 9 so, by elimination, the Wumpus is at 12.

Shoot

Which cave would you like to shoot into?

12

Congratulations you have shot the Wumpus!

2. The structure of the advisor.

2.1 Major capabilities

The Wumpus Advisor has several capabilities organised around an expert Wumpus player that embodies a considerable amount of knowledge about the game. This expert can evaluate the student's move, compare it against the best move and explain differences so that the student will improve his game. Future versions will include a model of the student as a perturbation of the expert. This will increase sensitivity to the particular problems facing each student of the game. In this section we outline the structure of the expert, its capabilities, its basic method of deduction and its advising and explaining strategies. Section 3 covers the details of each of these topics and section 4 outlines an improved approach developed by criticising our present effort.

Our expert Wumpus player has four major capabilities.

1. It deduces information about the state of the game from what it knows the player knows.
2. It can evaluate any move that the player can make.
3. It classifies all moves according to a set of categories designed to capture the major strategies of Wumpus playing.
4. Its evaluation of a move is modular.

At any time in a Wumpus game the player can see a small portion of the warren and can remember areas he has visited or has seen from a visited cave. He has partial knowledge of the warren from this information. He can use his memory of the location of bats he has come across and all the evidence from smells, breezes and squeaks that he has discovered in the course of the game. A good player should be able to deduce useful information about the position of various hazards by combining this information and using inference rules entailed by the rules of the game. The expert makes most of these deductions, only using information the student knows or ought to have remembered. In time, the advisor teaches the student to make all of these deductions himself in a reasonable manner and to use the information discovered to make a best play. There are two broad classes of information our expert can deduce. First, it can often determine exactly the positions of a bat, pit or the Wumpus, or can tell that a cave is definitely free of such hazards. This is clearly important to good play for hazards must be avoided and safe caves are worth investigating. Second, and very important in uncertain and incomplete situations where definite facts are unavailable, the expert can evaluate probabilities of hazards for any particular cave. Various heuristics are used for this and they represent qualitative knowledge about using evidence to make decisions.

Information gathered by these techniques is then used by the expert to evaluate each possible move. All moves are treated independently. There is no need to plan ahead in detail since a move can almost always be made at any time if at all. Only when a bat transfers a player to a remote part of the warren do caves become inaccessible. Even in this case the warren is so interconnected that it is unlikely to be much of a handicap. A move evaluation consists of a probability assignment for each hazard type and a simple measure of the information that would be gained by the move. So cave 3 may have a 0.3 probability of a pit, a certain bat and definitely no Wumpus. It may be near the Wumpus and so be likely to give information about it.

The expert has an executive which classifies all possible moves according to a seven point scale of goodness shown in figure I and discussed in detail in section 3.4. Each category is a distinct type. Safe moves are preferred to unsafe ones and given two moves of roughly equal safety, the one which reveals most information about the warren and the Wumpus is regarded as the best. All moves in the fringe area are considered. These are caves which are accessible but have not yet been visited. It is a waste of time to visit a cave that has already been visited unless it is on the way to another profitable cave in the fringe. If the player does visit such a cave it is assumed he is going somewhere valuable unless he wastes too much time by going in profitless circles.

The expert is composed of four main units, an executive and three specialists, one each for bats, pits and the Wumpus. Naturally, from the symmetry of the game, the bats and pits expert are very similar and use similar deduction rules. Each specialist deduces what it can about its associated hazard and reports to the executive. Modularity allows for a comprehensible expert which is a natural advantage for teaching purposes. The student's play can be evaluated separately for each speciality and also on their integration. We expect that this will make it easier to construct student models. It certainly allows the current advisor to advise about one particular module at a time.

<u>EXECUTIVE CLASSIFICATION</u>

TYPE NO.	IS THE CAVE SAFE? FROM BATS & PITS	IS THE CAVE SAFE? FROM THE WUMPUS	DOES THE MOVE GIVE INFORMATION? ON THE WARREN	DOES THE MOVE GIVE INFORMATION? ON THE WUMPUS
1	YES	YES	YES	YES
2	YES	YES	YES	NO
3	YES	NO	YES	YES
4	NO	YES	YES	YES
5	NO	YES	YES	NO
6	NO	NO	YES	YES
7	DEATH	DEATH	NONE	NONE

TYPE NO.	WUMPUS VALUE	BATS & PITS VALUE
1	1	
2	2	0
3	3	
4	1	
5	2	0 < VAL < 1
6	3	
7	-	1

Bat-pit safety has been given precedence. The bats/pits value of a cave is the probability of death by bats or pits in that cave. The Wumpus value is 1 if the cave is safe from the Wumpus but will give information about it, 2 if it is safe but will give no information, and 3 if it is unsafe.

figure 1.

2.2 Extra facilities

Several extra facilities have been added to the basic expert outlined above. They can be thought of as extra modules although they do not relate to the executive in the same clear way as the three hazard modules. All three of the facilities we next describe could be improved greatly and integrated into the advisor more cleanly.

We include a simple help *specialist* which will offer the student a good move when he is in trouble and will also present an explanation of it if the student desires. It is almost entirely a call to the expert for the current best move. We make no attempt to supply a move which is tailored to the students current difficulties. This enhancement will only be reasonable when student modelling is implemented.

Since the player may not remember all of the warren he has come across so far, we provide a route finder *specialist*. If he has any difficulty in reaching a goal suggested by the move suggester the advisor will offer a route through known safe caves. This is coupled with a help facility which gives the player information about any cave he has visited on request.

More important and most in need of further development is the *shooting specialist* whose job it is to prevent the player from wasting arrows and to advise him to shoot if he should be able to deduce the exact location of the Wumpus. It will dissuade the player from shooting if he has not located the Wumpus exactly or if he shoots into a cave that could not be the Wumpus, especially if there are other worthwhile things to be done. Future shooting specialists ought to weigh up the risks of shooting, the value of the arrow, the possibility of hitting the Wumpus and the availability of good plays elsewhere. We return to this when we consider a decision theory paradigm for future Wumpus advisors.

2.3 The advising paradigm

The advising paradigm for our current program is a simple one. This is because we do not yet have a component which effectively makes models of the student. Our system describes his immediate behaviour and not the reasoning that led him to this. As a consequence, the advisor will advise when the student makes any non-optimal move and will give him a description of his bad play which is usually too full. Nevertheless, there are some subtleties involved even using our simple techniques.

While discussing the expert we noted that the executive classifies the student's move according to a seven point set of categories (see figure 1). We associate a program called a move-type-analyst with each type in this category set. The job of such an analyst is to comment whenever the student makes a move of that particular type. Each analyst will check to see if the student made a move that was significantly worse than the best possible before it criticises him. The conditions for this vary according to the particular type and this is one reason for having separate analysts. In general the best moves are the ones with the lowest classification numbers and a drop of one makes a significant difference. This is not always the case. For example move-classification 4 (unsafe because of bats or pits but safe from the Wumpus while giving information about it) is not always significantly worse than class 3 (safe from bats and pits but in danger from the Wumpus) even though in general a drop of one class does make a significant difference.

The comments made to the student depend on move types as well as on the particular board state. Firstly, the analyst comments on the move type itself with some statement such as "that is a risky move". Of course if there is no safe move it will say "good luck" and leave the player to his fate but often more specific comment is needed. There are two types of bad feature a move may have, those that are avoidable and those that are not. The analyst only comments on the avoidable ones, a property which depends on the better moves available at the time. If the avoidable danger was a bat hazard the bats expert would be called in to give an explanation of the hazard. The implicit assumption is that the student did not see it. With a good student model we could distinguish between this and the case when the player noticed the hazard but failed to see any better move. The advisor focuses the player's attention and stimulates him into finding a better move by refering to the hazard as a reason for not making the move he tried. It is also possible that the student found other moves which were free from the criticism but noticed faults in these that he was mistaken about or that he gave too much weight to. A good modeller should allow us to adapt advice giving to cases like this.

Having criticised the player's move the analyst allows him to think for a while by asking him if he wishes to go ahead. The player can change his move and will then be offered a better one. On request from the player the analyst will compare its suggestion with the player's move. The explanation is comparative so no common features of the two moves need mentioning.

We have summarised that part of the advisor that currently fits nicely into a framework. Throughout the program are numerous patches that improve advice giving in ad hoc ways. Examples of such special cases are, advising about shooting, commenting on repeated mistakes and cautioning about time wasting by moving only into visited caves. We hope eventually to include these in our theory.

2.4 Sensitivity to the student

Although no student modelling is done by the current version of the system there are two comments to be made about the way the program deals with the issue. First, some adaptation to student performance levels is made even without active modelling. The student is asked to rate himself on a four point scale of Wumpus hunting ability. It would be fairly easy to have the program actively make such coarse judgements over a period of a few games. The rating influences the advisor behaviour in three ways

a) provision for initial advice,

b) pruning explanations,

c) pruning the expert's deductions.

If the player is a raw beginner there are certain features of the game he might not have realised. For example, bats are not as dangerous as pits since they usually land you in a safe cave. Immediate observations such as these are told perhaps once or twice to a beginner and are not mentioned again.

The program can generate detailed explanations by tracing through the deductions made by the expert in determining such facts as probabilities of bats. It is useful to prune this advice leaving only relevant facts. The two most general approaches involve techniques not yet included in our advisor. One involves natural language dialogue. If the student were able to ask the program for detailed explanations when he needed them, the advisor could explain in a top-down fashion, beginning with the main steps of the deductions and awaiting prompting for particular substeps. It is possible to allow some form of prompting without a natural language capability if for each lower level step the advisor asks the student whether he needs an explanation.

A second method requires a good student model to determine what the player already knows. We incorporate a coarse version of this procedure. The student is asked to describe his level of play as a number from 1 to 4. The difference between a very good player and a novice is enough to justify ommitting explanations of simple steps when advising the good player. Though this does not solve the problem of overwhelming a beginner with detail, it does improve the situation for a good player.

Finally, we assume that one who claims to be only a moderate player will not make any of the more sophisticated deductions or probability judgements that our expert can make. In this case we remove the relevant deduction rules from the expert to bring it more to the level of the player. This can be expressed as, "regardless of the student he must learn to walk before he runs". Because of the modularity of the rules we can make this adjustment easily. The same property should aid us in designing a realistic student modeller in the future. When carried through this leads to the notion of a "syllabus" which is an organisation of the teaching material that provides guidance for deciding in what order the material should be presented.

2.5 Deduction paradigm

Most moves in a game of Wumpus yield information which may be used as evidence for locating and evaluating dangers on the board. We describe the detailed deduction procedures used for doing this in section 3 but it is worthwhile to make some general observations about the deduction paradigm we used. We use four main headings for our description.

1) An assertional data base,

2) Antecedent theorems,

3) Special representation of disjunctions,

4) Mathematical functions for evaluating probabilities.

The assertional data base contains information representing the state of the warren when it is set up. It includes the connections between caves and the exact locations of the player and the hazards. Initially, the player knows nothing about the hazards so we distinguish properties and relations which describe his changing view of the world as the game progresses from the actual state of the world. The expert, of course, plays from the players point of view although it is conceivable that future programs with more sophisticated advising methods will "cheat" and help the player avoid difficulties he is unprepared to face. There are two types of properties and relations. One set of properties is a primary set including such properties as SMELL, VISITED, etc. It is assumed that any player will have these as part of his vocabulary since they are so closely tied to the way in which the rules of the game are presented to him. Other properties, such as 1-AWAY, 2-AWAY, are more remote. They appeared useful to us as we designed an expert. It is important to note that the student might not have these in his vocabulary until the advisor shows him that they are useful. Left to himself he could come up with a totally different representation for his play. We assume that there is only one good strategy and all the program's explanations are phrased in terms of the vocabulary needed for the inferences involved in this. The hope is to set the student thinking along the same lines. It is important for future work to remember that different people may represent problems differently so that a better advisor must be able to determine a student's representation and model him accordingly. In Wumpus type situations it may be important for the advisor to see how the student represents the warren diagramatically though, in general, multiple representations poses a very difficult question. To summarize, our program uses a single predesigned representation and attempts to impose this on the player.

Wumpus is a sufficiently simple game that antecedent methods can be used to keep track of new deductions. Whenever any new information appears the expert draws all implications it ever will between this and the old information. Thus we capture one aspect of a game player. He has a view of the game state which slowly changes as new information interacts with it. The expert has theorems which determine features of caves such as being one cave away from the wumpus, being safe, or containing the Wumpus. Some of these are simple, for example the condition that an arrow misses the Wumpus would trigger a theorem to assert that the cave the arrow was fired into is safe. Other theorems have several possible triggering conditions because a feature of a cave can depend upon features of all its neighbors. It also happens that a theorem may be triggered to prove a property already known to be true. In order to prevent unnecessary chain reactions of triggering an antecedent theorem always checks first to see if its result is true already.

These design features are common knowledge to AI programmers but take on a new light in an advice giving program. They are features which could improve a player's game if he organised his knowledge by them.

When the expert deals with bat and pit inferences it is interested in the probable locations of bats and pits. This requires it to represent disjunctions such as "there must be a bat in cave 1, 2 or 3". We were led to use a special representation in terms of *candidate sets*. In the example just given there would be a candidate set of (cave1 cave2 cave3). Bats and pits deduction procedures were designed around this notation and manipulated using intersection, size and set inclusion.

Evaluating the likelihood of a bat for any particular cave differs from the logical deduction process used to find the exact features of caves since it involves probability. It is extremely hard and messy to apply probability theory exactly to the Wumpus-situation. All probabilities are conditional on the partial information already accrued at the particular stage of the game. This leads to complex formulae at best and exhaustive combinatorial search at worst. Our expert is instead a model of heuristic and approximate probabilistic reasoning of the kind that knowledgable game players use in common sense judgements about the game. We determined four general methods that might well be used to estimate probabilities and adjustment the results to account for multiple evidence and the phenomenon of evidence being explained away. Our rules embody simplifying assumptions and are generally useful outside of Wumpus. Though we expect that most students will use some qualitative analogue of our rules, the advisor represents them as mathematical formulae embodied in procedures. This has a quantitative nature which makes verbal advice hard to give. The advisor overcomes this partially by pointing out the evidence it uses as data for its formulae and then saying that the student should deduce it is likely (probable, etc) that the cave in question contains a hazard. We don't yet know how much advice giving about common sense reasoning can be based on a quantitative model.

2.6 Generation of explanations

The Wumpus advisor gives detailed explanations of its reasoning. This leads the student to deduce useful properties of the board position and to use them when deciding on an appropriate move. Explanations are produced in a very simple way similar to that used in Stansfield (1975). An explanation bears an almost isomorphic relationship to the deduction procedure that is being explained. Each general rule of inference is associated with an explanation function. If the rule is of the form "A and B implies C", the explanation function prints out an explanation of the basic form "C because A and B". Since rules may be applied in many cases, many explanations can be produced by the same explanation function. This is only the simplest example of the method which is extended in two ways. First, A and B, the premises of the rule, may themselves be consequences of other facts and implied by other rules. The explanation function for "A and B implies C" calls the explanation functions for these rules and so on. Eventually a complete and detailed explanation of the inferencing is produced. Second, each explanation function is a procedure and can easily have idiosyncratic behaviour. One common addition is for a rule to state itself as well as the particular instance. So we could have "Caves you have visited are safe. You have visited cave 3 so it is safe". It would be possible by keeping a simple record to have the rule printed out with the instance for the first few times only. Other additions make the English output flow better and, occasionally, context sensitive aspects can be added. The program will usually refer to a visited cave as "cave x which has been visited" but because of context might say "cave x where you are now". Up to a point, these embellishments are easily added and the advisor has many. A general purpose English output program must be the next step (see Slocum 1975, McDonald (forthcoming), Riesbeck 1975).

Since the expert program is modular and contains an executive, the explanation functions fall neatly into classes. Some explain about bats and pits or about the Wumpus and some about the strategy as a whole.

It is easy to see from the example that the explanations become longwinded and detailed. To some extent their hierarchical nature eases this but it would be preferable for only the more relevant or important parts of the explanation to be given to the student so that he is not confused by too much information. We could have included various ad hoc techniques for pruning explanations which would have been moderately satisfactory. It seems more sensible from a research standpoint to first improve the student model so that there is a good basis for judgements of relevancy.

3. Program details

3.1 Bats and pits modules

The bats and pits modules of the expert embody about eight rules of inference and use them to determine the positions of bats and pits. They are of two kinds, logical rules which can be used to deduce the exact location of hazards, and probabilistic rules which can only estimate the likelihood of bats and pits in any particular cave. Both types of rule have already been discussed and here we describe them in detail.

There are four logical rules for bats.

a) A squeak heard in any cave implies that there is a bat in at least one neighbor of the cave.

b) Visiting a cave will tell you whether that cave contains a bat.

c) If a cave does not squeak then none of its neighbors can contain a bat.

d) If the total number of bats is given, they can sometimes be located exactly.

Rules for pits are almost identical, the one difference being that rule b) is of little use. If you fall in a pit the game is over whereas a bat may simply carry you to a safe cave elsewhere. Rule d) is fairly complex and is not implemented in our system. It works because if there are many more caves next to known squeak caves and only a few bats in the warren then only certain arrangements of bats will explain all the squeaks. The crucial point about rules a) b) and c) which a beginner may not immediately notice is that b) and c) may rule out possibilities suggested by a) to leave only one. In this case a bat or pit has been exactly located. Knowing the exact location of a bat in such a manner can in turn allow the probability rules to explain away certain squeaks neighboring that bat. This could lead the expert to conclude that certain caves are safe.

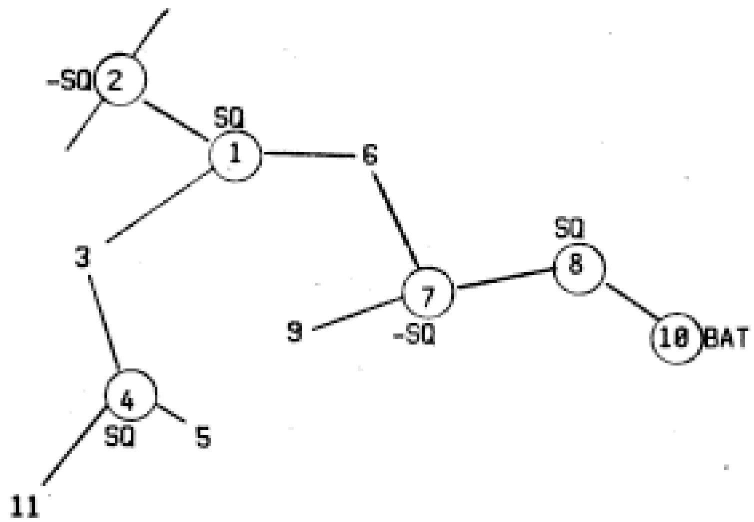


figure 2.

Consider the example in figure 2. Caves with circles around their numbers have been visited; caves 1, 8 and 4 are known to squeak; caves 2 and 7 are known not to squeak. Because of the squeak at cave 1, either cave 2, 3 or 6 must contain a bat by rule a). But 2 cannot by rule b) (it has been visited) and 6 cannot because of the lack of a squeak at cave 7 by rule c). This leaves only cave 3 as the bat cave. But a bat at 3 explains away the squeak at 4 so there is no reason to suspect a bat at 11 or 5.

To implement the rules we use candidate sets. Firstly, the state of the board as seen by the player is represented in the data-base using the properties KNOWN-SQUEAK, KNOWN-NOT-SQUEAK, VISITED, V-BAT and KNOWN-NEIGHBORS. V-BAT means the cave has been visited and contains a bat which therefore carried the player away before he saw the neighbors of the cave. Next a candidate set is generated for each squeak cave, duplicate sets being flushed. At least one bat must be in each candidate set. A unary candidate set is added to account for each visited bat cave. The sets produced to account for figure 2 would be

(2 3 6) (3 11 5) (7 10) (10)

Next, rules b) and c) are applied to remove caves from candidate sets. We now have the sets

(3) (3 11 5) (10)

Logically, in our example, we have deduced that caves 3 and 10 contain bats. If we knew that there were only two bats in the warren, the unimplemented rule d) could be used to prove that 11 and 5 are absolutely safe.

At this stage, the logical rules are exhausted and the probability rules take over. There are four probability rules, each corresponding to a fairly general rule for estimating likelihoods based on limited evidence. The rules are qualitative versions of the application of simple probability theory and Bayes' rule. We will describe each one saying a few words about its implementation. The rules are as follows.

a) Equal likelihood

b) Evidence can be explained away

c) Multiple evidence can increase probability

d) Multiple evidence can decrease some probabilities

Whenever exactly one of a set of equally likely outcomes must occur, simple probability says that the total probability must be I and an estimate can be made of the probability of each outcome. This rule applies approximately to any candidate set produced by the logical rules. If the set has N members then we may deduce that the probability of a bat being in any particular cave is I/N. We can compare the safety of alternative moves because caves in smaller candidate sete are more likely to contain bats. This rule is approximate for two reasons. Firstly, there may be two bats in any candidate set although for a large warren and few hazards this is unlikely to make the rule inaccurate. Secondly, knowledge about the remainder of the warren may influence the probability of a particular cave having a bat in subtle ways.

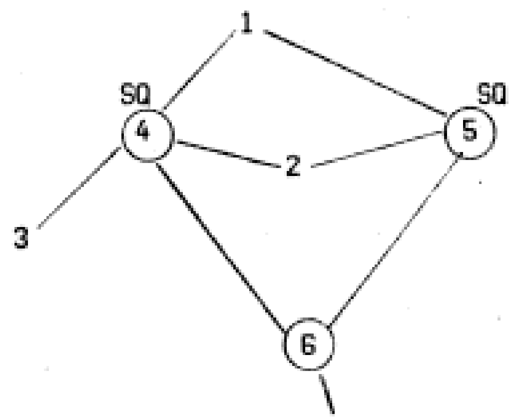


figure 3.

A particularly common way that this second case arises is that a probable or certain bat in one cave explains away evidence that supports a bats being in that cave as well as in several others. This rule can be applied whenever one candidate set is a subset of another. Figure 3 shows a case with two candidate sets (1 2 3) and (1 2). The bat in (1 2) due to the squeaking explains away the squeak at 4 that gave rise to (1 2 3) and there is no reason to believe a bat exists in 3. Evidence supporting 3 is explained away by the bat in (1 2). Our current advisor implements this by reducing the probability for 3 to the likelihood that a bat was put in 3 by the program which set up the board.

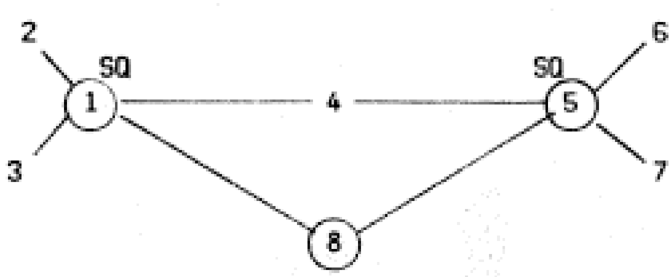


figure 4.

If two candidate sets overlap we have a situation of multiple evidence. Figure 4 shows a case where a squeak at 1 gave rise to a candidate set (2 3 4), and a squeak at 5 to a set (4 6 7). A bat at 4 would explain all this evidence. Alternatively, two pieces of evidence point to 4 but only one each to 2, 3, 6 and 7. We implement the rule for this situation by considering the probability of no bat at 4.

$$P(\text{bat at } 4) = 1 - P(\text{no bat at } 4)$$

$$= 1 - P(\text{bat in } (2\ 3)) * P(\text{bat in } (6\ 7))$$

A general version of the formula can easily be derived from this.

This rule introduces a problem. If the probability of the common case is increased then the total probability for each candidate set is raised above 1.0 which violates our initial approximation of one danger per cave. The greater probability of there being a bat in the common area should partially explain away the evidence and reduce the probabilities for the other cases. Since the exact formula for this would be cumbersome our program uses a rough formula to average out the discrepancy by reducing all the probabilities by a little. This is the fourth rule.

figure 5.

Another problem arises when more than one rule applies at once. Figure 5 shows two caves, 1 and 2, both squeak and are both neighbors of cave 3. If cave 1 is also next to a cave which is known to contain a bat then its squeak is totally explained away and gives no further information. It cannot be used in conjunction with cave 2 as a case of double evidence for a bat in the cave connecting 1 and 2. This means that we must apply the explain-away rule before the double-evidence rule. Such priority constraints occur often in programming so we should not be surprised when a student needs to know them as part of the his own program for playing a game well.

The four rules give estimates that fit the intuitive judgements generally made by players. The advisor states the factors used in the evaluation and gives a rounded off version of the result of its own formulae. It was unimportant for us that the student could precisely apply probability theory and we preferred that he be led towards making well-based estimates. The four rules we use are suitable for this and are applicable in other domains.

3.2 The Wumpus module

More complex deductions can be made about the location of the Wumpus than about bats and pits. Because a smell means that a Wumpus is within two caves rather than in a neighboring one it is weaker evidence than a squeak or breeze and gives rise to a much larger candidate set of possible Wumpus caves. On the other hand, absence of smell rules out more caves than would absence of the other types of evidence. Since smell-generated candidate sets have a radius of two caves it is possible that a neighbor of the smell cave is unvisited making the candidate set incomplete. It is also difficult to tell if moving from one smell cave to another takes you closer, further away or leaves you at the same distance from the Wumpus. All these factors lead to a more complex set of inference rules than we need for the bats modules.

There are two simplifications which make the problem tractable. Future programs might cover the more general case and it would also be interesting to vary the type of Wumpus evidence (intensity of the smell with distance from the Wumpus or number of Wumpi for example) to see what rules would then be needed. The two simplifications we have made are as follows.

1) The expert only makes logical deductions about the Wumpus and not probabilistic judgements.
2) In the original game the Wumpus may move when an arrow is fired which misses him. The Wumpus is fixed in our version.

We examine ways to make probabilistic judgements about the Wumpus later. If the second simplification is relaxed and the Wumpus is allowed to move, older evidence would be degraded but would not lose all its value. A smell cave which before a shot had implied that the Wumpus was within two caves, would now mean he must now be within three. A no-smell cave would now guarantee only that he is not in one of the cave's neighbors. The increase in variety of evidence would make the rules more complex.

We use five major Wumpus finding rules. Each is further away from the rules of play than the bats rules are and requires some simple proof of its correctness which naturally should play a part in the explanation of the rule given by the advisor. The rules are methods for deciding one of five properties of a cave namely , SAFE, TWO-AWAY, ONE-AWAY, WUMPUS, and MORE-THAN-ONE-AWAY.

Rule 1: GOAL - To prove a cave is SAFE

A cave is safe;

a) if it has been safely visited

b) if an arrow has been fired into the cave and no Wumpus was hit

c) if there is a NO-SMELL cave within two caves of it

This rule is easily justified and is invoked whenever one of the properties, VISITED, MISS, NO-SMELL is asserted about the cave in question.

Rule 2: GOAL - To prove a cave is MORE-THAN-ONE-AWAY

A cave is more-than-one-away from the Wumpus;

a) if we can prove it to be two-away

b) if it doesn't smell

c) if a neighboring cave does not smell

a) is obvious and b) and c) are simple since if a cave were the Wumpus or one away then all of its neighbors would smell. This rule is not exhaustive. There are probably other ways to prove more-than-one-awayness but its use is limited to these special cases as a help to later rules.

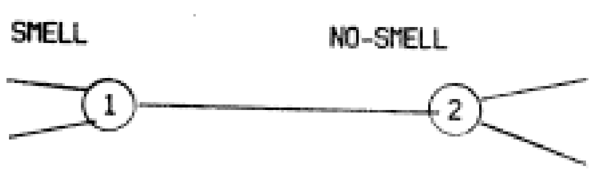
<u>Rule 3</u>: GOAL - To prove a cave is TWO-AWAY

A cave is two caves from the Wumpus;

a) if it smells and it is more-than-one-away

b) if it smells (so all the neighbors are known) and none of the neighbors is the Wumpus.

Both parts of this rule need comment. Rule a) depends on the configuration shown in figure 6.



Cave 1 must be exactly two from the Wumpus.

figure 6.

Since cave 1 smells it is within two caves of the Wumpus and must be either one or two caves away. But cave 2 must be more than two caves away and, as 1 and 2 are connected, the only consistent case is for cave 1 to be two away from the Wumpus. Both caves must be visited for this rule to be applied and the rule is triggered when any SMELL or NO-SMELL cave is discovered.

Case b) succeeds by proving that the cave is more-than-one-away from the Wumpus. Since it smells, it is either one or two away and so must be two away. Notice that rule 2 does not help here. Instead, we prove that no neighbor is the Wumpus cave so the cave in question is more-than-one-away. This rule is triggered when any cave is shown to be safe by rule 1. All neighbors of the new safe cave are checked for smells and any cave which does smell has the rule applied to it. Alternatively, a new smell cave may trigger the rule. If either case of rule 3 succeeds it will trigger rule 2.

<u>Rule 4</u>: GOAL - To prove a cave is ONE-AWAY.

A cave is one away from the Wumpus is it has a neighbor which is two away and all other neighbors of that cave are more-than-one-away.

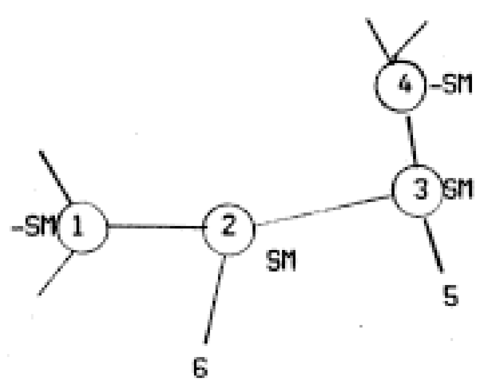


figure 7.

Figure 7 shows an example in which cave 6 must be one away from the Wumpus. The reasoning is as follows. By rule 3a), cave 2 is two away. But we know all its neighbors and one of them must be one away. Cave I cannot be, by rule 2b), and since cave 3 is two away, by rule 3a), cave 3 cannot be one away either by rule 2a). By process of elimination, this means that cave 6 must be one away.

Notice that rules 3b) and 4) are similar to the bats and pits logical rules. First a candidate set is generated in which at least one element has a desired property. Then all members are deleted and the remaining possibility becomes a certainty. This technique could be called "reasoning by elimination". In the bats case the property was directly related to the game rules whereas the Wumpus rules require some thought to discover relevant properties such as ONE-AWAY. It would be interesting to see if we could design an advisor that would lead a student to develop these Wumpus rules from the bats rules and to realise that reasoning by default is a commonly useful method worth identifying and naming. We leave it to the reader to see how the method generalises to give rules for detecting Wumpi who smell more and can be detected from greater distances. Rule 5 also uses reasoning by elimination.

<u>Rule 5</u>: GOAL - To prove a cave contains the Wumpus.

A cave must contain the Wumpus if it has a neighbor which is one away from the Wumpus and all others neighbors of that cave are safe.

We can see an example of rule 5 in figure 8, an extension of figure 7. Suppose the player visited cave 6 and discovered it smelled and connected with 3 and 10. Since 6 is one away and neither 2 nor 3 is the Wumpus by rule 5, cave 10 must be the Wumpus.

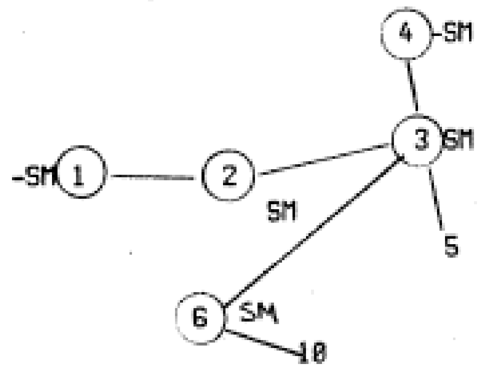


figure 8.

3.3 General comments on the Wumpus module

Despite the simplifications we made, the rules for Wumpus hunting are still complex. There are common elements and the rules inter-relate by triggering each other at several points. Nor are the rules complete. We could use the fact that there is only one Wumpus to help locate him. Figure 9 is an extension of figure 7 where we visit cave 6 and discover the new neighbors 7 and 8. The Wumpus must be one of these. But we have only one arrow left and daren't waste it. So we visit cave 5 and discover neighbors 8 and 9. We have two candidate sets for the one Wumpus, (7 8) and (8 9). He must be at 8.

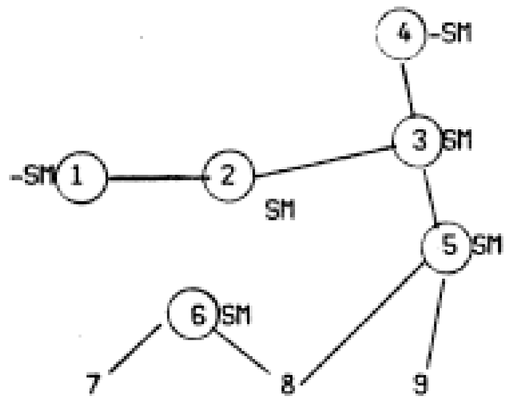


figure 9.

Such a large body of knowledge makes advice giving a difficult problem. Our advisor applies the rules, detects any instance in which the student could have made a better move and prints out a protocol of the rule's application. This naive tutorial technique could be improved in several ways. First, care needs to be taken over the distinction between a rule and its instances. Our advisor follows the paradigm of teaching by example. It should also teach by giving general explanations. Second, the rules interrelate and it is non-trivial to organise them all to simplify their application. It is possible that a player knows all the rules but is muddled about them in practice. Thirdly, we build no model of the student's knowledge so it is impossible to debug him when he uses an incorrect version of a rule. He may prove that a cave is two away by using rule 3a) but then think that all smell caves next to it must be closer to the Wumpus and must be one away. We need a way to classify, detect and correct these errors.

Just as our expert could make qualitative judgements about the probabilities of bats and pits, it is possible to introduce rules for judging the likely location of the Wumpus. There are two ways to do this. We can make use of the similarity between Wumpus hunting and bat finding where reasoning by elimination is used to set up candidate sets. All the probabilistic bat rules will then apply to the candidate sets. Rules 3a), 4 and 5 give rise to candidate sets for the properties TWO-AWAY, ONE-AWAY and WUMPUS respectively. There is a transitivity phenomenon too. Probability results from rules 3 and 4 can be used as evidence in rules 4 and 5 respectively. Here is possibly a general principle of plausible reasoning. An exact rule has a probabilistic counterpart for use when incomplete or uncertain evidence is fed into it. This would provide a nice basis for an advisor whose goal was to teach plausible reasoning by weighing evidence.

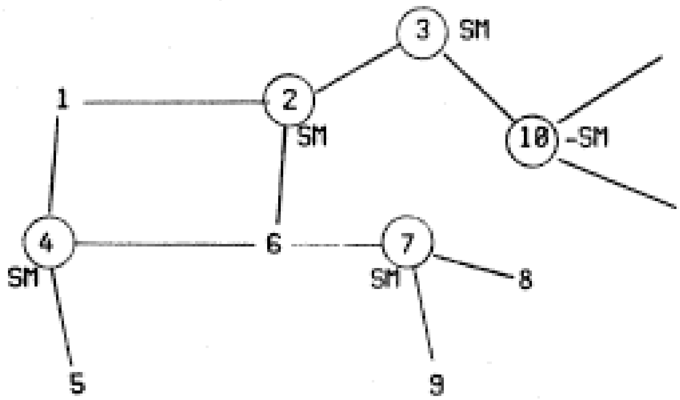


figure 10.

A second totally different strategy for making probability judgements is possible and gives rise to further principles of plausible reasoning of very general application. Given a board state such as that in figure 10, we can enumerate several hypotheses for the location of the Wumpus. Consider for example caves 1, 5 and 8. Each of these hypotheses will explain away some of the evidence in the figure. None of the hypotheses is totally discounted but each requires a different set of extra properties to be true of the board which are still to be tested. A Wumpus at 6 would explain all the smells and also the smell/no-smell pair at 3/10. It needs no extra things to be true of the board. Hypothesising cave 5 however, does not explain the smells at 2, 3 and 7. It thus needs extra board connections and these may or may not exist. Some measures of the evidence explained and the extra constraints imposed on future discoveries can be used to compare the likelihoods of various hypotheses. Both measures are needed. Constraint measures can be used to compare hypotheses and the explanation measure provides some absolute measure of confidence.

3.4 The executive's move classification

The bats, pits and Wumpus experts are used to determine the probabilities of meeting a hazard in any particular cave. This information must be used by the executive to evaluate a move. The executive forms the strategy component of a Wumpus player but since the game requires little lookahead, planning strategies are hardly needed. Each move can be evaluated on the basis of the current state and the available alternative moves. Two strategies exist and a players behaviour can follow either or both for several moves even though he makes all his decisions move by move. The strategies are called "playing safe" and "gaining information". Wasting time can be thought of as a third but is a degenerate case of the first and the advisor deals with it impatiently.

Playing safe means making the safest move you can find. Clearly the safety of a cave depends on the probability of it containing a hazard and this is reported on by the respective experts. Pits and the Wumpus mean certain death so they are easy to deal with. They are independent and their joint probabilities for any cave can be computed. A bat may be relatively safe since it does not necessarily leave the player in a deadly cave. The executive estimates the danger by using a simple formula which we will derive. If the number of caves is N, the number of bats b, and the number of pits p, then if we assume that no cave contains more than one hazard (a good approximation if N is much larger than p and b) we can reason as follows.

P(death by bat)

=P(you land on a pit)

+P(you land on the Wumpus)

+P(you land on a bat)*P(death by bat)

P(death by bat) = deadly caves/non bat caves = (p+1)/(N-b)

This works because after being dropped by a bat in a bat cave again the chances of death are the same as they were on first moving into a bat cave. Another way of thinking of this would be to sum an infinite series with a term for each total number of bats it is possible to land on in one move. A third way is to realise that the process of being moved about by bats must eventually stop in a non-bat cave and there is no reason to prefer one over any other so the chances are equally likely.

We explained the derivation of this formula in such detail because it is an opportunity to consider the amount of knowledge about the application of probability that a perfect advisor might need to explain. The moral is cautionary. In practice our executive simply evaluates the formula and states the likelihood of death as a part of its explanation of the danger in a cave. The student is expected to come to some similar decision qualitatively and to improve his reasoning to be coincident with the advisor's.

Shooting arrows is also a tricky type of move to evaluate. Our executive only deals with this in special cases when the Wumpus is either located or known to be in a different direction from the shot. A true estimate of the risk involved should include the probability of hitting the Wumpus since arrows can only be dangerous when they miss.

The second strategy for play is to gain information. Again, a move which has been made before gains nothing and the strategy degenerates into time-wasting. Information can be gathered in two main ways. Moving to a new cave gives information about the warren and perhaps also about bats and pits. However, new information about bats and pits can hardly be predicted. If a cave is suspected of being a bat or a pit, discovering that it is not could allow inferences to be drawn about the actual location of the hazard. In Wumpus, examinations in such detail are not very significant but it is easy to imagine real-world situations where a risk is worth taking for the negative information that may be obtained. A naive Wumpus player may rush into dangers for this reason and the advisor will caution him. Since Wumpi can be smelled from two caves away and as certain caves can be deduced to be two away it is possible and often safe to move into a cave that has a good chance of giving information about the Wumpus. Again, the true value of the information can only be gauged by considering the inferences it would allow. For our purposes we simply distinguish between "possible" information gain and "probable" gain.

The two strategies interact so that a decision theory model is needed to compare accurately the information gained with the risk involved. Since the version of Wumpus we use places no time constraints on the player, our advisor makes safe play more important than informative play. Before describing the mechanism for this, consider the following example of a case of complex evaluation. In the beginning of the game it may be useful to take a bat to reach new parts of the warren, especially if all other moves in the locality are dangerous. There are relatively few pits so it is unlikely that death will ensue. Later in the game the safety of a bat is unchanged. At this stage, most of the warren might have been investigated in which case the information value of taking a bat is lowered considerably. It may no longer be worth the risk. It is possible for the player to be completely trapped so he can only make deadly moves or repeat his old ones. In this case the value of taking a bat is that it might drop you in a new situation even if this had been visited earlier. A decision theory and planning theory of Wumpus could in future be the basis of an advisor for this level of play.

EXECUTIVE CLASSIFICATION

	SAFETY		INFORMATION	
TYPE NO.	BAT & PIT	WUMPUS	WARREN	WUMPUS
1	YES	YES	YES	YES
2	YES	YES	YES	NO
3	YES	NO	YES	YES
4	NO	YES	YES	YES
5	NO	YES	YES	NO
6	NO	NO	YES	YES
7	DEATH	DEATH	NONE	NONE

TYPE NO.	WUMPUS VALUE	BATS & PITS VALUE
1	1	
2	2	0
3	3	
4	1	
5	2	0 < VAL < 1
6	3	
7	-	1

Bat-pit safety has been given precedence.

figure 11.

Figure 11 shows the move classification scheme used by the current executive to capture the two strategies. Firing arrows and using bats to gain information have been excluded from the evaluation. Safety is factored into safety from bats and pits, and safety from the Wumpus. There are seven classes of move excluding repeat moves and they are numbered roughly in order of goodness. The seven can be divided into groups of three, three and one. The first three are totally safe from bats and pits as proved by the experts. Types 4, 5 and 6 are unsafe according to bats and pits and type 7 is certain death. The two groups of three are similarly organised according to Wumpus conditions. Best of all are moves known to be safe but next to smells and therefore likely to reveal information about the Wumpus. Second are those caves which are safe from the Wumpus but unlikely to give information about it. Finally, we have the caves which are unsafe from the Wumpus and therefore likely to give information about it. Each move type 4, 5 and 6 can be further ranked according to the actual degree of bat and pit unsafeness.

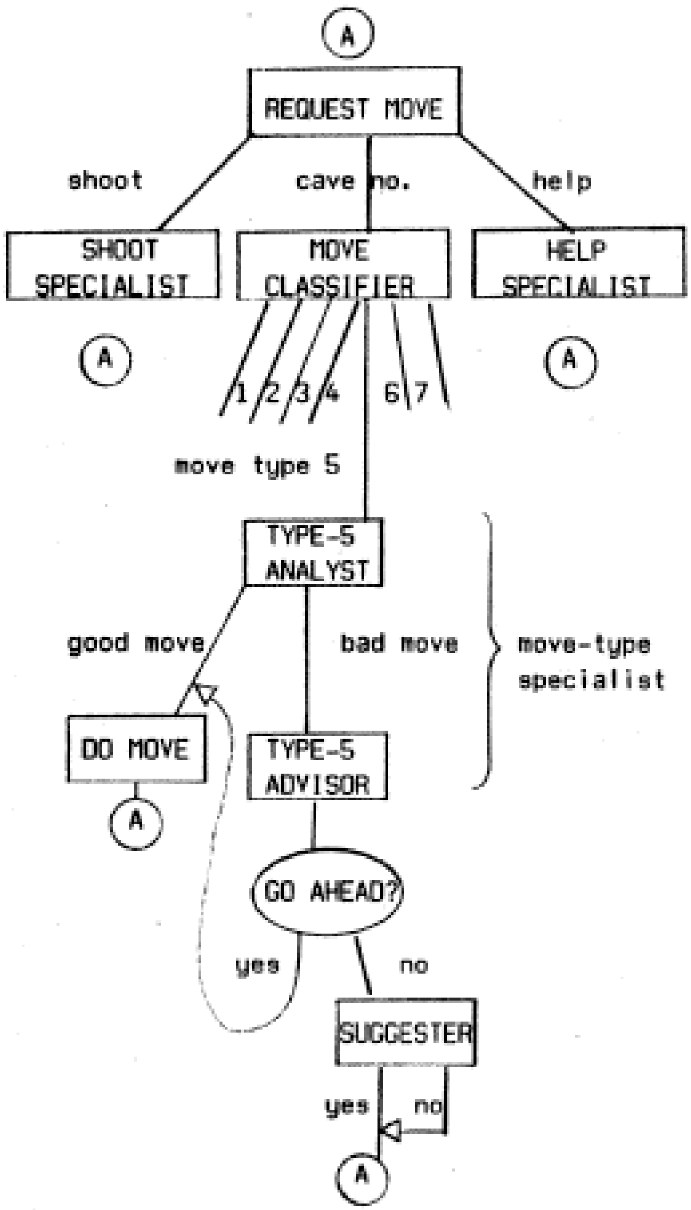
The classification is effective and to some extent distinguishes the strategies and places them in order of safety. It also clarifies the advice-giving role of the executive for as, we shall describe, each move type has a corresponding analyst which specialises in advising about moves of that type.

There are difficulties in capturing the interplay between strategies in a classification scheme. Consider move types 3 and 4. Both provide the same kind of information so their ranking can only be determined for particular moves by the relative dangers involved. Again, classes 4 and 6 give the same information and under certain conditions each could be better than the other. A better viewpoint is to consider decision-making under dangerous conditions to be a decision theory problem. The expert should be able to compare risks and profits and its explanations should be in these terms.

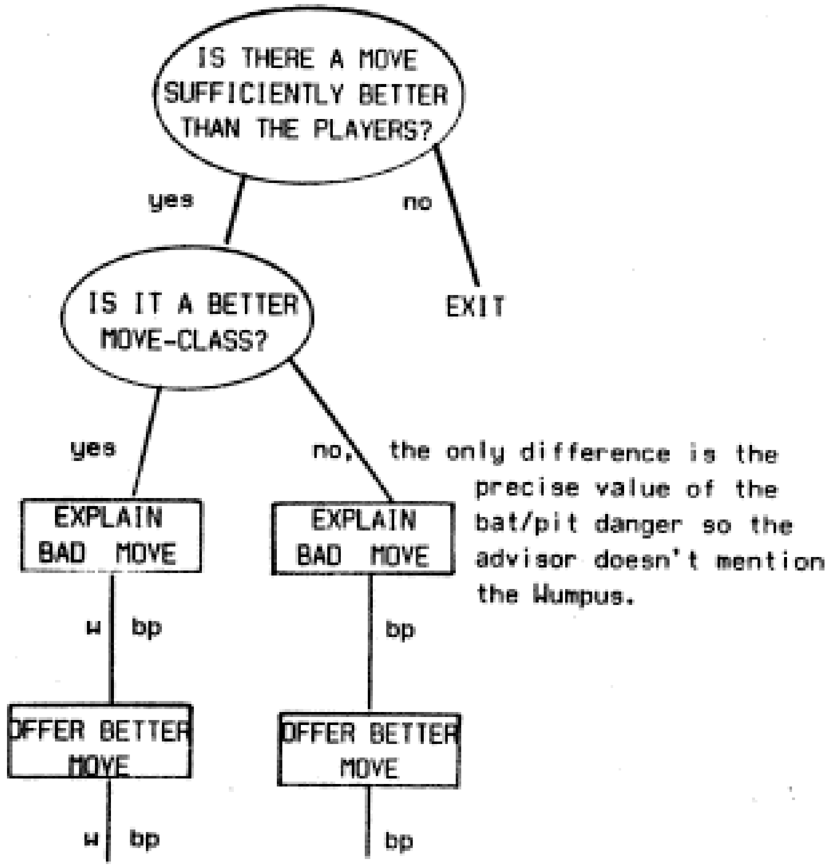
3.5 The flow of control

Figure 12 shows a simplified flowchart for the system. Whenever the program requests a move, control is at point A at the head of the flowchart. Certain special case such as shooting and requests for help are dealt with by special programs. Otherwise, the expert is called to classify all possible moves, in particular the one the player actually wanted to make, and control is switched to an appropriate analyst for the player's move type. Analysts consider the available moves to decide if the player made a good move. If he did it allows him to go ahead but otherwise it explains why the move was bad, partly using its own explanation functions and partly using those associated with the individual specialists for bats, pits and the Wumpus.

The player is always allowed the option of proceeding but if he wishes to change his move he is offered advice. When accepted, this takes the form of a good move and an explanation of the benefits of this move over the player's.



Flow of control
figure 12.



Move-type specialists
(example is type 5)
w means "explain about the Wumpus"
bp means "explain about bats and pits"

figure 13.

Figure 13 shows the schema for move-type analysts. It is self-explanatory except for a few points. If two moves are of the same type they may or may not be of sufficiently different quality to invoke advice-giving. Since move-types 4, 5 and 6 have a range of safety from 0 to 1, one move can be very safe while another of the same class is very risky. Second, the explanation functions are context sensitive. A move which is dangerous both because of the Wumpus and pits would not always give rise to an explanation of the Wumpus danger. If no available move was safe from the Wumpus the advisor gives the player the benefit of the doubt and assumes he has seen this. It assumes he chose the wrong move because he omitted to take proper account of the difference in pit safety. These assumptions are a recent addition to the advisor and we only discovered the need for them by using the program. It is remarkable how interaction with a program reveals glaring design ommisions which would otherwise be unnoticed.

Together the specialist modules for bats, pits and the Wumpus, the executive and the advice-giving components of each make up the majority of the advisor.

4. A decision theory approach.

The executive module of the wumpus expert represents various types of danger and the ways information can be gathered by means of a table. In effect, all the decisions about trade-offs between risks and gains are compiled. This method is restrictive and some subtleties of the trade-offs are omitted. We now describe a more uniform and general way of dealing with such decisions that will be suitable for an improved version of the advisor. It is based on decision theory which is especially designed to represent problems of choice in uncertain situations like Wumpus. The analysis of a problem using decision theory has three components.

1. A decision tree.

This is a tree of states of the world rather like a lookahead tree for game theory or planning. It is rooted at the initial state and at each state the player is given a set of alternate actions from which he may chose one. In Wumpus a state represents the position at a point in play and the choices facing the player are his legal moves. For any move the player makes, the world can respond in a variety of ways and each has an associated probability of occuring. If the player moves into a risky cave then two possible outcomes are that the cave actually contains the danger or that it does not. A more detailed description of the outcomes might specify the possible new neighbors that might be discovered. A decision tree thus has two types of arc, those corresponding to the players choices and those that correspond to the world's. The only difference from a game tree is the special way that the player's opponent behaves. In game theory he tries to make the best move whereas in decision theory he behaves according to probabilities that can be estimated.

2. An evaluation function for terminal nodes of the decision tree.

The terminal nodes of the decision tree have values for the decision-maker which can be evaluated if some procedure for doing so is specified. This procedure must take into account all of the good points of being at that state and weigh them against all of the bad points. It calculates trade-offs. The most common method is to measure each cost or gain with a single number and to combine these by simple linear weighting. The value of each feature is multiplied by a weighting factor and totalled with the others. If a feature is very good or very bad then it has a larger weighting factor either positively or negatively.

3. A back-up function.

Given a tree of possibilities and values for each of the terminal nodes it remains only to decide on the best action to take at the initial state. It is possible to work out what expected utility each action has by working backwards from the terminal values. Suppose we have a state which allows several actions each of which has several outcomes all of which are terminal. We know the probability of each outcome for a given action and we know their values since they are terminal. The expected utility for that action is easy to evaluate using simple probability theory. Which action should we choose? Clearly the one with the highest expected utility. This means that the expected utility for the state is the highest of the expected utilities of the actions available at that state. Now the state can be considered a terminal state since it has been valued and we can continue backing up the tree until we determine which action to take from our starting state.

This approach to the analysis of a decision problem assumes that the value of a state can be determined from the values of its component features. Four of these components clearly occur in Wumpus.

1. Risk of death

The utility of dying should be very large and negative. It cannot be minus infinity since this would multiply by any probability of death to be minus infinity. Instead, utilities could be a function of the probability of death. There are various ways that death can occur, falling into a pit, wandering into the Wumpus, shooting yourself with an arrow, or being carried away by a bat into a dangerous place. These possibilities reveal themselves in the decision tree. If a student fails to account for any of them it is reflected in his incomplete decision tree. The probabilities of several of these cases are quite tricky to deal with.

2. Information gain

The amount and value of information gained by any move are important. The value depends on what is already known as new facts may allow important inferences. Information may be gained about the warren itself and about the dangers in it. In variations of the game where the Wumpus may move it is possible to lose information. Inferences must be dealt with by a set of logical and probabilistic rules such as we have in the existing advisor.

3. Goal

The ultimate goal of the game is obviously an important consideration in deciding upon the value of a state. It is not sufficient to make safe moves or to find out information. It is also important to kill the Wumpus. Killing the Wumpus must thus have a high positive value. A small chance of killing it may be better than a large chance of gaining information. In variations of the game it would be possible to injure the Wumpus perhaps slowing him down if he can move around the warren.

4. Resources

A very important value in real-world situations is the value of resources. This was after all one of the main reasons for inventing money. The only resource used in our current version of the game is a supply of arrows. It is clearly very silly to take a chance with your last arrow though it may be worthwhile testing hypothetical Wumpus locations with the first few. Many other resource types could be added to the game, time constraints being one of the more general. Given a fixed time to play before the warren falls in on you will affect your play. It would become bad play to waste time. A more interesting way to introduce time is to make the Wumpus actively look for the player, eating him when it finds him. This could become a two player game with the advisor watching or else the advisor could be one of the players.

From the discussions of each of these components it is easily seen that Wumpus can have many interesting variations and all of the variations will easily fit into the framework of decision theory. A newer advisor based on such an approach would be able to advise a user about playing all the different variations. So far our goal for the advisor has been to introduce people to a situation in which the implications of a few logical rules are important for sensible decision making. In particular we chose a situation which had uncertain information. This naturally leads to the extension of teaching decision theory. When we consider this we discover at least six types of bug a student may have which directly concern decision theory some of which were out of the scope of our current advisor.

1. Failure to judge probabilities.

Failure to determine the likelihoods of the various outcomes of an action will cause errors when trying to back up the decision tree.

2. Inappropriate utility functions.

The student may have utility functions which are inappropriate for winning the game. He may think that pits are less dangerous than the Wumpus for example. Or he may be playing the game according to a strategy which requires a different set of utility functions. He may wish to fall into pits to help him remember the result of such an action or to check his hypothesis about what will happen. He might also be more interested in playing for fun than playing efficiently. An advisor that could recognise and relate to this would need to take account of the player's values accordingly.

3. Failure to see all the alternatives.

Expressed in the decision theory paradigm this bug corresponds to an incomplete procedure for generating a decision tree.

4. Refusal to cut losses.

This does not occur in Wumpus because there are no long term plans involved. It is however a common bug which manifests itself in a distorted set of values. Past losses are weighted too heavily and actions are taken which have only a small probability of annulling them.

5. Myopia.

A decision tree which is not deep enough will give rise to short-sightedness. Small immediate gains will be preferred to long-term ones. Large long-term losses will not even be considered.

6. Preoccupation with details.

This is related to the myopia bug but instead of the tree being too shallow it is one-sided. All the planning resources are used to plan ahead on only a few paths. The result is that when a move is eventually made it is either on the wrong track or based upon too shallow an investigation.

Wumpus has very simple strategies for play and though this was one reason for its choice it is perhaps time to consider what additional properties we would like a game to have for our advisor to teach in an interesting way. The simplicity of Wumpus largely arises because all decision making for a move can be done at the time of the move with only the information available at that time. Each move is made separately. Unlike chess, the player does not need to make up strategies which govern the style of his play for a sequence of moves. Nor are there ploys and trick methods which help lead an opponent into an error. In short the Wumpus expert needs to do no planning ahead more than one move. The basic cycle of play is to make inferences from current knowledge about the current state of the board, pinpoint the dangers, choose a move to avoid these dangers, make the move, thereby gain information and finally go to the beginning of the cycle.

A more advanced game would combine incomplete information with need for planning. Look-ahead would be necessary along sequences of actions each of which might have an uncertain outcome. There should be different methods of play that are applicable in different situations. Since evidence gathering is as important as evidence weighing, the game situation should allow the player to design a set of methods or strategies for gaining information. Action in an uncertain situation is a feedback loop. Evidence is gathered and weighed and plans are made both for acting and for gaining new information. The plans may be based on hypotheses, and information gathering should be designed to test these hypotheses as well as possible. One possible candidate for a game is the game "Clue". A murder has been committed and each player tries to play the part of a detective and discover three pieces of information, the weapon, the place, and the culprit. Each player has certain information and by combining everyones it would be clear what the answer was. A player may only get a limited amount of information from another at any one time. He thus has to make up strategies to determine the information he requests. Other players hear every player's request but do not know the implications of the answer fully. Players have to move around a board to particular locations before they can ask particular questions so an extra cost is involved and other players may be able to infer things from this behaviour.

Whatever game is chosen it will be necessary to combine planning with decision theory. Feldman (1975) has shown how this can be done. The principle is easy to describe. A decision tree is effectively a planning tree showing all the possible plans. The results of actions in these plans are uncertain but provision is made for each possible outcome. Instead of looking for the utility of a terminal state and moving so as to increase your expectation of this value, all the steps of the plan have to be taken into account. Each step has costs and gains associated with it and they must be added up to determine the value of the plan as a whole. Then the plans can be compared and the best one taken. An important feature of planning in an uncertain situation is that plans must be revised after each step is executed since new information may change the situation.

Summing up, it seems that decision theory provides a rich framework for improvements in the Wumpus advisor. In particular, the problems associated with making complex decisions involving conflicts of goal, limited resources, and uncertain information arise in a form which can be taught usefully by an advising program. These problems confront people often in everyday life when they interact with others and when they try to make plans for the future. Although an advising program written at this early stage will not teach them how to cope with more than a toy situation, it is a step towards a deeper understanding of teaching in this area.

References

Brown J.S. and Burton R.R., "*Multiple Representations of Knowledge for Tutorial Reasoning.*" in "*Representation and Understanding*" Eds. D.G. Bobrow and A. Collins, Academic Press, N.Y. (1975).

Burton, Richard R., and John Seely Brown, "*A Tutoring and Student Modelling Paradigm for Gaming Environments,*" in R. Colman and P. Lorton Jr. (eds.), *Computer Science and Education* (Advance Proceedings of the Association for Computing Machinery Special Interest Groups on Computer Science Education and Computer Uses in Education Joint Symposium, Anaheim, Cal.), SIGCSE Bulletin, Volume 8, Number 1 (SIGCUE Topics Volume 2), February 1976, pp. 236-246.

Collins A., Warnock E.H., Aiello N. and Miller M.L., "*Reasoning from Incomplete Knowledge.*" in "*Representation and Understanding*" Eds. D.G. Bobrow and A. Collins, Academic Press, N.Y. (1975).

Feldman J.A. and Sproull R.F., "*Decision Theory and Artificial Intelligence, II: The Hungry Monkey.*". To appear in Cognitive Science (1976).

Goldman N., "*Conceptual Generation*" in Schank R.C., "*Conceptual Information Processing*", North-Holland/American Elsevier (1975).

Goldstein I.P. and Miller M.L., "*Structured Planning and Debugging. A Linguistic Approach to Problem Solving.*" Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Working Paper 125, (1976).

McDonald D.M., "*Linguistic Reasoning in Language Generation.*" Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Technical Report (forthcoming).

Slocum J., "*Speech Generation from Semantic Nets.*" J.A.C.L. fiche no. 33, (1975).

Stansfield J.L., "*Programming a Dialogue Teaching Situation.*" Unpublished PhD Thesis, Department of Artificial Intelligence, University of Edinburgh, Scotland, (1975).

Sussman G.J., "*A Computational Model of Skill Acquisition.*" Massachusetts Imstitute of Technology, Artificial Intelligence Laboratory, Technical Report no. 297, (1973).

Winograd T., "*Understanding Natural Language.*" Academic Press, (1972).

Yob G., "*Hunt the Wumpus.*" Creative Computing, Sept-Oct 1975