MIDDLESEX UNIVERSITY COURSEWORK PART 1

2009/10

BIS3226

AI Techniques in Information Management

Roman V Belavkin

This assignment is worth 25% of the overall grade. The submission date is **Friday, December 11, 2009**. You should do the assignment **individually**.

Contents

1 Aims and Objectives

The aim of this part of the coursework is to develop a simple problem-solving or search agent using the ACT-R cognitive architecture and to test its operation. You will have to select a task or a problem (a list of example problems will be provided below), and then give the agent the required knowledge to solve the task. The knowledge will consist of several declarative facts (chunks) about the problem domain and several rules (procedures) that will enable the agent to search for a solution. Your work will be assessed based on a written report, which should include the following components:

1. The description of the task;
2. The description of the agent and its knowledge representation;
3. Evaluation of the agent's performance and the whole work.

2 Software Required

You will need a copy of the ACT-R cognitive architecture, which is free and can be downloaded from the following address:

`http://act-r.psy.cmu.edu/actr6/`

You can download the standalone version for Windows or Mac OS X. This version contains the ACT-R environment, documentation and tutorials. The tutorial contains several useful examples and exercises explaining the main principles of ACT-R programming. Most of the skills that you will need in this coursework are covered in Unit 1 of the tutorial.

ACT-R is based on the Common Lisp programming language, and ACT-R programs are called *models*, because they are used to model the way human thinks and solves problems. The models are standard text files which contain the code, and therefore to program them, you will need a text editor. For example, ACT-R environment has a built in text editor to edit the models. You can also use any other text editor, such as Notepad or Emacs. Unit 1 of the ACT-R tutorial contains `tutor-model.lisp` file, which you can use as a template for a new ACT-R model.

3 Problem Examples

Below are several examples of classical problems that have been used to study human problem-solving strategies. You can either use one of these problems in your coursework, or you can find and use a similar problem. You will see that all these problems have the following characteristics:

- Any moment in the problem can be described by a problem *state*, and usually there are many states.

- A transition from one state to another is achieved by an *action* of an agent.
- Each problem has one or more *goal states*, and the problem's initial state is not the same as the goal state.
- At each state, the agent has a choice of more than one actions (e.g. move forward or back).
- The choice of actions is limited so that it is usually not possible to jump directly from the initial state to the goal state (i.e. the agent has to perform a series of actions to achieve the goal).

3.1 Word Reversal

States

A horizontal strip is divided into several squares (e.g. six squares). Some of the squares are empty, while some have letters:

	A	T		R	

Actions

There are two kinds of actions:

Push : a letter can be moved into an empty position:

	A	T—	→	R		⟶		A		T	R	

Jump : a letter can jump over a single letter into an empty position:

	A—	T	→	R		⟶			T	A	R	

Initial and Goal states

The initial sate is a word that appears on the left. The goal is to have the same word reversed on the right.

A	R	T				⟶				T	R	A

Strategies

There is a choice of pushes and jumps in each state, and so there are many ways to achieve the goal state. The aims is to achieve it in as few moves as possible. The difficulty of the task increases if we increase the number of squares or the number of letters in the word.

3.2 The 8-puzzle

States

A square is divided into nine squares. One squares is empty, and others have numbers:

2	4	1
7	3	5
	6	8

Actions

A number can be pushed into an empty square horizontally or vertically:

2	4	1
7	3	5
←	6	8

$\longrightarrow$

2	4	1
7	3	5
6		8

or

2	4	1
7	3	5
↓	6	8

$\longrightarrow$

2	4	1
	3	5
7	6	8

Initial and Goal states

The goal is to assemble the numbers in order from any other initial state:

2	4	1
7	3	5
	6	8

$\longrightarrow$

1	2	3
8		4
7	6	5

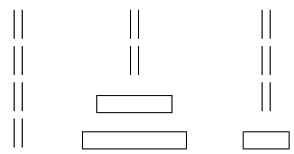
Strategies

There is always a choice of actions. If the empty square is in the middle, then there are four possible actions. If the empty square is in one of the four corners, then there are two possible actions. There are three possible actions if the empty square is in other places. The aims is to assemble the puzzle in as few moves as possible.

3.3 The Tower of Hanoi

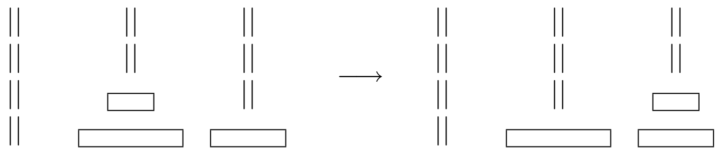
States

There are three rods and three disks that can slide on the rods. The disks are of different size. A larger disk cannot be put on top of a smaller one.



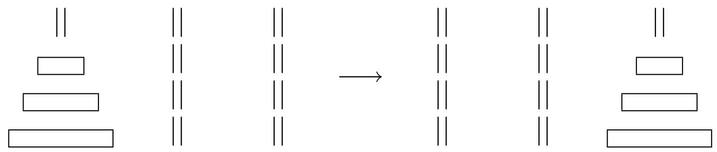
Actions

A disk can be moved from one rod to another, but only if it is empty or if it already contains a larger disk. A disk cannot be put on top of a smaller disk.



Initial and Goal states

At the beginning, all the disks are on the left rod. The goal is to move all the disks on to the right rod.



Strategies

In each state, there is a choice to move one of the top disks onto one of two other rods. However, some moves are not allowed if a rod contains a smaller disk. There are several possible strategies, and the aim is to move the tower in as few moves as possible.

3.4 The Cave World

This problem is a simplified Wumpus World problem.

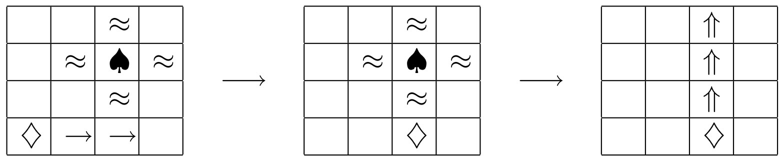
States

The world is represented by a 4×4 grid, where cells can have different properties. One of the cells contains a monster (wumpus), which does not move. One of the cells contains the agent (you), who does not know where the wumpus is. In the cells directly (not diagonally) adjacent to wumpus, the agent perceives a stench. The figure below shows the world with the agent $\diamondsuit$ in cell $(1, 1)$, the wumpus $\spadesuit$ in cell $(3, 3)$ and stench $\approx$ in the adjacent cells.

		$\approx$	
	$\approx$	$\spadesuit$	$\approx$
		$\approx$	
$\diamondsuit$			

Actions

The agent can move in four directions: North (up), South (down), West (left) and East (right). The agent also has one arrow, which can be fired in one of these four directions. The wumpus is killed if it was in the line of fire of the arrow.



Initial and Goal states

The agent always starts in cell $(1, 1)$. The wumpus can be at any cell. The agent loses if it does not kill the wumpus or if it walks into the cell with the wumpus. The agent wins if it kills the wumpus.

Strategies

At each state, the agent has a choice of moving or firing the arrow in one of the four directions. To find the wumpus, the agent has to explore the world and find at least two cells with a stench. This, however, is risky as after finding a cell with a stench, there is 1/3 chance to walk into the cell with the wumpus. So, the aim is to develop a strategy so that the agent wins at least two out of three times on average.

4 Assessment of the Components

4.1 Problem Description (30%)

Here you need to present the problem by describing its states, actions and goals. You can use diagrams showing states and transitions between states. You also need to analyse how the goal can be achieved. This can be supplemented by an example of a solution. You should also consider what makes one solution better than another, and how a strategy can be improved to use better solutions.

4.2 Description of the Agent and Knowledge Representation (30%)

Describe how you represent states, goals and actions in your ACT-R model. Describe which chunk-types you used for declarative knowledge, what chunks does the model has at the beginning. Describe some production rules of your model and outline which problem solving strategy they implement. Attach the complete code of your model into the Appendix of your report.

4.3 Evaluation of Agent's Performance (30%)

At the very least, you need to attach to the Appendix the trace showing the model's operation on the task. This can be done by running the model in the ACT-R environment and then copying the output trace from the Listener window. In addition, you can provide some quantitative analyses of your model's performance. For example, you can run the model several times and count the average number of times it achieves the goal, compute the average time or number of actions required to achieve it.

4.4 Presentation (10%)

Your report should be well presented. A good guide is the *Publication Manual* of the American Psychological Association (e.g. see `http://www.apastyle.org/`). At the very least, your report should be clear, typed or nicely hand-written document with good spelling, grammar and easy to understand English. There is no word limit, but a useful report should be just long enough to describe the work. A sensible limit is about 10 pages of typed text. Beyond this, you are probably being a bit too verbose. Tables, graphs, careful labelling and numbering are all well established and effective presentation tools.

Things to avoid are:

- Including images or diagrams that you did not create yourself or did not obtain the permission to use from the author (even if the image is from the Internet).
- Including graphs or diagrams that you do not describe in the text.
- Forgetting to label the axes on the charts.
- Using 3D charts to display 2D information.
- Including material irrelevant to the work.

Note also that you do not need to use coloured charts, as these can be quite expensive to print. A lot of of information can be displayed using black and white patterns or gradations of grey.

5 Assignment Submissions

Submit your report to the the Computing Science Student office, room TG18 by **Friday, December 11, 2009**, 16:00 hours. Do not include a disk or any other materials. Ensure that your work is clearly labelled with your name, student number, campus, course and the name of the module leader. Ensure that it is securely bound and easy to open. You should attach a coursework feedback form which will be dated and receipted. You should keep your receipt — it is for your own protection. Do not hand the coursework directly to your tutor.