Situation Calculus

CS 475

September 8, 2003

1 Introduction

Situation calculus is a formalism for reasoning about action and change (RAC). It was first introduced in 1969 by McCarthy and Hayes in [?].

A *fluent* is a property of the world that changes over time. The world (at any particular time), called a *situations*, can be described by a set of fluents. To say that fluent F holds in the situation S, we write

$$Holds(F, S) \tag{1}$$

We will use a special constant S_0 to denote the initial situation.

For example, in the wumpus world, we can represent the fact that the agent (we call him $Agent$) is at the square $[1, 1]$, we can write

$$Holds(At(Agent, [1, 1]), S_0) \tag{2}$$

Fluents only represent the properties of situations. *Actions* make the world changes from one situation to another situation. Actions have their *effects* such as the action of *forward* of the agent makes him being at the square $[1, 2]$ etc. We write $Do(a, s)$ to denote the situation resulting from the execution of action a in situation s. With this notation, we can represent the fact that moving forward ($a = Forward$) in a situation s will result in the agent being at the square $[1, 2]$ as follows:

$$\begin{array}{l} \forall s[\\ \quad Holds(At(Agent, [1, 1]), s) \\ \quad \rightarrow Holds(At(Agent, [1, 2]), Do(Forward, s)) \\ ] \end{array} \tag{3}$$

Similarly, we can represent the effect of the action $Turn\_right$ is that the agent will face $East$ if he faces $North$ before the action is executed as follows.

$$\begin{array}{l} \forall s\ [\\ \quad Holds(Facing(Agent, North), s) \\ \quad \rightarrow Holds(Facing(Agent, East), Do(Turn\_right, s)) \\ ] \end{array} \tag{4}$$

2 Representing the Wumpus World

In the wumpus world, each situation can be characterized by:

- The position of the agent
- The location of the gold, the pit, and the wumpus
- The status of the agent (alive or dead)
- The stench of the wumpus that the agent can perceive (smell)
- The breeze of the pit that the agent can perceive (feel)
- The glitter of gold that the agent can perceive (see)

In all, we have the following constants

- $Agent$ – our agent
- $Gold$, $Pit1$, $Pit2$, $Pit3$, and $Wumpus$ – that represents the gold, the pits, and the wumpus, respectively, and
- the locations $[1,1],[1,2],\ldots,[4,4]$
- $Stench, Breeze, Glitter$

we can use the following fluents (predicates) to represent a situation

- $At(x,y)$ – "object x is at the location y", where x is an object ($Agent$, $Gold$, $Pit1$, $Pit2$, $Pit3$, or $Wumpus$), and y is a location
- $Alive(Agent)$ – "$Agent$ is alive" ($\neg Alive(Agent)$ – $Agent$ is dead) – Whoelse could be dead?
- $Has(z,y)$ – "location y has the property z", where $z \in \{Stench, Breeze, Glitter\}$ and y is a location

Some formulas about the properties of the world

1. *In the square containing the wumpus and in the directly (not diagonally) adjacent squares the agent will perceive a stench* – this can be represented by:

$$\forall y, y' \forall s[\quad Holds(At(Wumpus,y),s) \wedge Adjacent(y,y') \rightarrow Holds(Has(y',Stench),s) \quad] \tag{5}$$

   where $Adjacent(y,y')$ is a predicate describing the relation between two squares y,y' [What should we add? Should we write $Holds(Adjacent(y,y'),s)$ or just $Adjacent(y,y')$]

2. *In the squares directly adjacent to a pit the agent will perceive a breeze*

$$\begin{aligned}\forall y, y' \forall x \forall s[\\ & (x = Pit1 \vee x = Pit2 \vee x = Pit3) \wedge \\ & At(x, y) \wedge Adjacent(y, y') \\ & \rightarrow Holds(Has(y', Breeze), s) \\ ]\end{aligned} \tag{6}$$

If we have a predicate called $Pit(x)$ or and a set of constants $Pit1, Pit2, Pit3$, we can write the above equation by $\forall y, y' \forall x \forall s[Pit(x) \wedge At(x, y) \wedge Adjacent(y, y') \rightarrow Holds(Has(y', Breeze), s)$

3. Complete the formulization of the environment (page 154 ..)

So far, we have only represented the properties of the environment (the relationship between fluents). What is missing here is the representation of actions and their effects. Now, let see which actions are necessary for us. From the description of the environment, we see the following:

1. $Forward$ – the agent moves forward
2. $Turn\_right$ – the agent turns right
3. $Turn\_left$ – the agent turns left
4. $Shoot(y, y')$ – the agent is in the square y and he shoots to the square y'
5. ???

Representing the effects of actions:

$$\begin{gathered} Holds(At(Agent, [n, m+1]), Do(a, s)) \leftrightarrow \\ a = Forward \wedge Holds(At(Agent, [n, m]), s) \wedge Holds(Facing(North), s) \\ \vee \\ Holds(At(Agent, [n, m+1]), s) \wedge a \neq Forward \end{gathered} \tag{7}$$

But the agent can face 'South', 'East', and 'West' also ...

$$\begin{gathered} Holds(At(Agent, [n, m-1]), Do(a, s)) \leftrightarrow \\ a = Forward \wedge Holds(At(Agent, [n, m]), s) \wedge Holds(Facing(South), s) \\ \vee \\ Holds(At(Agent, [n, m-1]), s) \wedge a \neq Forward \end{gathered} \tag{8}$$

$$\begin{gathered} Holds(At(Agent, [n+1, m]), Do(a, s)) \leftrightarrow \\ a = Forward \wedge Holds(At(Agent, [n, m]), s) \wedge Holds(Facing(East), s) \\ \vee \\ Holds(At(Agent, [n+1, m]), s) \wedge a \neq Forward \end{gathered} \tag{9}$$

$$
\begin{array}{c}
Holds(At(Agent, [n-1, m]), Do(a, s)) \leftrightarrow \\
a = Forward \land Holds(At(Agent, [n, m]), s) \land Holds(Facing(West), s) \\
\lor \\
Holds(At(Agent, [n-1, m]), s) \land a \neq Forward
\end{array}
\tag{10}
$$

Find out what is missing in the above equations? Can we rewrite (7)-(10) as one single axioms? For now, let continue with other actions:

$$
\begin{array}{c}
Holds(Facing(North), Do(a, s)) \leftrightarrow \\
(a = Turn\_right \land Holds(Facing(West), s)) \\
\lor \\
(a = Turn\_left \land Holds(Facing(East), s)) \\
\lor \\
(Holds(Facing(North), s) \land a \neq Turn\_right \land a \neq Turn\_left)
\end{array}
\tag{11}
$$

Write the axioms for other directions!!!

$$
\begin{array}{c}
Holds(Alive(Wumpus), Do(a, s)) \leftrightarrow \\
Holds(Alive(Wumpus), s) \land \\
\neg(a = Shoot(y, y') \land Holds(At(Agent, y), s)) \land Holds(At(Wumpus, y'), s)
\end{array}
\tag{12}
$$

Question: What do I miss in (12)?

Continue with other actions, effects, ...

What holds at the beginning?

$$
\begin{array}{c}
Holds(Alive(Agent), S_0) \\
Holds(Alive(Wumpus), S_0) \\
Holds(At(Agent, [1, 1]), S_0) \\
Holds(At(Wumpus, [1, 3]), S_0) \\
Holds(At(Pit1, [3, 1]), S_0) \\
Holds(At(Pit2, [3, 3]), S_0) \\
Holds(At(Pit3, [4, 4]), S_0) \\
Holds(At(Gold, [2, 3]), S_0) \\
Holds(Facing(North), S_0)
\end{array}
\tag{13}
$$

3 Situation Calculus

The theory in the previous section is a situation calculus theory. In short, a situation calculus theory consists of

- A set of fluents
- A set of actions
- Situations, with S_0 is a constant denoting the initial situation
- Function Do that maps a pair of an action and a situation into a situation; $Do(a, s)$ denotes the successor situation to s resulting from executing action a
- Predicate $Holds$ whose first parameter is a fluent and second parameter is a situation
- A set of axioms about the initial situation (what is true/false in the initial situation? e.g. $Holds(At(Agent, [1, 1]), S_0)$ etc.)
- A set of axioms that describes the effects of actions
- A set of axioms that describes the precondition of actions; for each action a, the theory consists of one formula of the form $Poss(a, s) \leftrightarrow Holds(\varphi, s)$ where φ is a fluent formula.

Given a situation calculus theory (which is essential a set of first order axioms) – under certain conditions we can prove (or predict) what will be true/false after executing a sequence of actions in a situation. This is called the *projection problem*. We next discuss the assumptions needed in solving the projection problem. But first, for simplicity, let us leave the complicated 'Wumpus' world and describe the 'Block' world which will be used as the example for the discussion subsequently.

3.1 The Block World

Let us consider the block world in the following picture.

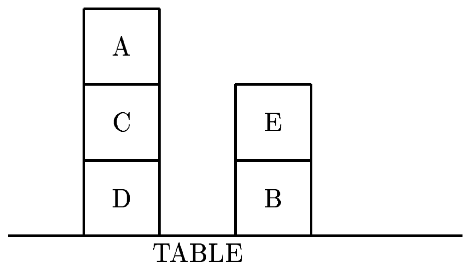

Figure 1: A Block World

We assume that the world consists of the above blocks and an agent who can pick up block and put it down on the table or on top of other blocks which are clear.

Objects of the block world are the blocks, the table, and the agent. The blocks are described by their names, $a, b, c, \ldots$. The table will be denoted by t. Since there is only one agent and no actions of the environment (for example, the agent will not die in this world - there is no 'Wumpus' or 'Pit'!), we do not need to have a constant to denote the agent.

We will use the predicate $Block(x)$ to denote that object x is a block.

The actions of the agents are:

1. $Pickup(x)$ - pickup block x,
2. $Put(x, y)$ - put block x on top of block y, and
3. $Put(x, T)$ - put block x on the table.

The fluents of the block world are:

1. $On(x, y)$ - block x is on the block y,
2. $On(x, T)$ - block x is on the table,
3. $Holding(x)$ - (the agent) is holding block x, and
4. $Clear(x)$ - block x is clear.

We note that

$$Clear(x) \leftrightarrow \neg \forall y (Block(y) \wedge \neg On(y, x))$$

which means that some properties of the world can be automatically deduced when we know about other properties.

The initial situation can be described by the following axioms

$$Holds(On(A, C), S_0) \tag{14}$$
$$Holds(On(C, D), S_0) \tag{15}$$
$$Holds(On(D, T), S_0) \tag{16}$$
$$Holds(On(E, B), S_0) \tag{17}$$
$$Holds(On(B, T), S_0) \tag{18}$$

Now, let us describe the set of precondition axioms and the set of effect axioms.

Precondition axioms: Before an action is executed, it requires that some conditions are true. This condition is called the action's *precondition*. For example, to put down a block x on top of block y, the agent must hold x and y must be clear. Another example is that to pick up a block x, the agent must not hold another block and x must be clear etc... We represent these by

$$Poss(Pickup(x), s) \rightarrow \forall y(Holds(\neg Holding(y), s)) \wedge Holds(Clear(x), s) \tag{19}$$
$$Poss(Put(x, y), s) \rightarrow Holds(Holding(x), s) \wedge Holds(Clear(y), s) \tag{20}$$
$$Poss(Put(x, T), s) \rightarrow Holds(Holding(x), s) \tag{21}$$

Problem The above formulas do not allow us to prove when an action, say $Pickup(x)$, is possible. I.e., when $Poss(Pickup(x), s)$ is true. Remember how to prove whether a literal is true?

Can we reverse the axioms (change $\rightarrow$ with $\leftarrow$) and then use them to prove when $Poss(Pickup(x), s)$ is true? No, this will not work since it is not correct! There might be conditions which we have not considered? For example, the block can be too heavy, the agent might be tired etc... So, we need to specify all of these conditions, called *qualifications*. But there are infinitely many conditions like this?

The above is called the *qualification problem*. It calls for the need for *nonmonotonic reasoning* in AI.

To overcome the qualification problem, we make the assumption that the antecedent of the above formula represents the sufficient and necessary conditions for the action to be executed. The *final* action precondition axioms for the above three actions are:

$$Poss(Pickup(x), s) \leftrightarrow \forall y(Holds(\neg Holding(y), s)) \wedge Holds(Clear(x), s) \tag{22}$$
$$Poss(Put(x, y), s) \leftrightarrow Holds(Holding(x), s) \wedge Holds(Clear(y), s) \tag{23}$$
$$Poss(Put(x, T), s) \leftrightarrow Holds(Holding(x), s) \tag{24}$$

We will use the above axioms instead of the axioms (19)-(21).

Given the above axioms and some axioms about the , we can now deduce whether the action $Pickup(A)$ is possible in the initial situation, i.e., $Poss(Pickup(A), S_0)$ is true. How?

Effect axioms: Actions have effects. For example, if the agent pickups a block he will hold it in the successor situation.

$$Poss(Pickup(x), s) \rightarrow Holds(Holding(x), Do(Pickup(x), s)) \tag{25}$$

$$Poss(Put(x, y), s) \rightarrow Holds(On(x, y), Do(Put(x, y), s)) \tag{26}$$

$$Poss(Put(x, T), s) \rightarrow Holds(On(x, T), Do(Put(x, T), s)) \tag{27}$$

$$\begin{aligned} Poss(Pickup(x), s) \rightarrow [Holds(On(x, y), s) \\ \rightarrow Holds(\neg On(x, y), Do(Pickup(x), s))] \end{aligned} \tag{28}$$

But, the above equations only represent the 'positive effects' of actions (what becomes true) and 'negative effects' of actions (what becomes false) as well. In (28), for example, if the agent pickups block x which is on a block y in the situation s then $On(x, y)$ will not hold in $Do(Pickup(x), s)$; or if the agent puts block x on a block y in the situation s then $Clear(y)$ will not hold in $Do(Put(x, y), s)$, etc. **Try to complete the set of effect axioms.**

Furthermore, there are fluents whose values do not change after the execution of an action. For example, if x is on y in situation s ($Holds(On(x, y), s)$ is true) and the agent puts block z on top of x, then x is still on y in situation $Do(Put(z, x), s)$ ($Holds(On(x, y), Do(Put(z, x), s))$ is true). Representing the unchange effects of actions require axioms of the form

$$Holds(On(x, y), s) \rightarrow Holds(On(x, y), Do(Pickup(z), s))$$

These axioms are called *frame axioms.*

If we have n actions, m fluents, we would need $2 \times n \times m$ frame axioms. This is too many!!!

Representing frame axioms in a compact way is a challenge for quite sometime. To date, there are many solutions to the frame problems. Under reasonable assumptions, we can write, for each fluent f, one *successor state axiom* in the following form

$$Holds(f, Do(a, s)) \leftrightarrow \gamma_f^+(a, s) \vee (Holds(f, s) \wedge \gamma_f^-(a, s)) \tag{29}$$

where $\gamma_f^+(a, s)$ (resp. $\gamma_f^-(a, s)$) summarizes the conditions for the fluent f to become true (resp. false).

For example, in the block world we have

$$Holds(On(x, y), Do(a, s)) \leftrightarrow \gamma^+_{On(x,y)}(a, s) \vee (Holds(On(x, y), s) \wedge \gamma^-_{On(x,y)}(a, s)) \tag{30}$$

where

$\gamma^{+}_{On(x,y)}(a, s) = (a = Put(x, y)) \wedge Poss(a, s)$ and

$\gamma^{-}_{On(x,y)}(a, s) = (a \neq Pickup(x)) \wedge Poss(a, s)$.

Complete the description!

3.2 Projection Problem

Let us denote the situation calculus theory for the block world by $\mathcal{T}$. We can prove that

$$\mathcal{T} \models Holds(On(A, C), S_0)$$

$$\mathcal{T} \models Holds(\neg On(A, C), Do(Pickup(A), S_0))$$

In general, we can prove that

$$\mathcal{T} \models Holds(f, Do(a_n, \dots, Do(a_1, S_0))) \tag{31}$$

for a fluent f and a sequence of actions $a_1, \dots, a_n$.

We also write

$$\mathcal{T} \models Holds(f, Do(\alpha, S_0))) \tag{32}$$

$\alpha = [a_1, \dots, a_n]$.

3.3 Planning in Situation Calculus

The projection shows that we could formulate a planning task by asking for a sequence of actions (α) that makes the goal (say, a fluent f) true in the resulting situation of executing α in S_0. That is, for a planning task of achieving the goal φ (a fluent formula), we ask for a sequence of actions α such that

$$\mathcal{T} \models Holds(\varphi, Do(\alpha, S_0))) \tag{33}$$

Example 3.1 *Find a sequence of actions that achieves the goal of having A on D. That is, we need to find α such that $\mathcal{T} \models Holds(On(A, D), Do(\alpha, S_0))$.*

It is easy to see that one of the possibility is

$\alpha = [Put(A, E), Put(C, T), Put(A, D)]$. **Why?**

Can you find another sequence of actions?

4 Planning in Situation Calculus

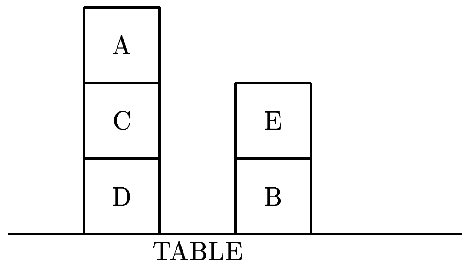

Figure 2: A Block World

We present a slight different situation calculus theory for the block world in the above picture. It consists of:

1. Constants: A, B, C, D, E, T, two unary (domain) predicate $Block$ and $Location$, where $Block = \{A, B, C, D, E\}$ and $Location = \{A, B, C, D, E, T\}$;

2. Actions: $Pickup(x, y)$, $Put(x, y)$, and $Put(x, T)$ (Instead of $Pickup(x)$ we use $Pickup(x, y)$ – to say that the agent pickup the block x from a location y – which can be a block or the table);

3. Fluents: $On(x, y)$, $On(x, T)$, $Holding(x)$, $Clear(x)$.

4. The set of precondition axioms:

$$\begin{aligned} Poss(Pickup(x, y), s) \quad &\leftrightarrow \quad x \neq T \wedge Location(y) \wedge \\ & \qquad \forall z(Holds(\neg Holding(z), s)) \wedge \\ & \qquad Holds(Clear(x), s) \qquad (34) \\ Poss(Put(x, y), s) \quad &\leftrightarrow \quad Block(x) \wedge Block(y) \wedge x \neq y \wedge \\ & \qquad Holds(Holding(x), s) \wedge \\ & \qquad Holds(Clear(y), s) \qquad (35) \\ Poss(Put(x, T), s) \quad &\leftrightarrow \quad Block(x) \wedge Holds(Holding(x), s) \qquad (36) \end{aligned}$$

5. The set of effect axioms:

$$Poss(Pickup(x, y), s) \rightarrow Holds(Holding(x), Do(Pickup(x, y), s)) \qquad (37)$$

$$Poss(Pickup(x, y), s) \rightarrow Holds(\neg On(x, y), Do(Pickup(x, y), s)) \qquad (38)$$

$$\begin{aligned} Poss(Pickup(x, y), s) \rightarrow [y \neq T \rightarrow \\ Holds(Clear(y), Do(Pickup(x, y), s))] \end{aligned} \qquad (39)$$

$$Poss(Put(x, y), s) \rightarrow Holds(On(x, y), Do(Put(x, y), s)) \qquad (40)$$

$$Poss(Put(x, y), s) \rightarrow Holds(\neg Holding(x), Do(Put(x, y), s)) \qquad (41)$$

$$Poss(Put(x, y), s) \rightarrow Holds(\neg Clear(y), Do(Put(x, y), s)) \qquad (42)$$

$$Poss(Put(x, T), s) \rightarrow Holds(\neg Holding(x), Do(Put(x, T), s)) \qquad (43)$$

$$Poss(Put(x, T), s) \rightarrow Holds(On(x, T), Do(Put(x, T), s)) \qquad (44)$$

6. The set of successor state axioms:

$$\begin{array}{c} Holds(On(x,y), Do(a,s)) \leftrightarrow (a = Put(x,y)) \wedge Poss(a,s) \\ \vee(Holds(On(x,y),s) \wedge \neg(a = Pickup(x,y) \wedge Poss(a,s))) \end{array} \tag{45}$$

$$\begin{array}{c} Holds(On(x,T), Do(a,s)) \leftrightarrow (a = Put(x,T)) \wedge Poss(a,s) \\ \vee(Holds(On(x,T),s) \wedge \neg(a = Pickup(x,T) \wedge Poss(a,s))) \end{array} \tag{46}$$

$$\begin{array}{c} Holds(Holding(x), Do(a,s)) \leftrightarrow (a = Pickup(x,y)) \wedge Poss(a,s) \\ \vee(Holds(Holding(x),s)\wedge \\ \neg((a = Put(x,z) \vee a = Put(x,T)) \wedge Poss(a,s))) \end{array} \tag{47}$$

$$\begin{array}{c} Holds(Clear(x), Do(a,s)) \leftrightarrow (a = Pickup(y,x)) \wedge Poss(a,s) \\ \vee(Holds(Clear(x),s)\wedge \\ \neg(a = Put(y,x) \wedge Poss(a,s))) \end{array} \tag{48}$$

What do I miss in the above formulas (45)-(48)?

7. Initial situation:

$$Holds(On(A,C), S_0) \tag{49}$$
$$Holds(On(C,D), S_0) \tag{50}$$
$$Holds(On(D,T), S_0) \tag{51}$$
$$Holds(On(E,B), S_0) \tag{52}$$
$$Holds(On(B,T), S_0) \tag{53}$$

Let us denote the theory by $\mathcal{T}$. Under the assumptions that

1. changes are caused by actions and
2. no thing happens before S_0

we can view each situation as a sequence of actions. Thus, we can prove what is true/false in each situation and hence we can formulate the planning task as follows.

Planning problem: Given a situation calculus $\mathcal{T}$ and a fluent formula φ. A *plan achieving* φ is a sequence of actions $a_1, \ldots, a_n$ such that

$$\mathcal{T} \models Holds(\varphi, Do(a_n, \ldots, Do(a_1, S_0)))$$

Example 4.1 *[Planning in Block World]*

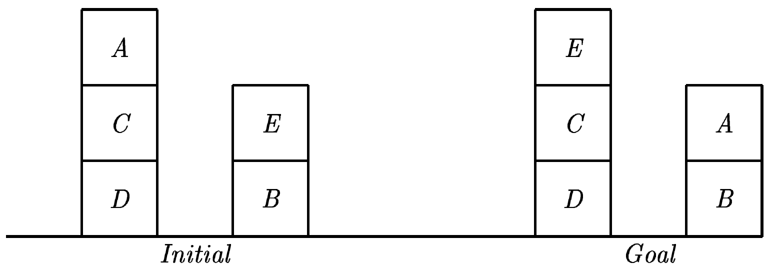

Figure 3: A Planning Problem in Block World

Solving a planning problem means to finding a plan! How can we find a plan?

Generate and Test: The most simple way to find a plan. We can generate a sequence of actions, say α, and then test for $\mathcal{T} \models Holds(\varphi, Do(\alpha, S_0))$. So, we have the following algorithm (pseudo code for implementation):

Algorithm 4.1 (First Idea) *Given are $\mathcal{T}$ and φ*

main

while true *do*

α = *generate_sequence_action*

if testOK(α) *return* α *endwhile*

function

generate_sequence_action

return a sequence of actions

function

testOK(α)

if $\mathcal{T} \models Holds(\varphi, Do(a_n, \ldots, Do(a_1, S_0)))$ *return* true

else return false

Implementation problem: Checking if φ holds in $Do(\alpha, S_0)$, i.e., checking whether

$$(*) \quad \mathcal{T} \models Holds(\varphi, Do(\alpha, S_0))$$

holds or not.

Another problem is how to represent the goal?

Answer: We can use a theorem prover, that tells us what is true in $Do(\alpha, S_0)$? to check (*). We can simplify our life (for now) by representing the goal by a set of fluents. Thus, the goal of the planning problem in Example 4.1 can be represented by the set

$$\{On(A, B), On(C, D), On(D, T), On(E, C), On(B, T)\}.$$

Running the algorithm:

The set of actions: $\{Put(x, y), Put(x, T), Pickup(x, y)$

Iteration No. 1

Possible action in S_0: $Pickup(A, C), Pickup(E, B)$

We then have two situation $S_1 = Pickup(A, C)$ and $S_2 = Pickup(E, B)$. In S_1, the following fluents are true: $On(C, D), On(D, T), On(E, B), On(B, T), Holding(A)$ whereas $On(A, C), On(C, D), On(D, T), On(B, T), Holding(E)$ are true in S_2. (Notice that we have not listed ALL the fluents that are true in each situation, i.e., this list is incomplete.)

The following picture illustrates the different possible situations (in a separate file).

Appendix – The Wumpus World

The wumpus world [**?**] is described as follows. The wumpus world is a grid of squares surrounded by walls, where each square can contain agents and objects. The agent always starts in the lower left corner. The agent task is to find the gold, return to where he starts and climb out of the cave. (the picture is in a separate file).