Points to recall

- Heuristic function must be ____________ for A* Graph Search Algorithm to give optimal solution.

Points to recall

- Heuristic function must be ____________ for A* Graph Search Algorithm to give optimal solution.
- The heuristic function that ____________ will lead to a fewer number of nodes being expanded.

Relaxed 8-puzzle problem

- To the original state space for the 8-puzzle problem we add the following:

Relaxed 8-puzzle problem

- To the original state space for the 8-puzzle problem we add the following:
  - We allow additional states where more than one tile can be present at a position and more than one blank space can be present in the puzzle.

Relaxed 8-puzzle problem

- To the original state space for the 8-puzzle problem we add the following:
  - We allow additional states where more than one tile can be present at a position and more than one blank space can be present in the puzzle.
  - We allow additional actions where any tile can be moved to an adjacent position, which may result in more than one tile being present at the same position.

Relaxed 8-puzzle problem

- To the original state space for the 8-puzzle problem we add the following:
  - We allow additional states where more than one tile can be present at a position and more than one blank space can be present in the puzzle.
  - We allow additional actions where any tile can be moved to an adjacent position, which may result in more than one tile being present at the same position.
- Each action can move one tile one position closer to its goal position.

Relaxed 8-puzzle problem

- To the original state space for the 8-puzzle problem we add the following:
  - We allow additional states where more than one tile can be present at a position and more than one blank space can be present in the puzzle.
  - We allow additional actions where any tile can be moved to an adjacent position, which may result in more than one tile being present at the same position.
- Each action can move one tile one position closer to its goal position.
- What will be the cost of the shortest path in this relaxed state space?

Relaxed 8-puzzle problem

- To the original state space for the 8-puzzle problem we add the following:
  - We allow additional states where more than one tile can be present at a position and more than one blank space can be present in the puzzle.
  - We allow additional actions where any tile can be moved to an adjacent position, which may result in more than one tile being present at the same position.
- Each action can move one tile one position closer to its goal position.
- What will be the cost of the shortest path in this relaxed state space? Manhattan distance (h_2)

h_2 is consistent in the relaxed state space for 8-puzzle

To prove : $h_2(n) \leq c(n, a, n') + h_2(n')$
Proof:

Approach to generating a heuristic function

1. Relax a problem in such a manner that the shortest path cost in the relaxed problem becomes easy to compute.

Approach to generating a heuristic function

1. Relax a problem in such a manner that the shortest path cost in the relaxed problem becomes easy to compute.
2. The shortest path costs in any relaxed problem problem will be consistent.

Approach to generating a heuristic function

1. Relax a problem in such a manner that the shortest path cost in the relaxed problem becomes easy to compute.
2. The shortest path costs in any relaxed problem problem will be consistent.
3. Therefore, these shortest path costs can form a consistent heuristic for the original problem.

Another way to relax the 8-puzzle problem

- To the original state space for the 8-puzzle problem we add the following:

Another way to relax the 8-puzzle problem

- To the original state space for the 8-puzzle problem we add the following:
  - We allow additional states where more than one tile can be present at a position and more than one blank space can be present in the puzzle.

Another way to relax the 8-puzzle problem

- To the original state space for the 8-puzzle problem we add the following:
  - We allow additional states where more than one tile can be present at a position and more than one blank space can be present in the puzzle.
  - We allow additional actions where any tile can be moved to any position, which may result in more than one tile being present at the same position.

Another way to relax the 8-puzzle problem

- To the original state space for the 8-puzzle problem we add the following:
  - We allow additional states where more than one tile can be present at a position and more than one blank space can be present in the puzzle.
  - We allow additional actions where any tile can be moved to any position, which may result in more than one tile being present at the same position.
- Each action can move one tile to its correct position.

Another way to relax the 8-puzzle problem

- To the original state space for the 8-puzzle problem we add the following:
  - We allow additional states where more than one tile can be present at a position and more than one blank space can be present in the puzzle.
  - We allow additional actions where any tile can be moved to any position, which may result in more than one tile being present at the same position.
- Each action can move one tile to its correct position.
- What will be the cost of the shortest path in this relaxed state space?

Another way to relax the 8-puzzle problem

- To the original state space for the 8-puzzle problem we add the following:
  - We allow additional states where more than one tile can be present at a position and more than one blank space can be present in the puzzle.
  - We allow additional actions where any tile can be moved to any position, which may result in more than one tile being present at the same position.
- Each action can move one tile to its correct position.
- What will be the cost of the shortest path in this relaxed state space? Number of misplaced tiles (h_1)

h_1 is consistent in the relaxed state space for 8-puzzle

To prove : $h_1(n) \leq c(n, a, n') + h_1(n')$
Proof:

- Knowledge base

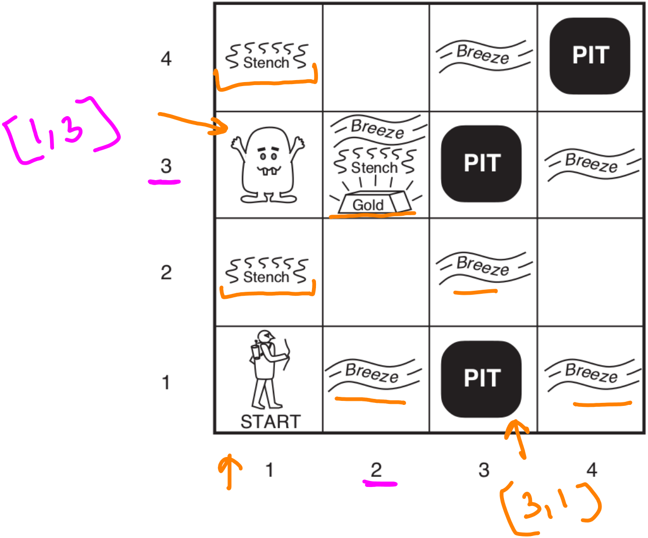
- Knowledge base
- Propositional logic

Chapter 7: Logical Agents

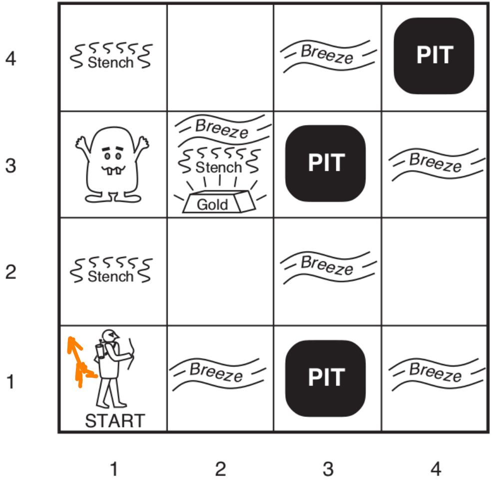
- Knowledge base
- Propositional logic
- Inference

Logical Agents

[2, 3]



Logical Agents



Percept in each time step: [*Stench*, *Breeze*, *Glitter*, *Bump*, *Scream*]

First Two Steps

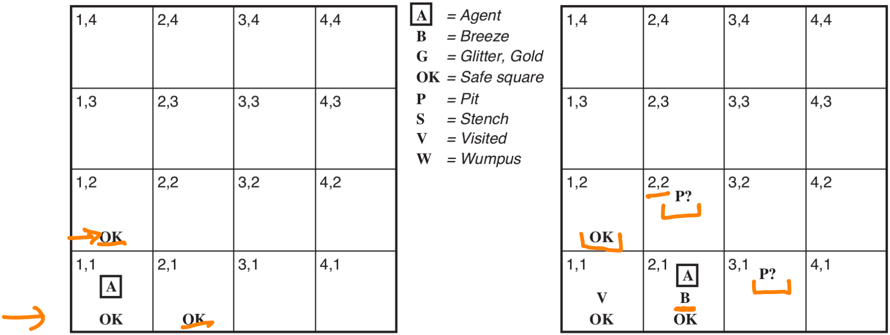


Figure 7.3 The first step taken by the agent in the wumpus world. (a) The initial situation, after percept [*None*, *None*, *None*, *None*, *None*]. (b) After one move, with percept [*None*, *Breeze*, *None*, *None*, *None*].

Next Steps

[2,1] Or [3,1]

→ 2,1

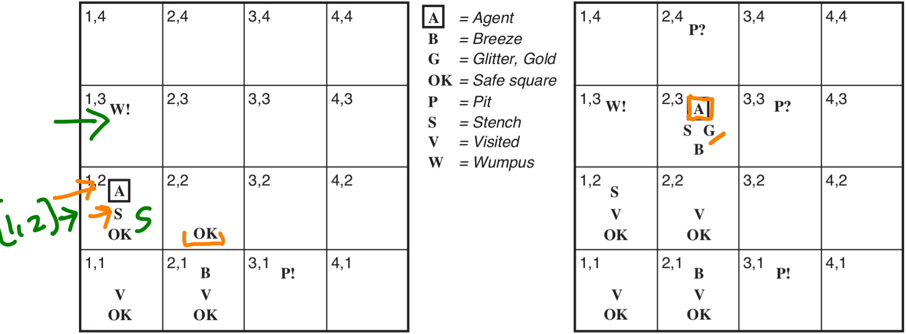
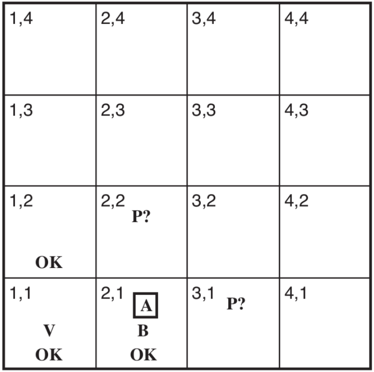


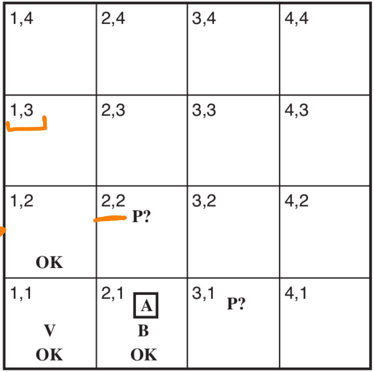
Figure 7.4 Two later stages in the progress of the agent. (a) After the third move, with percept [*Stench*, *None*, *None*, *None*, *None*]. (b) After the fifth move, with percept [*Stench*, *Breeze*, *Glitter*, *None*, *None*].

Wupus-world inference example



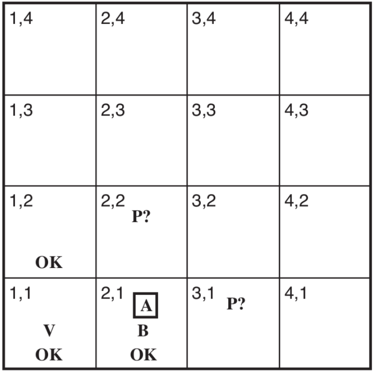
- KB contains agent's percepts (in the first 2 steps) and rules of the Wumpus world

Wupus-world inference example



- KB contains agent's percepts (in the first 2 steps) and rules of the Wumpus world
- Agent wants to know whether pit is present in [1,2],[2,2] and [1,3].

Wupus-world inference example



- KB contains agent's percepts (in the first 2 steps) and rules of the Wumpus world
- Agent wants to know whether pit is present in [1,2],[2,2] and [1,3].
- $\alpha_1 \equiv$ "No pit in [1,2]"
- $\alpha_2 \equiv$ "No pit in [2,2]"

Wupus-world inference example

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 **OK**	2,2 **P?**	3,2	4,2
1,1 **V** **OK**	2,1 $\boxed{\textbf{A}}$ **B** **OK**	3,1 **P?**	4,1

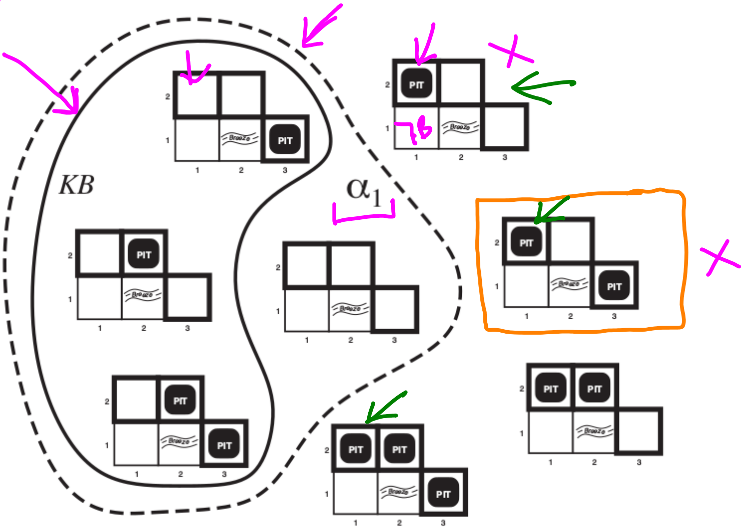
- KB contains agent's percepts (in the first 2 steps) and rules of the Wumpus world
- Agent wants to know whether pit is present in [1,2],[2,2] and [1,3].
- $\alpha_1 \equiv$ "No pit in [1,2]"
- $\alpha_2 \equiv$ "No pit in [2,2]"
- $KB \models \alpha_1$?

Wupus-world inference example

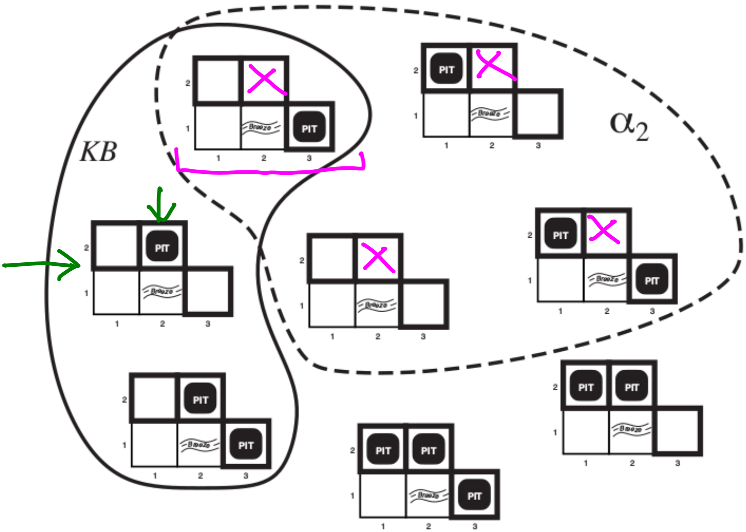
1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 **OK**	2,2 **P?**	3,2	4,2
1,1 **V** **OK**	2,1 $\boxed{\textbf{A}}$ **B** **OK**	3,1 **P?**	4,1

- KB contains agent's percepts (in the first 2 steps) and rules of the Wumpus world
- Agent wants to know whether pit is present in [1,2],[2,2] and ~~[1,3]~~. [3,1]
- $\alpha_1 \equiv$ "No pit in [1,2]"
- $\alpha_2 \equiv$ "No pit in [2,2]"
- $KB \models \alpha_1$?
- $KB \models \alpha_2$?

$\underbrace{KB}$ $\models \alpha_1$? $\alpha_1 \equiv$ no pit [1,2]



$KB \models \alpha_2$?



Knowledge Base and Models