Agents with Memories

1 Motivation for Adding a Memory

At this point, we put aside multi-agent systems and return to our consideration of environments in which there is only one agent. But, we now move away from reactive agents and begin to consider more intelligent agents. Most important will be to start to consider agents that 'plan ahead'. But, before we do that, we'll make a smaller change: we'll give our agent a memory.

The need for memory may seem obvious. The more complicated the task, the more likely it is that the agent needs memory in order to accomplish the task. But we'll try to argue this more precisely.

An initial question is: why do we need to remember things at all? The answer concerns the *accessibility* of the environment. Most environments are not wholly accessible to an agent's sensory devices. Sensor devices are mounted only at certain positions around the agent's body, and they have only a limited range.

This becomes a problem if the following can arise: there are two or more states of the world, requiring different actions from the agent, but the agent's limited sensory equipment cannot distinguish the states. Since it doesn't know which of the states it is in, it cannot reliably choose the right action.

Reactive agents, which choose their actions on the basis of their immediate sensory stimuli, are quite susceptible to this problem.

Question. *Remember the wall-following agent. We tried to design a production system for such an agent. I warned you that our agent would not be able to deal with environments in which there were 'tight dead-end corridors', i.e. a dead-end path between two walls that is less than two cells wide. And I asked you why reactive agents couldn't reliably wall-follow in such environments. Now is the time to think about this again. Find two states, requiring different actions of the agent, but which look identical to the agent.*

Equipping an agent with a memory is not sufficient to guarantee that the problem *never* arises, but it can *often* solve the problem. The agent might be able to choose the right action on the basis both of what it can sense and what it has remembered. (The information it has remembered may enable it to distinguish which of the otherwise indistinguishable states it is actually in.)

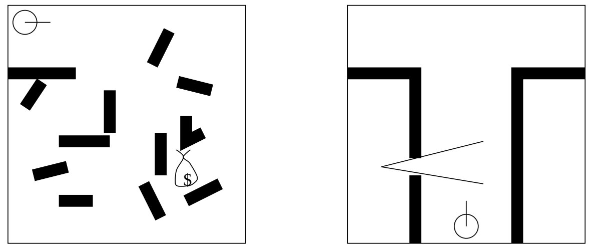
Before we look at memory proper, let's make one more observation. Agents can often avoid the need for (internal) memory: the world can act as their memory. When they need to 'store' something, they might be able to carry out an action in the world whose effects can later be sensed when they need to 'recall' what they 'stored'.

Question. *Here are two tasks that seem to require memory. But both can be solved by agents with no memory.*

In each case, how could you build a reactive agent (i.e. an agent with no memory) that could solve the task?

In the left-hand diagram, an agent must randomly traverse an object-strewn world until it find some treasure. Then, it must pick up the treasure and retrace its steps back to its starting point.

In the right-hand diagram, an agent must travel down a corridor. Halfway down the corridor, a light might (or might not) be flashed at the agent. Later, when the agent reaches the T-junction, it must turn left if, earlier, the light had been flashed; it must turn right if, earlier, the light had not been flashed.



Using the world as your 'memory' can be cumbersome. An agent might be reluctant to modify the world, just for the sake of 'storing' information; when the time comes for 'recall', the part of the environment where the information is 'stored' might not be immediately reachable by the agent's sensors; and, if there are other agents in the world, they can corrupt the information that an agent has 'stored' in the world. These problems do not arise (or arise to a lesser extent) with 'internal' memory.

2 Adding a Memory

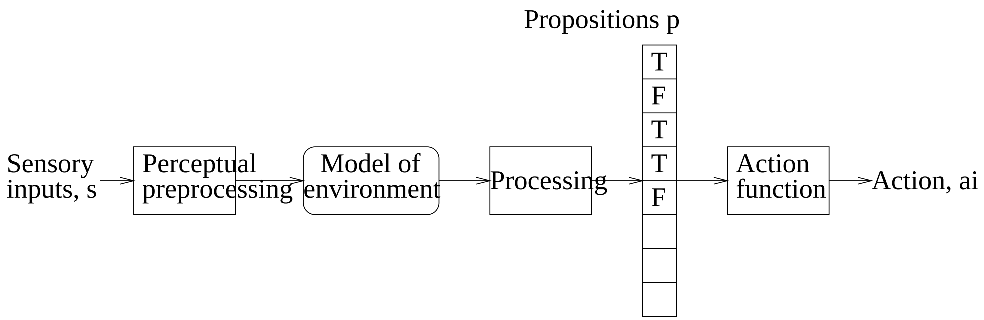
So we want to consider now agents that have a different action function from those that we have been considering. In our reactive agents, the action function maps

- from $\mathbf{s}$ (the immediate sensory inputs)
- to $\mathbf{a}$ (the actions the agent can perform)

Now, we're considering agents whose action function maps

- from $\mathbf{s}$ (the immediate sensory inputs) *and* $\mathbf{m}$ (what's in its memory)
- to $\mathbf{a}$ (the actions the agent can perform)

There are many ways of organising such an agent. The diagram shows one of these:



Sensory inputs, s, are used to update a model of the environment. The truth or falsity of propositions about the environment are then determined from the model. And then, in this particular example, actions are chosen, e.g. by a production system, on the basis of the propositions. (A similar architecture can be used when the agent function is implemented as a fuzzy controller or a neural network.)

The model of the environment is kept in the agent's memory. It is updated each time the sense/plan/act cycle is executed. New sensory inputs update the model. And the agent's actions will not only effect the world, they will also update the model. For example, if the agent decides to execute a move action, then it needs to both move in the world and update the model so that its new position in the world is reflected in the model.

3 Iconic Memory

In AI, when we're building models of the world, we choose between *iconic representations* and *logical representations*. There's no clear-cut distinction between the two, but it's a useful distinction to make because they have different strengths and weaknesses.

- In an *iconic representation*, there is a strong structural similarity between the representation and the world being modelled. Maps are good examples of iconic representations.
- In a *logical representation*, the world is described by statements in some language but the structure of the statements is not necessarily a reflection of the structure of the world. Modelling the world as a vector of **true** or **false** propositions is an example of a logical representation. Describing the world using first-order predicate calculus (later in this course) is another example.

One might say that an iconic representation is an *analogue* of the world; a logical representation is a *description* of the world.

Question. *What do you think the relative strengths and weaknesses of the two representations are?*

For the next good while, we'll be using iconic representations. Then, later in the course, we'll switch to using logical ones.

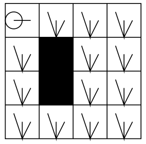
4 An Example

So, let's round off this lecture with an example of an agent who chooses actions based on an iconic model of the world.

The agent we will design is an 'intelligent' lawn-mower. The agent inhabits a grid-like lawn. It has eight touch sensors mounted around its body, so it can detect obstacles on the lawn. However, it has no way of sensing whether a part of the lawn has been mown already or not. But, by keeping track, in its memory, of where it has been, it can nevertheless know which parts of the lawn it has already mown.

An iconic model of this world can take the form of a two-dimensional array. Cells of the array represent locations in the environment. At any time, the cells in the model can contain one of five values: 'O' if the corresponding location in the world contains an obstacle; 'A' if the corresponding location in the world is where the agent is currently located; 'N' if the corresponding location in the world has not yet been mown; 'M' if the corresponding location in the world has been mown; and 'U' if the status of the corresponding location in the world is unknown.

Suppose the world is as shown in the left-hand diagram. And suppose the agent knows its initial location. But suppose the agent knows nothing else about the environment. Therefore, its initial model of the world is as shown in the right-hand diagram.



A	U	U	U
U	U	U	U
U	U	U	U
U	U	U	U

Remember, your God-like perspective lets you see the left-hand diagram; but the agent knows only what is shown in the right-hand diagram.

Our agent starts off, then, in a position of considerable ignorance. (In some cases, we, as designers of an agent, might choose to give the agent a less underspecified model of the world. We might even give it a fully-specified map of the world, showing the location of all objects. Obviously, the more complete the initial model, the better.)

The agent's action function will choose actions on the following basis. (Whether these are good lawn-mowing rules, I don't know.) It will turn clockwise until directly ahead of it there is a cell that it knows to be not yet mown ('N'), and it will move forward to that cell. If there is no such cell, it will turn clockwise until directly ahead of it there is a cell that has previously been mown ('M'), and it will move forward to that cell.

We'll look at the effects of a few sense/plan/act cycles. Remember that the model is updated twice in each such cycle. First, the latest sensory inputs are incorporated into the model. Later, the chosen action is not only executed in the real world, its effects are also reflected in the model.

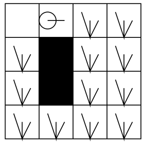
Sense:

A	N	U	U
N	O	U	U
U	U	U	U
U	U	U	U

Plan:

Move fwd

Act:



M	A	U	U
N	O	U	U
U	U	U	U
U	U	U	U

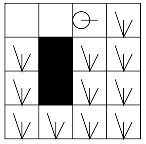
Sense:

M	A	N	U
N	O	N	U
U	U	U	U
U	U	U	U

Plan:

Move fwd

Act:



M	M	A	U
N	O	N	U
U	U	U	U
U	U	U	U

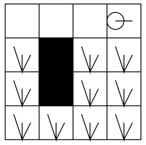
Sense:

M	M	A	N
N	O	N	N
U	U	U	U
U	U	U	U

Plan:

Move fwd

Act:



M	M	M	A
N	O	N	N
U	U	U	U
U	U	U	U

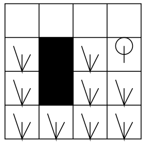
Sense:

M	M	M	A
N	O	N	N
U	U	U	U
U	U	U	U

Plan:

Turn 90R
Move fwd

Act:



M	M	M	M
N	O	N	A
U	U	U	U
U	U	U	U

Some cycles later...

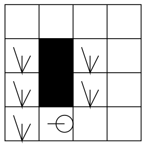
Sense:

M	M	M	M
N	O	N	M
U	O	N	M
U	N	A	M

Plan:

Move fwd

Act:



M	M	M	M
N	O	N	M
U	O	N	M
U	A	M	M

Later still...

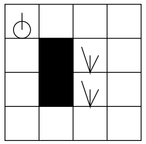
Sense:

M	M	M	M
A	O	N	M
M	O	N	M
M	M	M	M

Plan:

Move fwd

Act:



A	M	M	M
M	O	N	M
M	O	N	M
M	M	M	M

And even later...

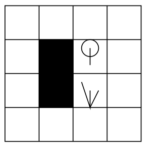
Sense:

M	M	A	M
M	O	N	M
M	O	N	M
M	M	M	M

Plan:

Turn 90R
Move fwd

Act:



M	M	M	M
M	O	A	M
M	O	N	M
M	M	M	M

This agent eventually build such a complete model that it has no need to sense the world at all. Obviously, this only happens in certain situations (e.g. a finite world that changes only as a result of the agent's actions).

You might like to consider what happens in worlds in which there are multiple agents and so our agent is not the only agent who is changing the world. What this means is that memories no longer provide information that has perfect certainty. (E.g. a cell that was obstacle-free earlier may not be obstacle-free now. Or, e.g., if there's an agent acting in the role of 'nature', then more grass will grow in cells that were mown earlier). This brings the complication of keeping the model up-to-date. This can be done either by revisiting and re-sensing parts of the world, or it might be done through knowledge and reasoning: you might be able to decide what you *expect* parts of the world to be like on the basis of knowledge that you possess (e.g. that grass re-grows), and so you might update your model using these expectations.

Question. *Consider an obstacle-free rectangular lawn, with the agent starting in the top-left. The agent, using the action function that I have given, will mow the whole lawn. Explain why.*

Question. *There are environments that the agent would never successively mow in their entirety. Can you come up with any?*