Logical Agents

CH 7

Slides based in part on material from Andrea Thomaz and Maithilee Kunda

Overview

- Reflex agents
  - very little knowledge
- Goal-based agents
  - specific kind of knowledge (transition model, heuristics, ...)
- Knowledge-based agents
  - more generic kind of knowledge, and generic ways to combine == more flexible agents

Overvi

* Reflex agents
  * very little knowledge
* Goal-based agents
  * specific kind of knowledge (transition model, heuristics, ...)
* Knowledge-based agents
  * more generic kind of knowledge, and generic ways to combine == more flexible agents

Overvi

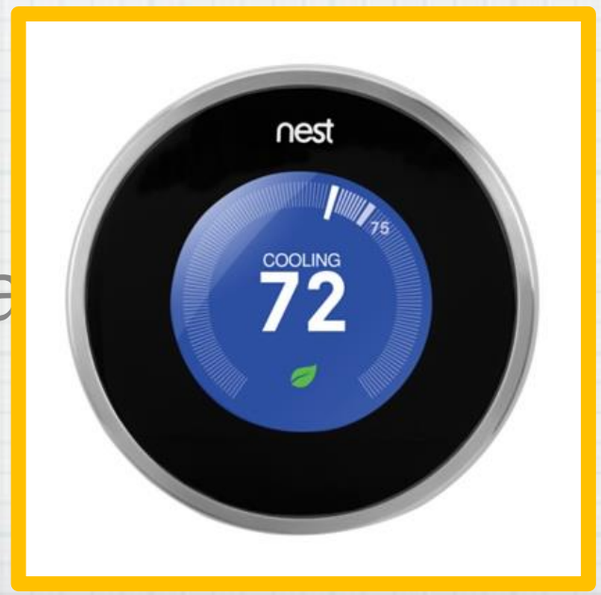
* Reflex agents
  * very little knowledge
* Goal-based agents
  * specific kind of knowledge (transition heuristics, ...)
* Knowledge-based agents
  * more generic kind of knowledge, and generic ways to combine == more flexible agents

Over

* Reflex agents
  * very little knowledge

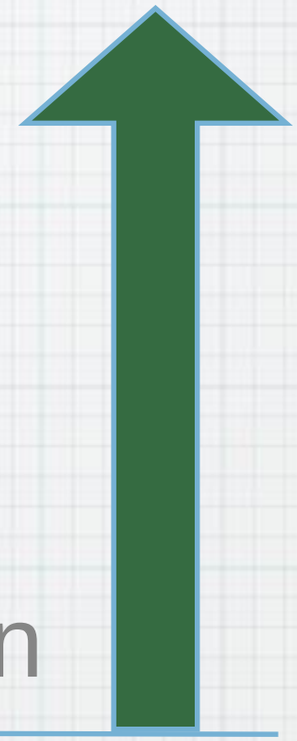
* Goal-based agents
  * specific kind of knowledg model, heuristics, ...)

* Knowledge-based agents
  * more generic kind of knowledge, a ways to combine == more flexible



Overview

- Intro basic Logical Agent Design
- Wumpus World -- example environment
- Concepts of Logic in General
- Propositional Logic
- First-order Logic
- Inferences with FOL



Knowledge-Based Agent Design

Knowledge Base (KB)

List of sentences, assertions about the world

Tell:
Add new sentences to the KB

S1

S2

…

…

Sn

Ask:
Answer some query based on current facts of the KB

Simple KB-agent

```
function KB-Agent(percept) returns an action
    static: KB, a knowledge base
            t, a counter, initially 0, indicating time

    Tell(KB, Make-Percept-Sentence(percept, t))
    action ← Ask(KB, Make-Action-Query(t))
    Tell(KB, Make-Action-Sentence(action, t))
    t ← t + 1
    return action
```

Simple KB-agent

```
function KB-Agent(percept) returns an action
    static: KB, a knowledge base
            t, a counter, initially 0, indicating time

    Tell(KB, Make-Percept-Sentence(percept, t))
    action ← Ask(KB, Make-Action-Query(t))
    Tell(KB, Make-Action-Sentence(action, t))
    t ← t + 1
    return action
```

The agent must be able to:

- Represent states, actions, etc.
- Incorporate new percepts
- Update internal representations of the world
- Deduce hidden properties of the world
- Deduce appropriate actions

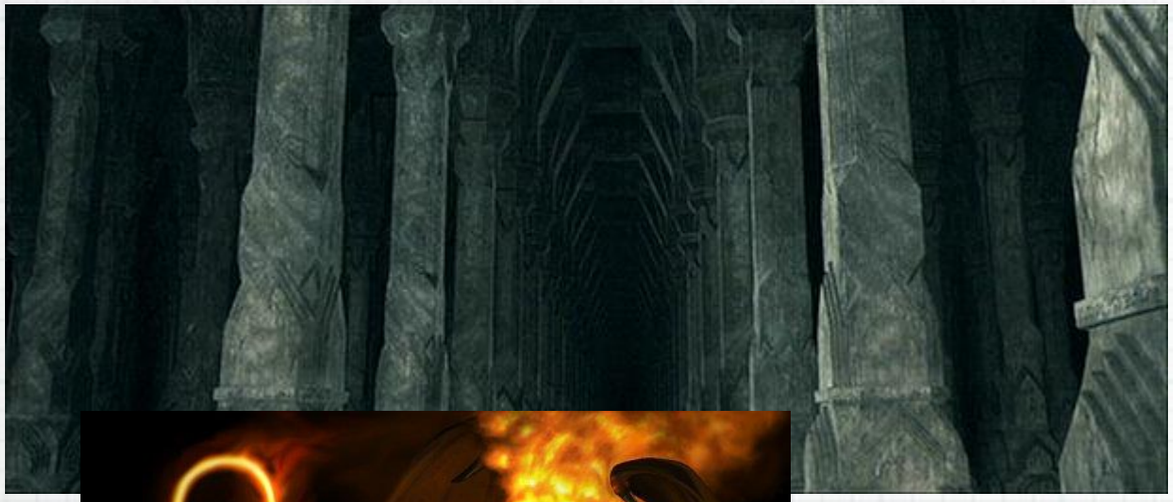
This is the big idea!

Logic Domain:

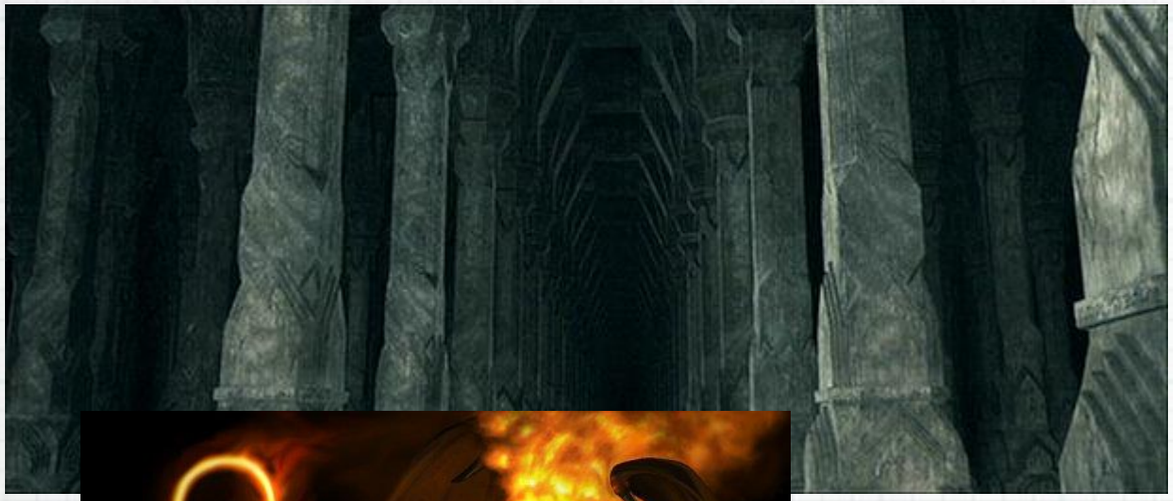
The Wumpus World

- System of caves and passageways
- There is a wumpus! It will eat you.
- You have only one arrow.
- There are bottomless pits.
- There is gold.

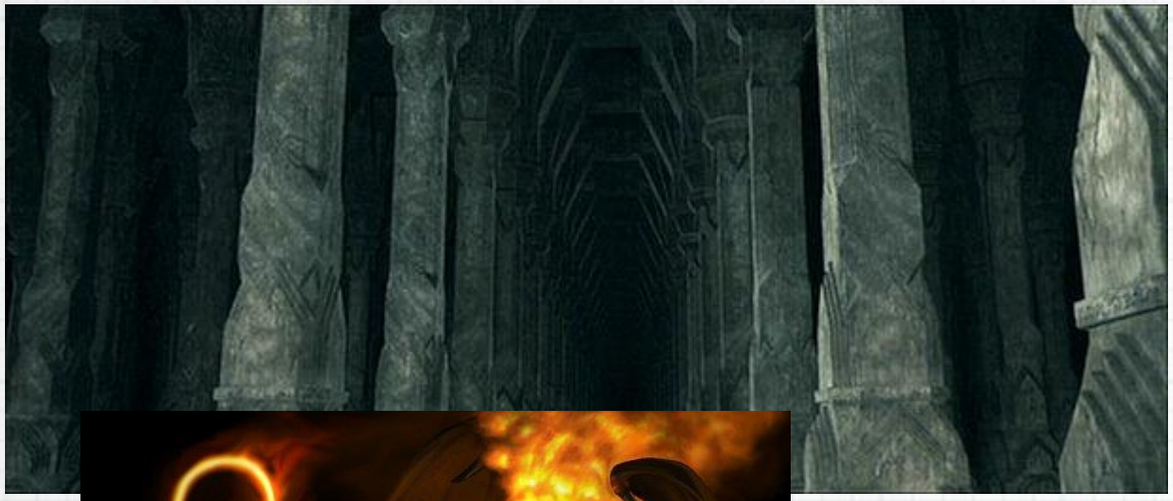
* System of caves and passageways
* There is a wumpus! It will eat you.
* You have only one arrow.
* There are bottomless pits.
* There is gold.



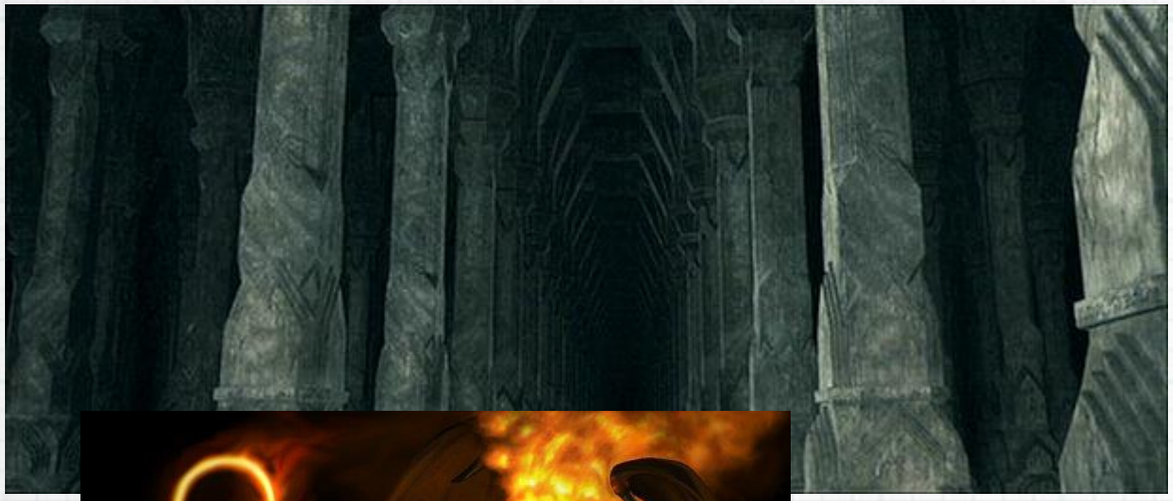
* System of caves and passageways
* There is a wumpus! It will eat you.
* You have only one arrow.
* There are bottomless pits.
* There is gold.

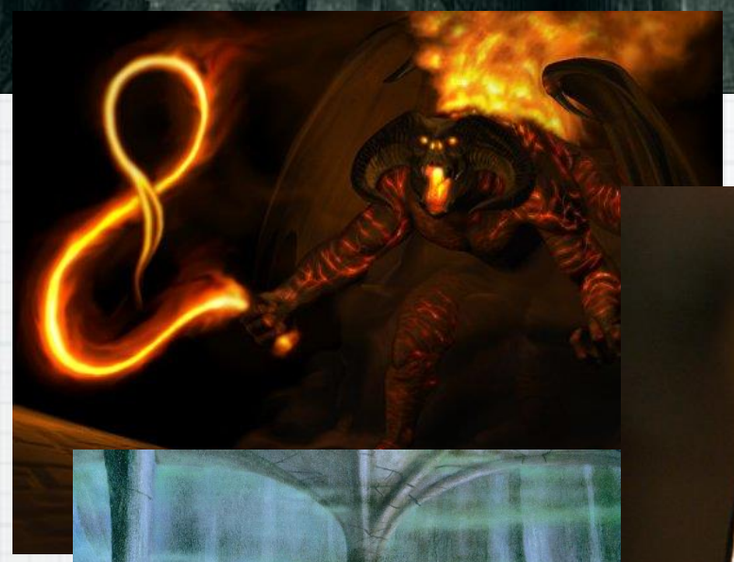


- System of caves and passageways
- There is a wumpus! It will eat you.
- You have only one arrow.
- There are bottomless pits.
- There is gold.



- System of caves and passageways
- There is a wumpus! It will eat you.
- You have only one arrow.
- There are bottomless pits.
- There is gold.



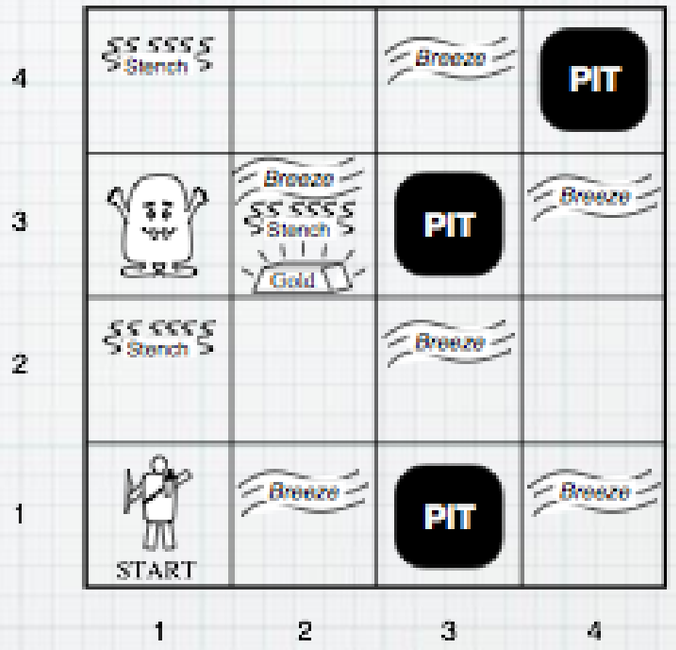


* System of caves and passageways
* There is a wumpus! It will eat you.
* You have only one ~~arrow.~~ wizard.
* There are bottomless pits.
* There is gold.

Wumpus World (PEAS)

Performance measure
- gold +1000, death -1000
- -1 per step, -10 for using the arrow

Environment
- Squares adjacent to wumpus are smelly
- Squares adjacent to pit are breezy
- Glitter iff gold is in the same square
- Shooting kills wumpus if you are facing it
- Shooting uses up the only arrow
- Grabbing picks up gold if in same square
- Releasing drops the gold in same square



Actuators Left turn, Right turn,
Forward, Grab, Release, Shoot

Sensors Breeze, Glitter, Smell

Wumpus World

Observable??

Deterministic??

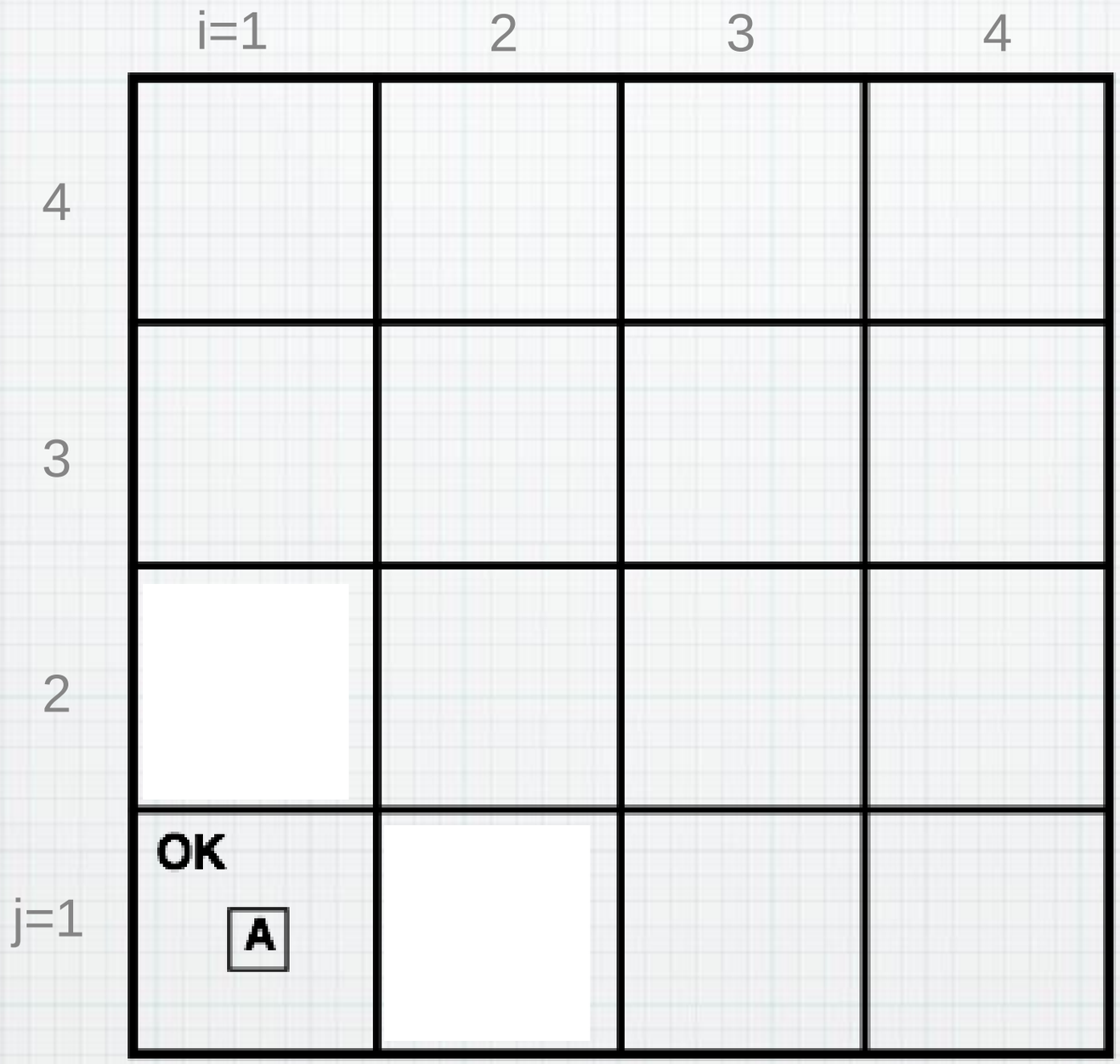
Episodic??

Static??

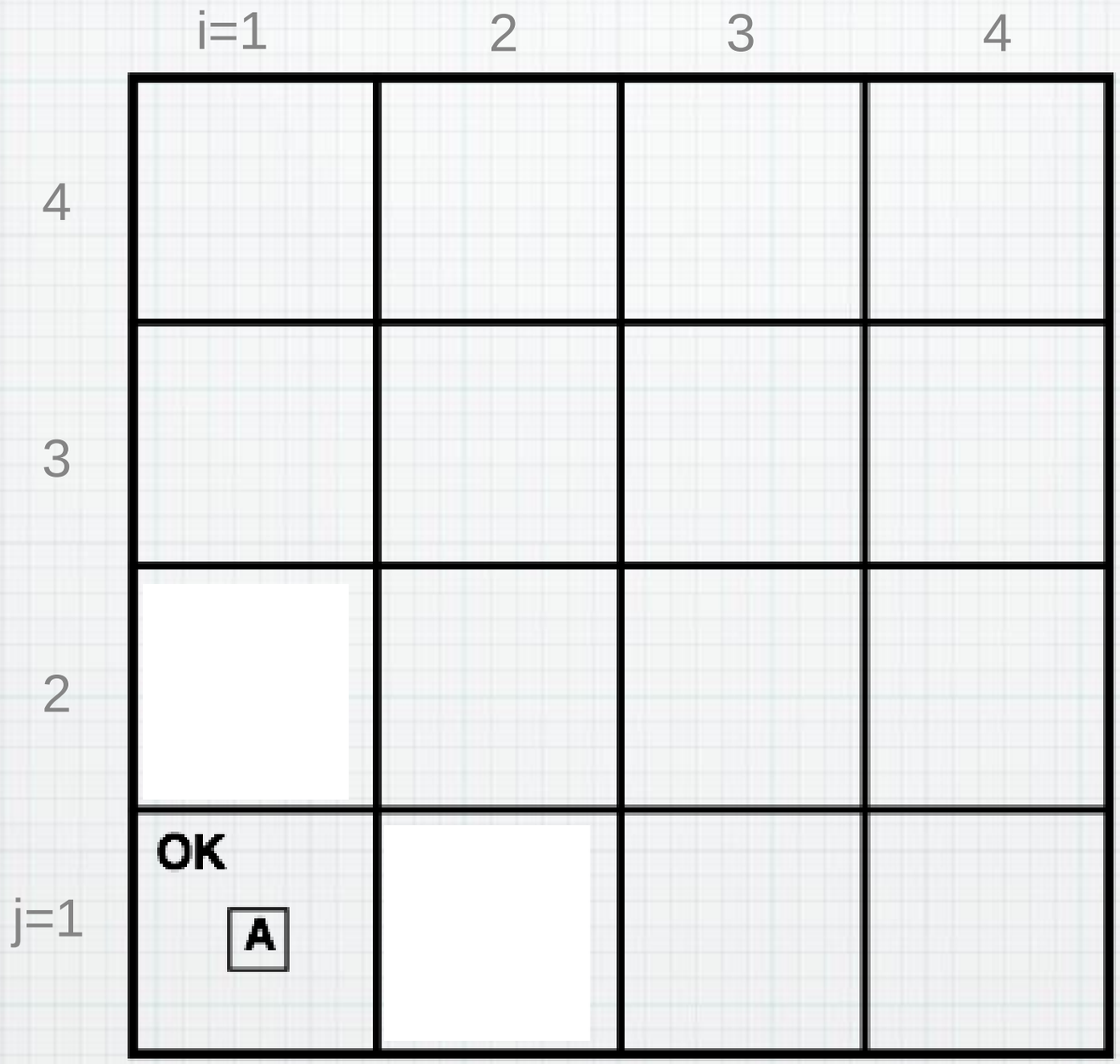
Discrete??

Single-agent??

Exploring in Wumpus World



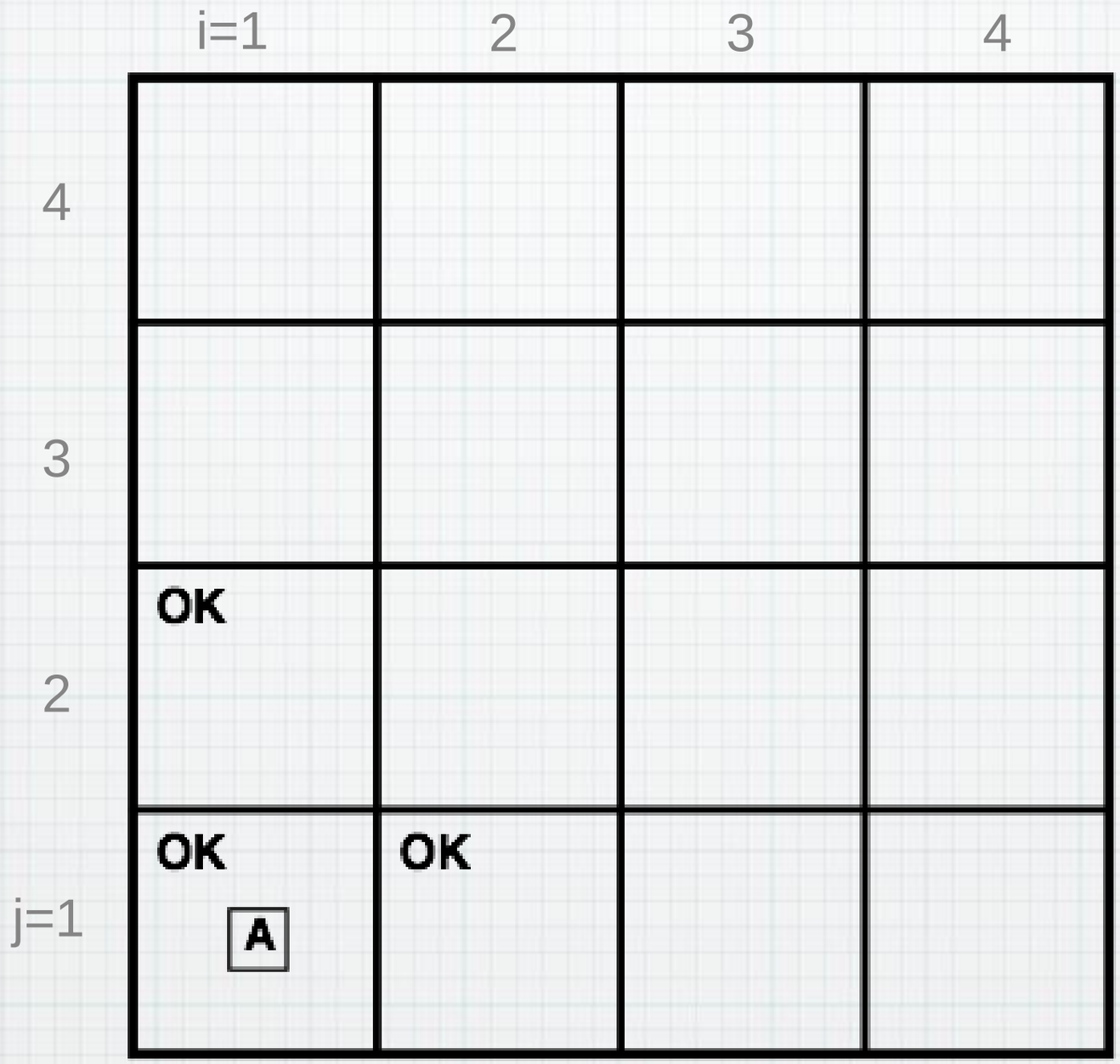
Exploring in Wumpus World



Percept [1,1]: None

Action: Forward to [1,2]

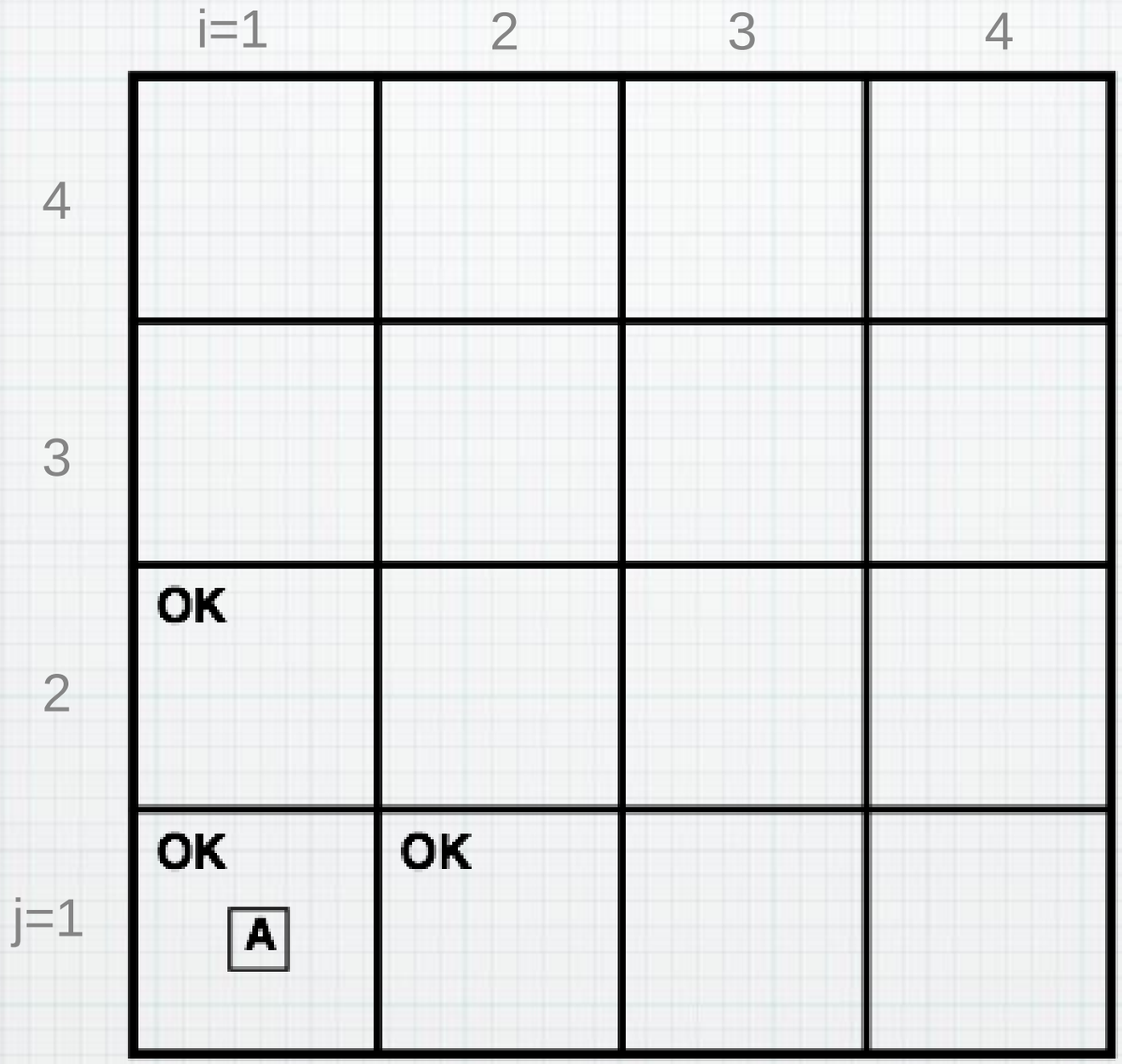
Exploring in Wumpus World



Percept [1,1]: None

Action: Forward to [1,2]

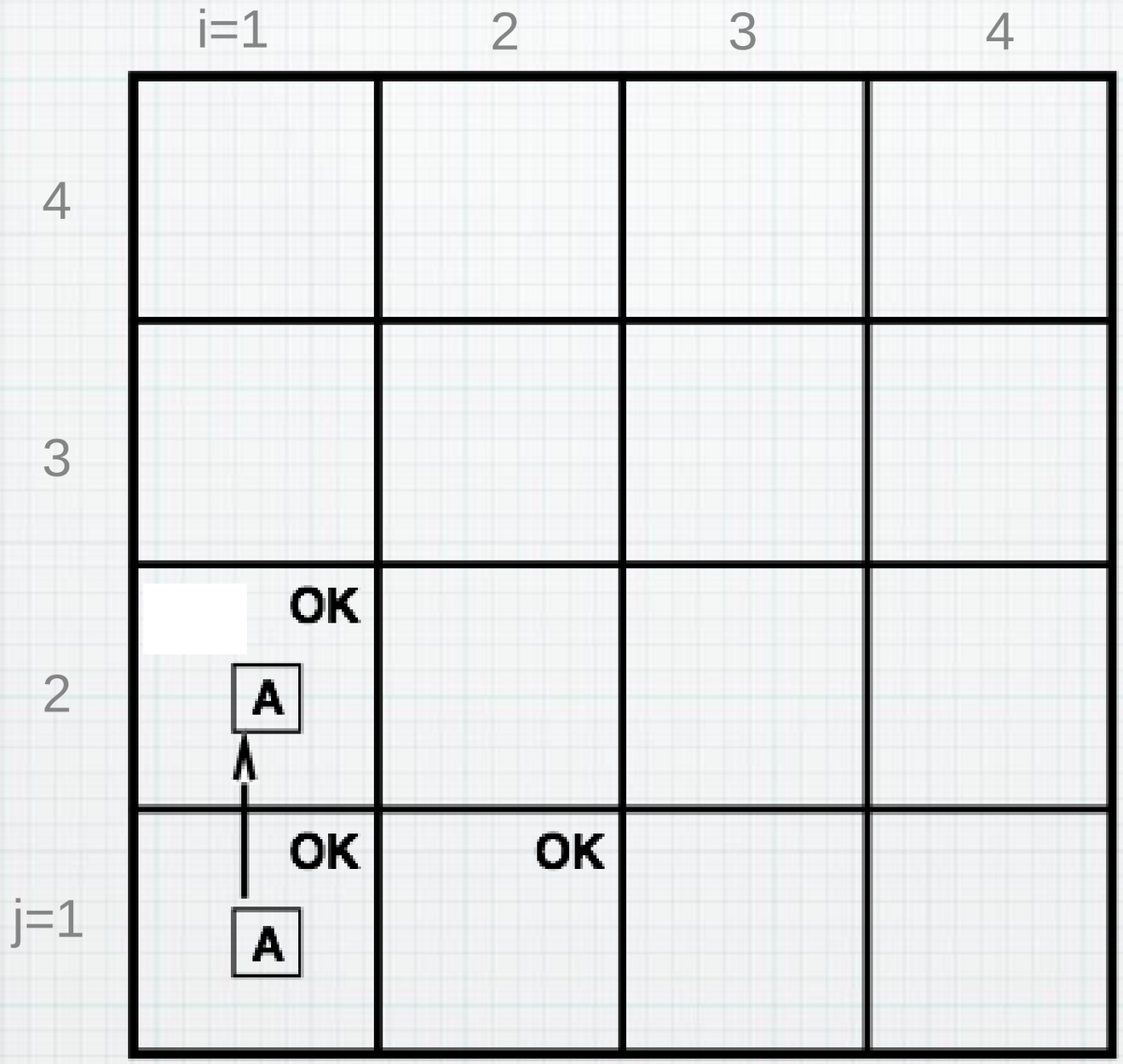
Exploring in Wumpus World



Percept [1,1]: None

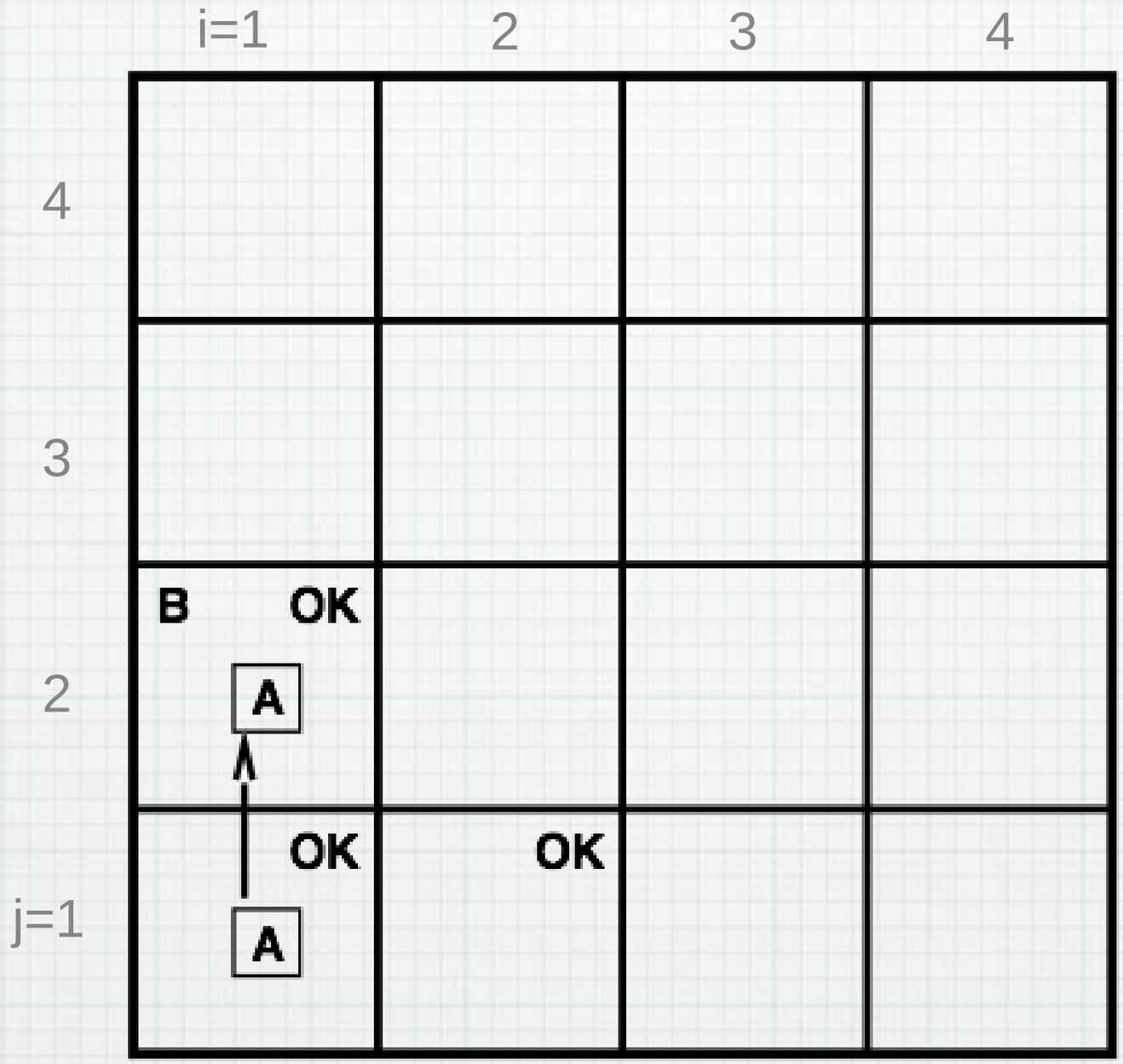
Action: Forward to [1,2]

Exploring in Wumpus World



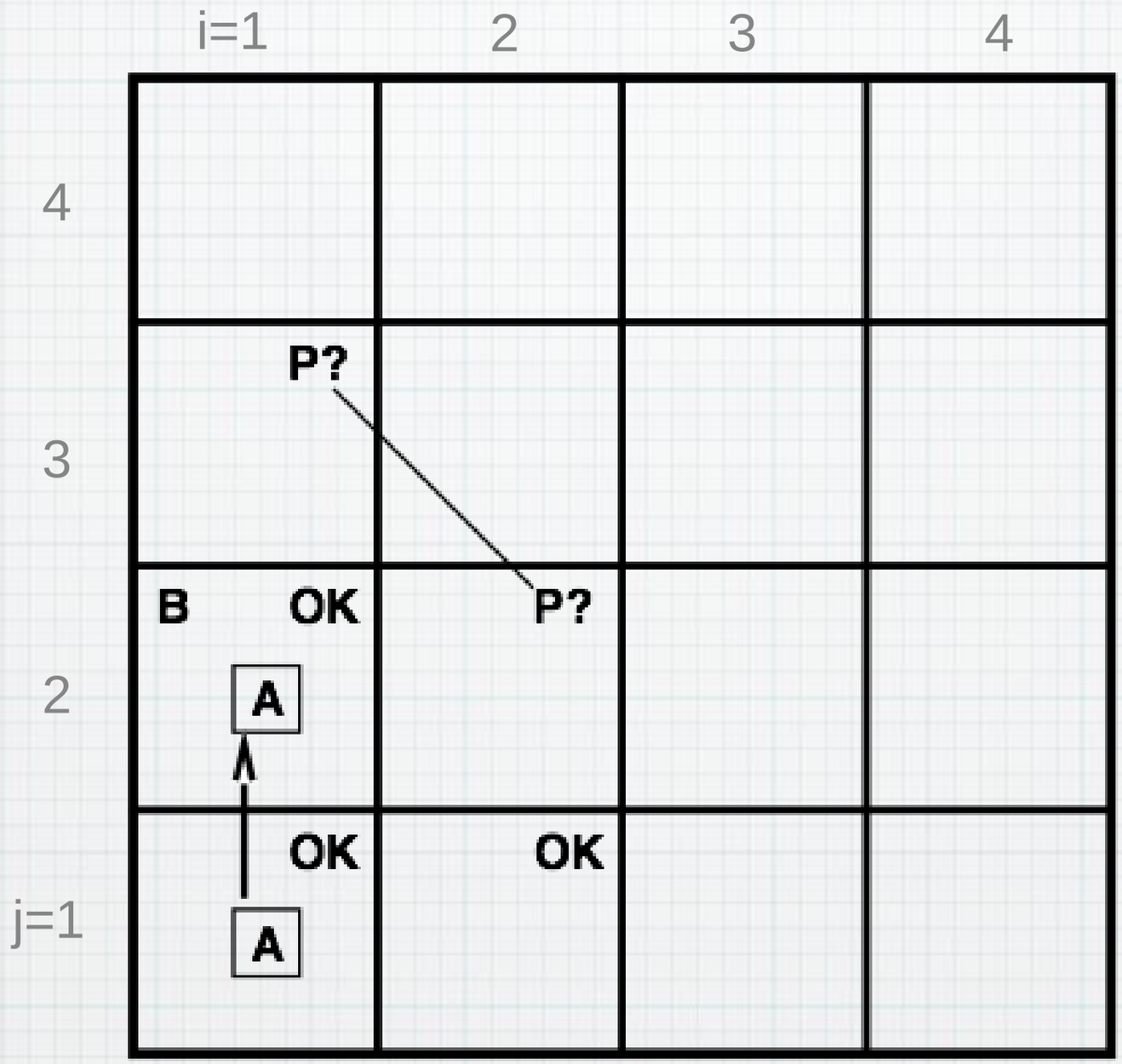
Percept [1,2]: Breeze

Exploring in Wumpus World



Percept [1,2]: Breeze

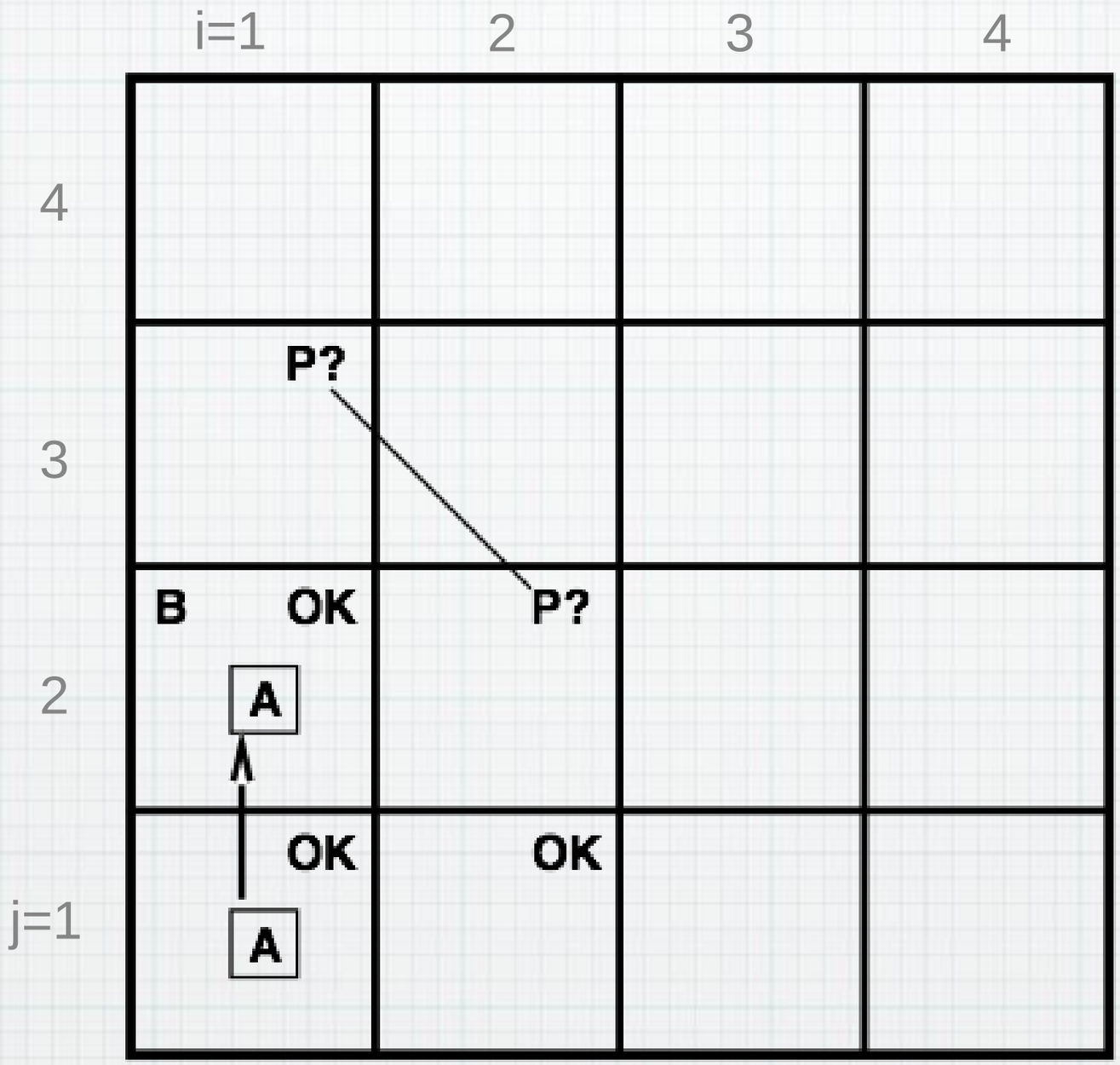
Exploring in Wumpus World



Percept [1,2]: Breeze; Pit in [1,3] or [2,2]

Action: turn back, go to [2,1]

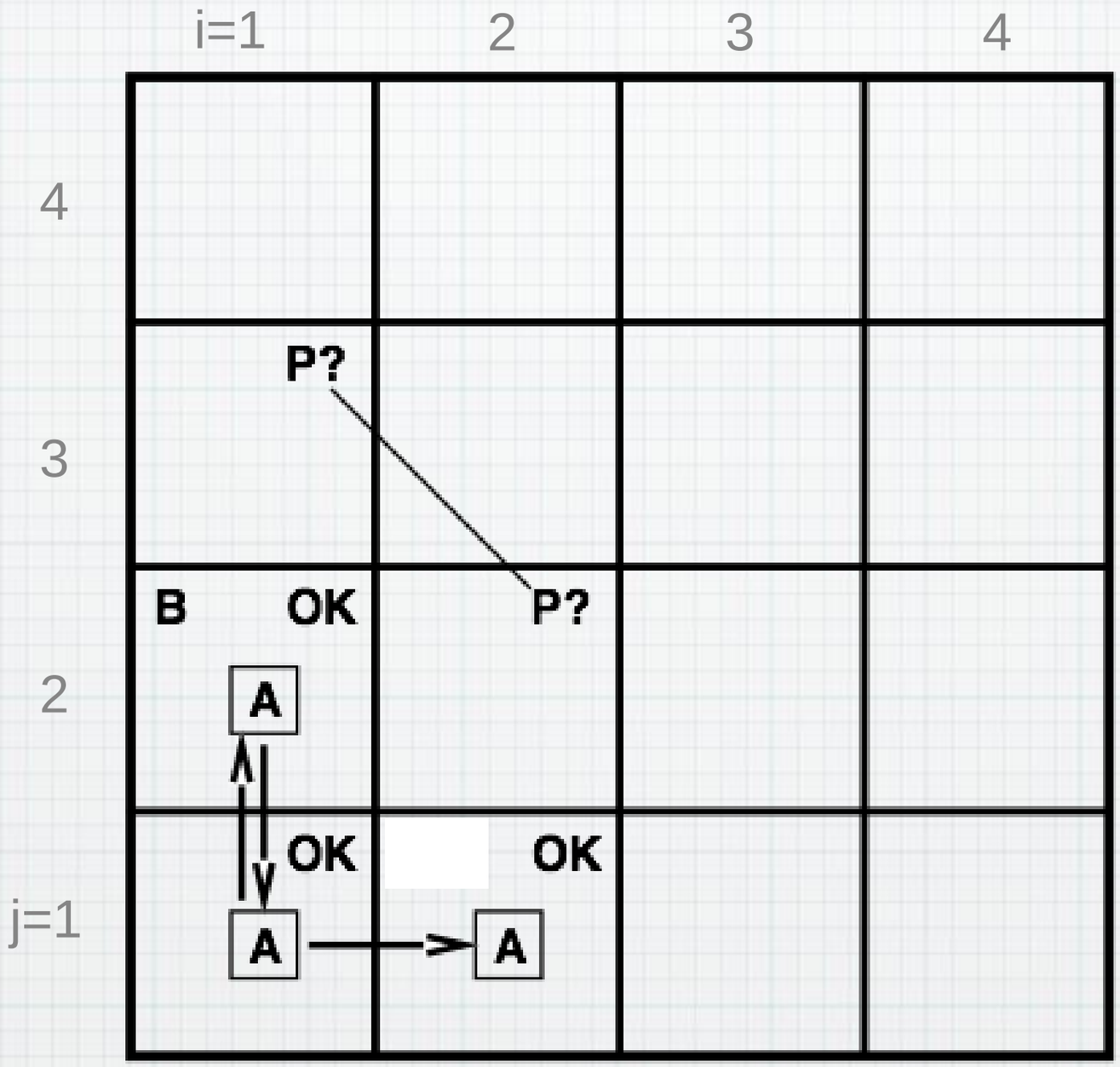
Exploring in Wumpus World



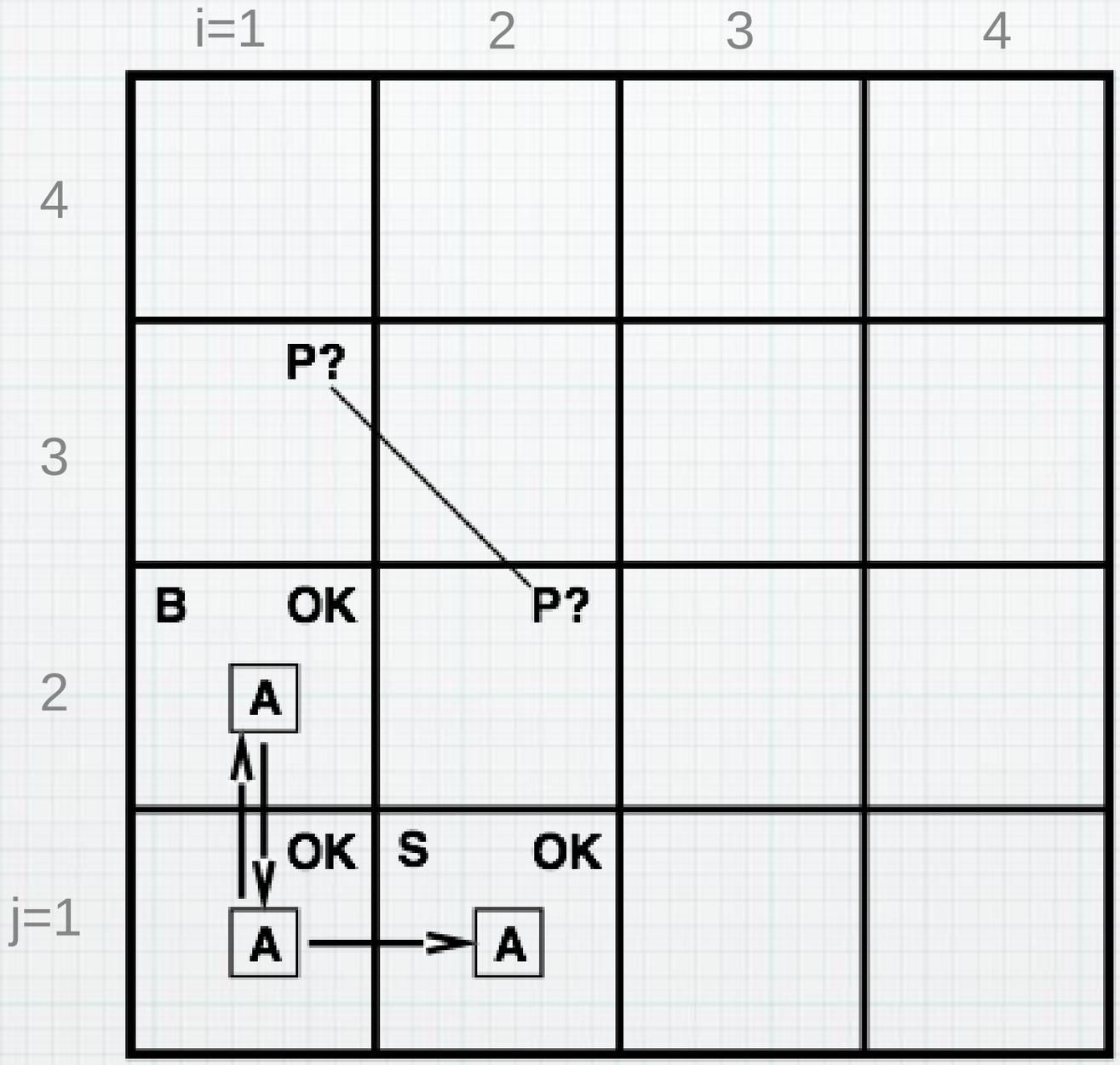
Percept [1,2]: Breeze; Pit in [1,3] or [2,2]

Action: turn back, go to [2,1]

Exploring in Wumpus World

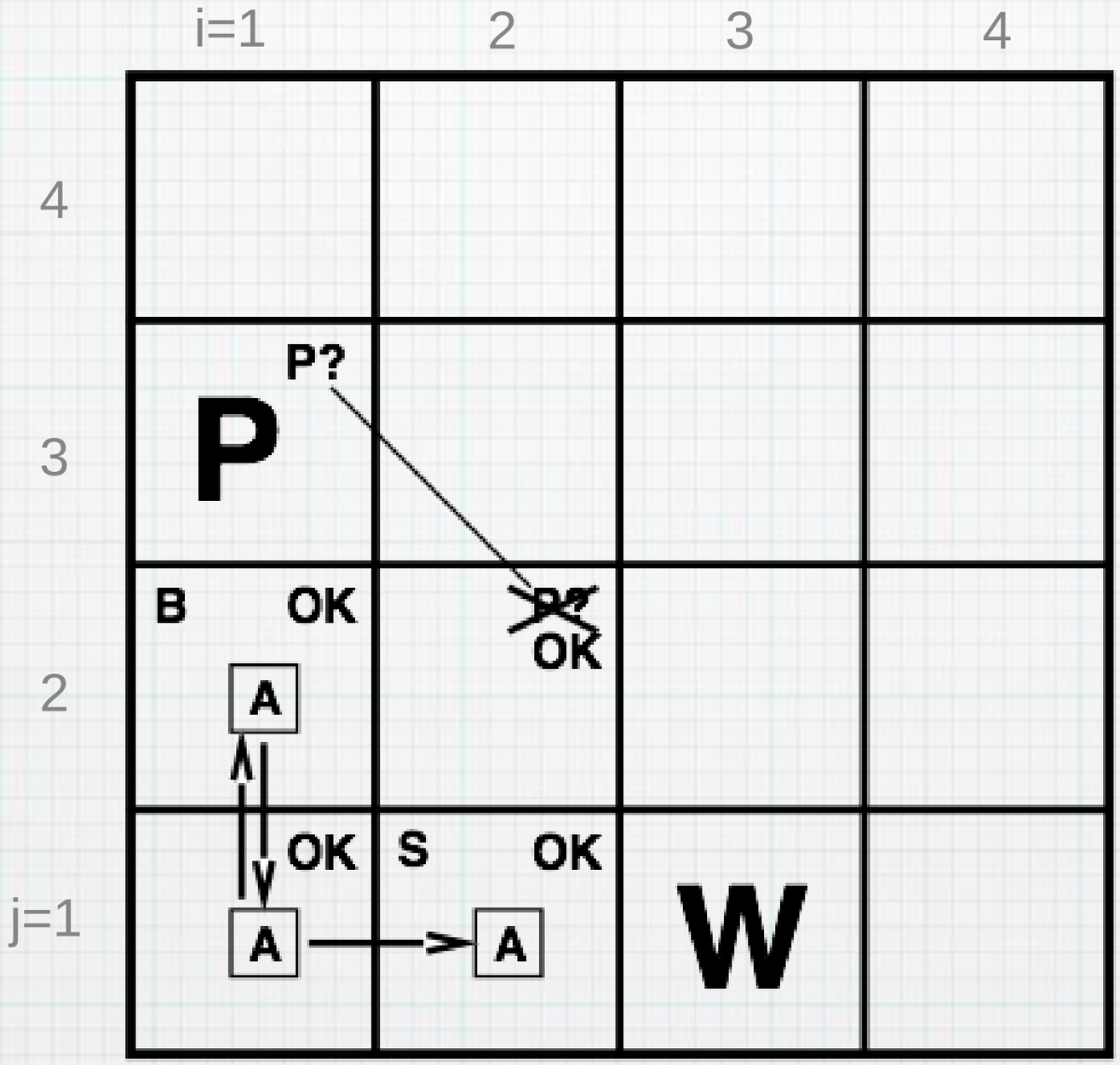


Exploring in Wumpus World



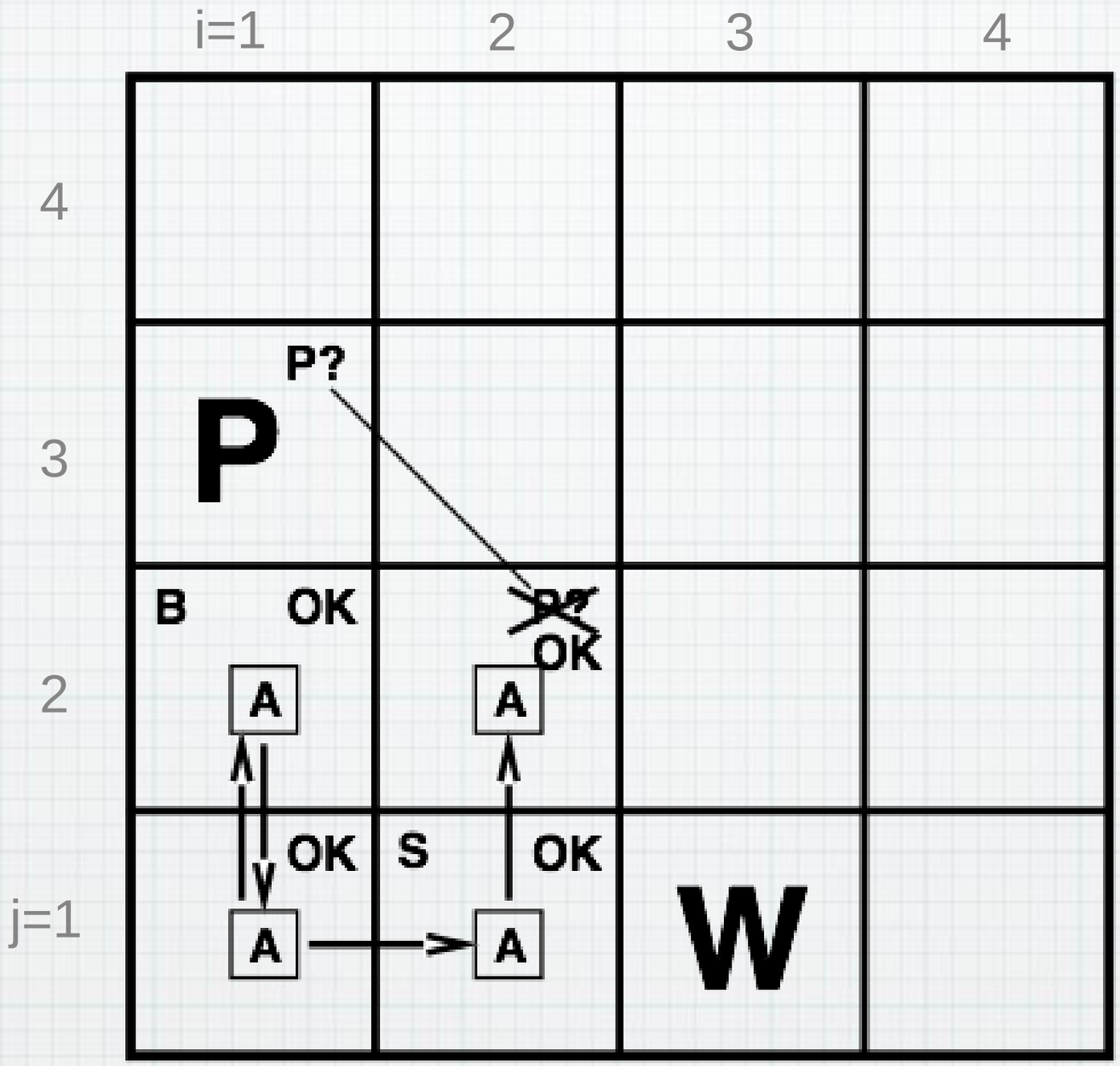
Percept [2,1]: Smelly;

Exploring in Wumpus World



Percept [2,1]: Smelly; W in [3,1] & [2,2] is OK

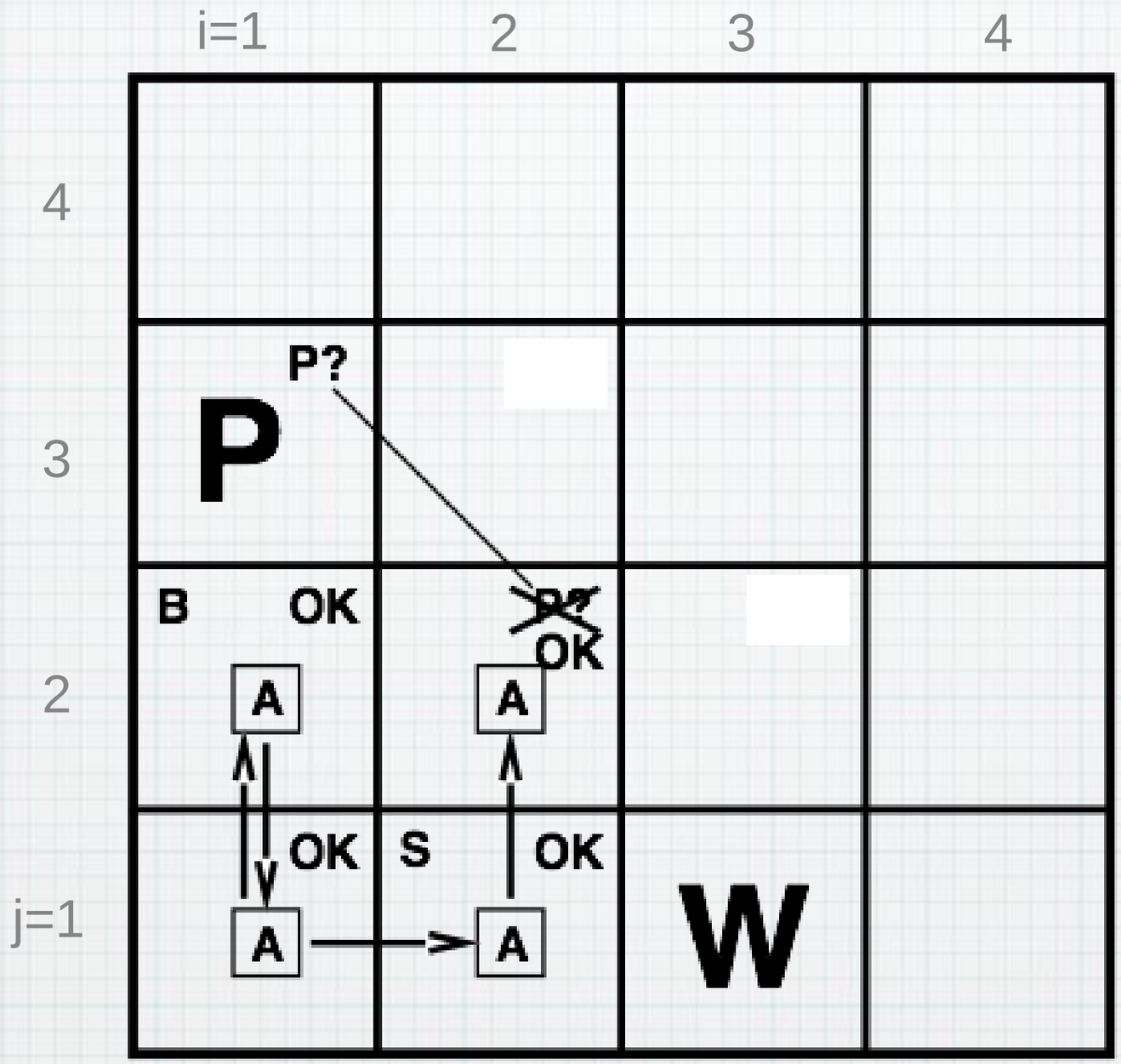
Exploring in Wumpus World



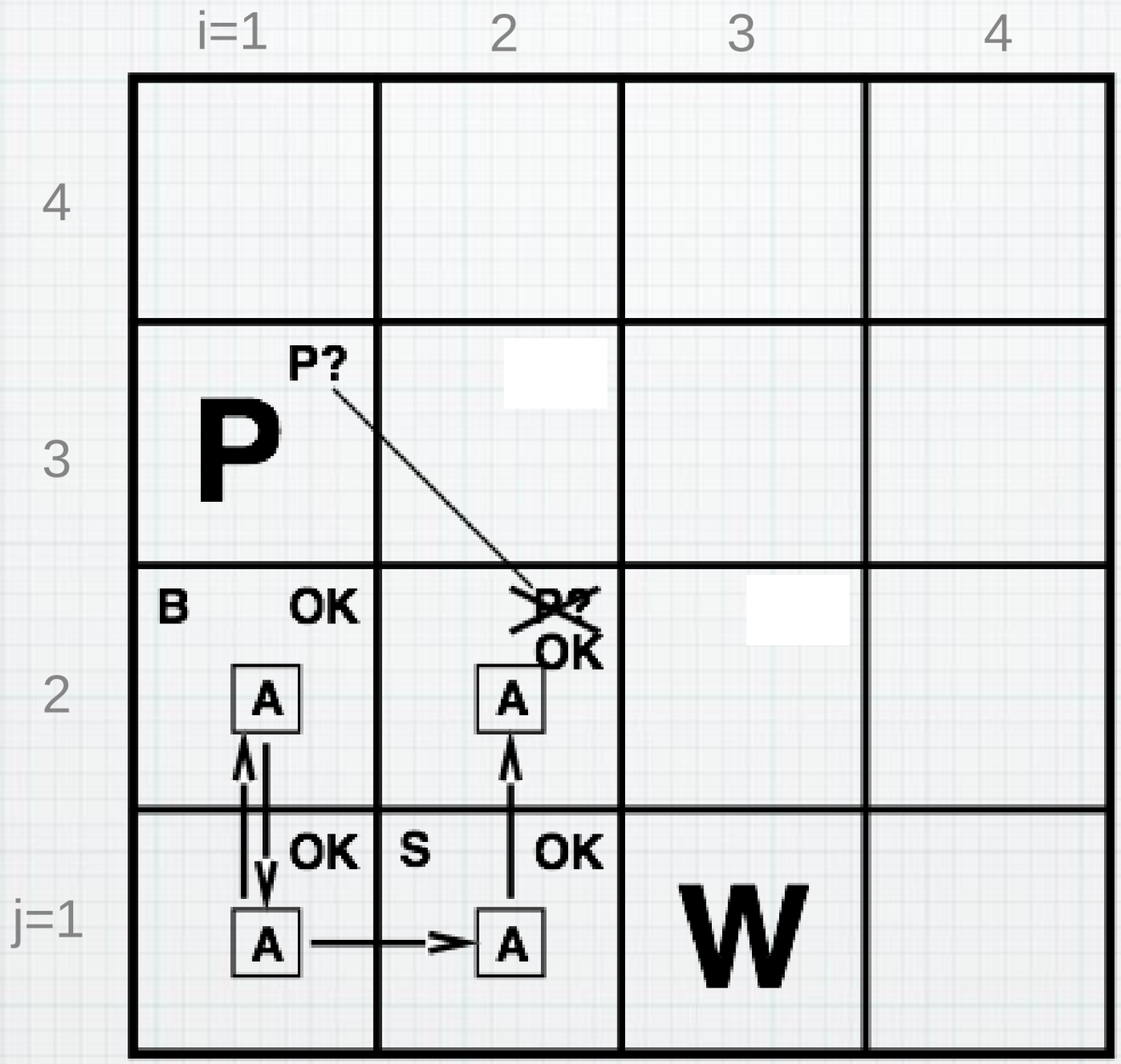
Percept [2,1]: Smelly; W in [3,1] & [2,2] is OK

Action: goto [2,2]

Exploring in Wumpus World

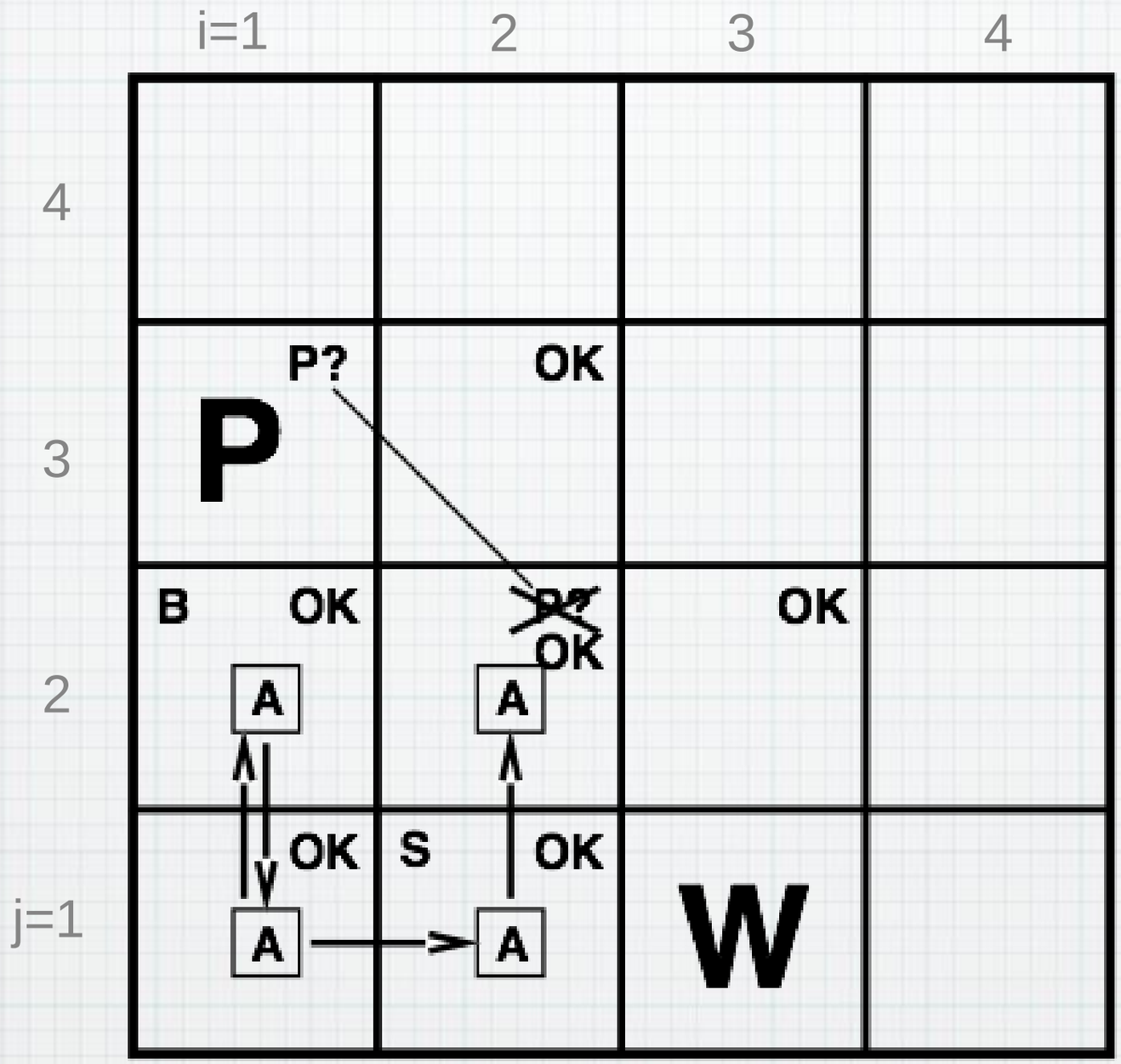


Exploring in Wumpus World



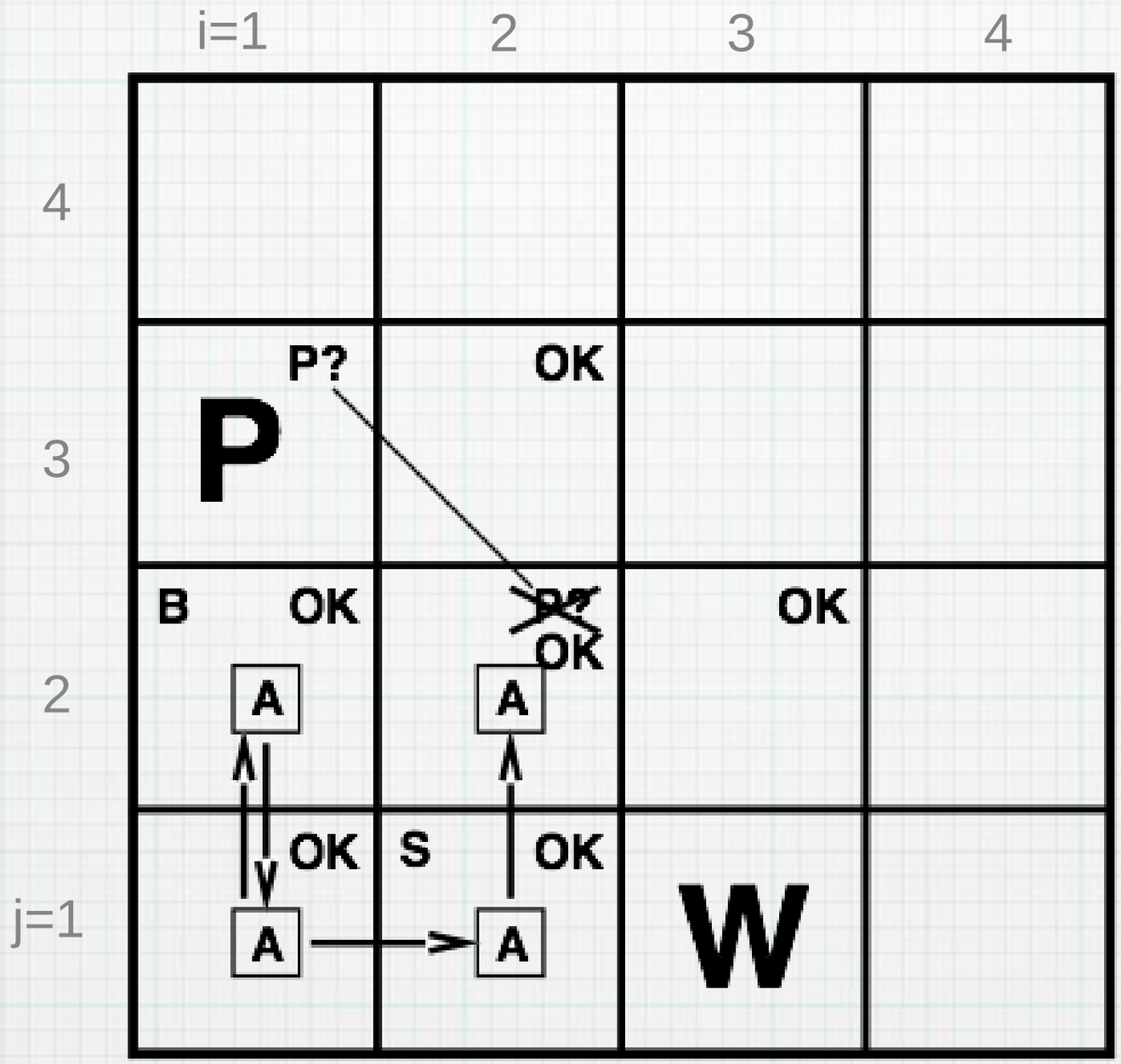
Percept [2,2]: None

Exploring in Wumpus World



Percept [2,2]: None; [2,3] & [3,2] ok

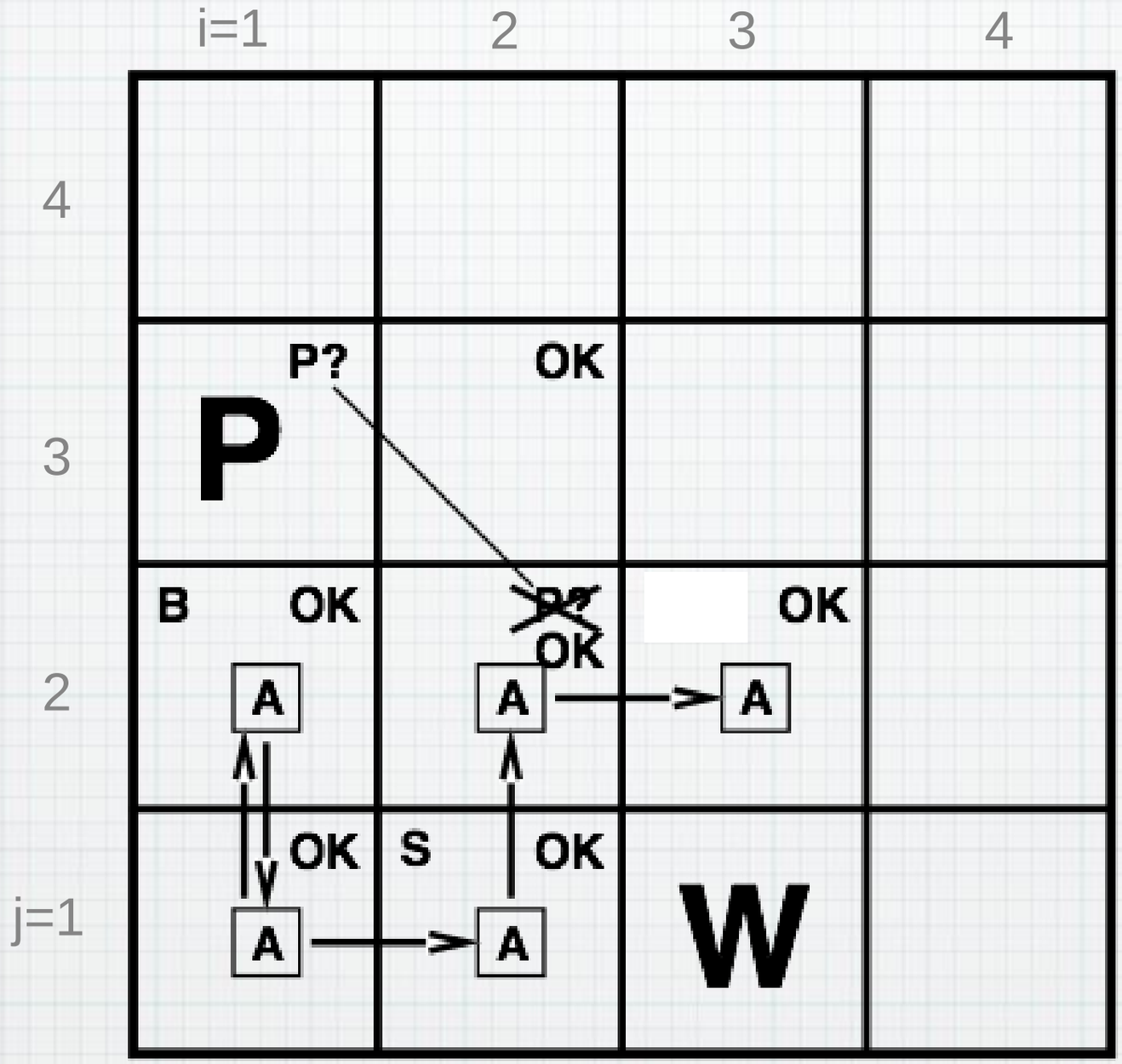
Exploring in Wumpus World



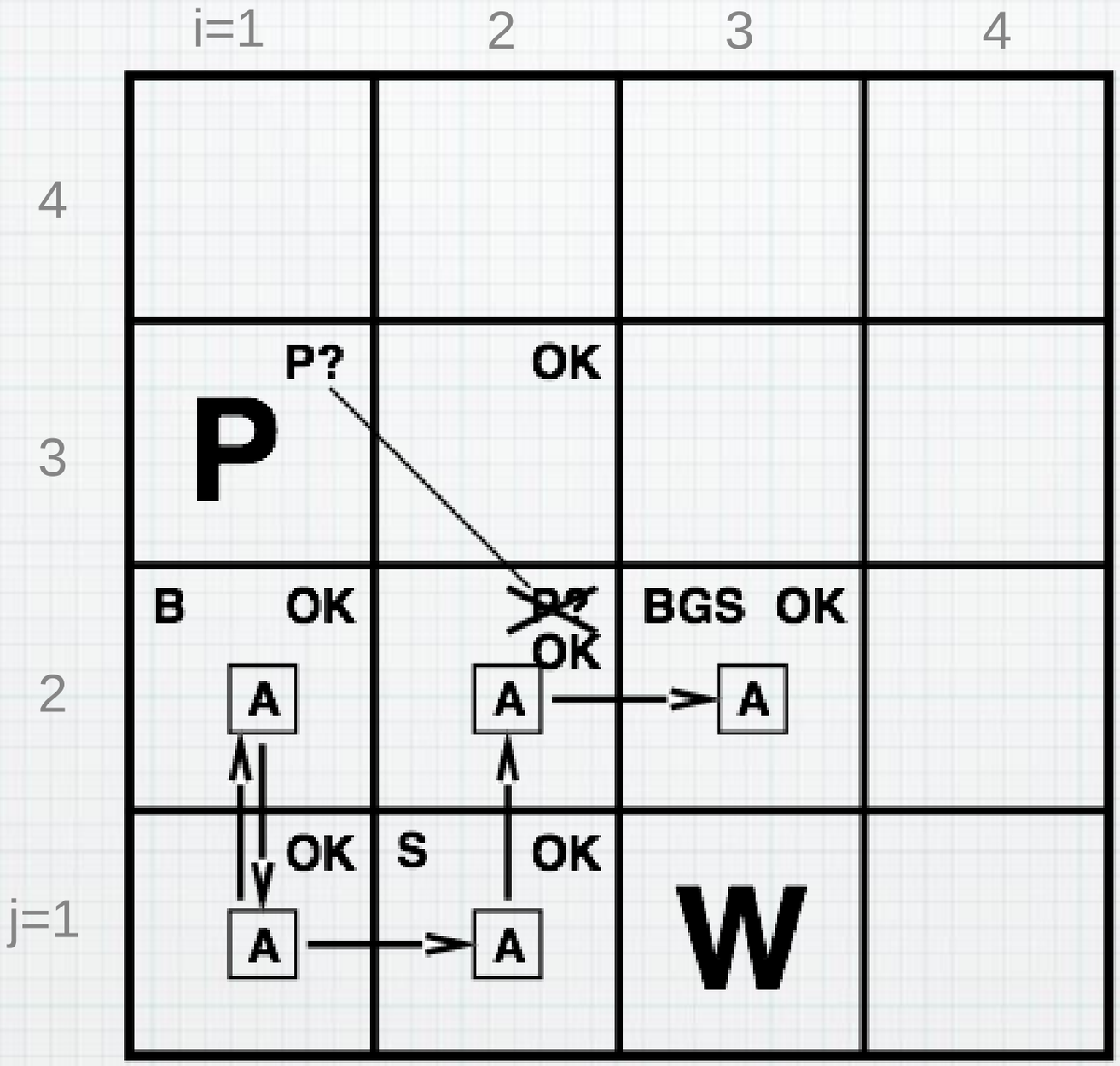
Percept [2,2]: None; [2,3] & [3,2] ok

Action: goto [3,2]

Exploring in Wumpus World

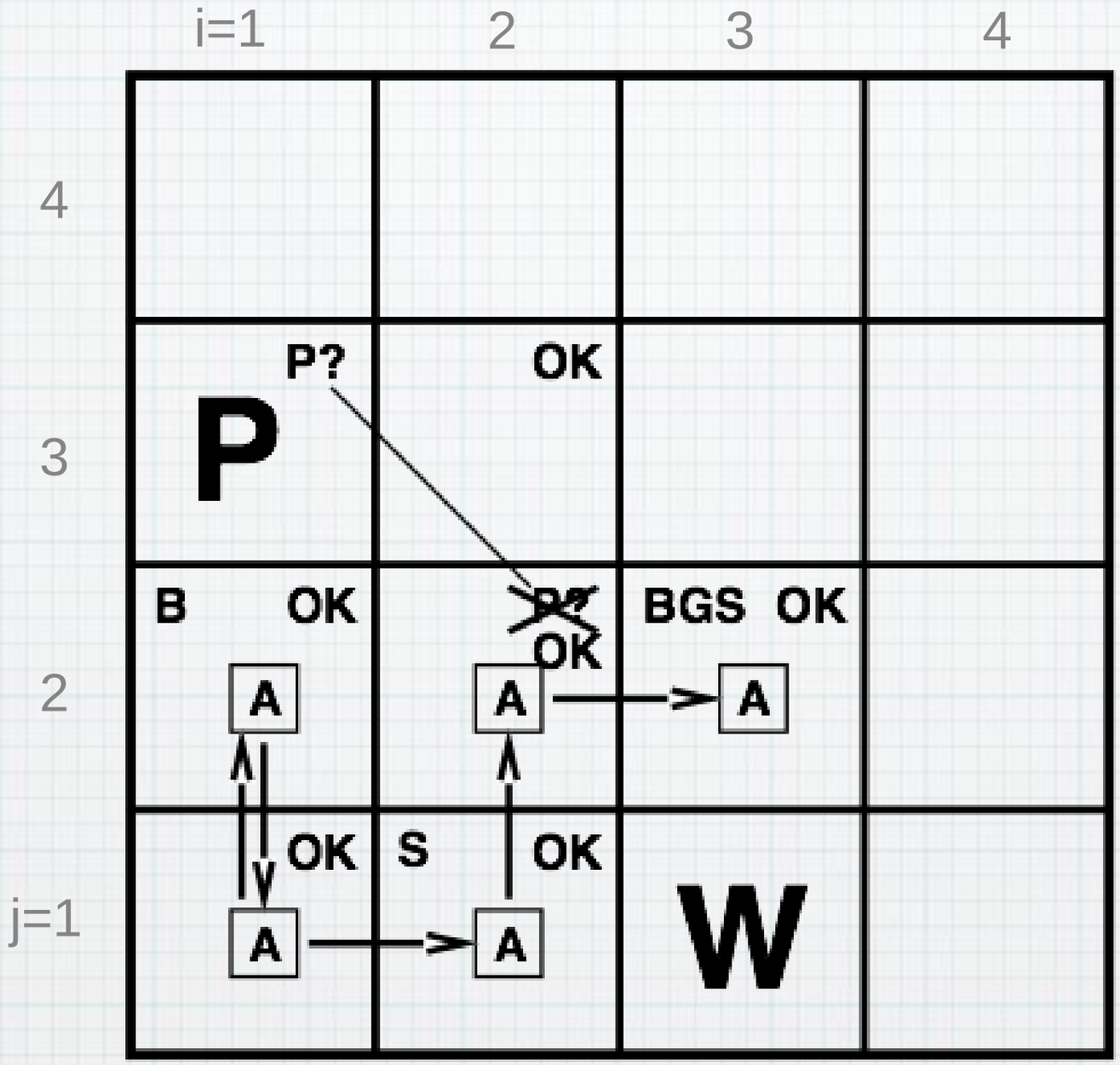


Exploring in Wumpus World



Percept [3,2]: Breeze, Glitter, Smelly

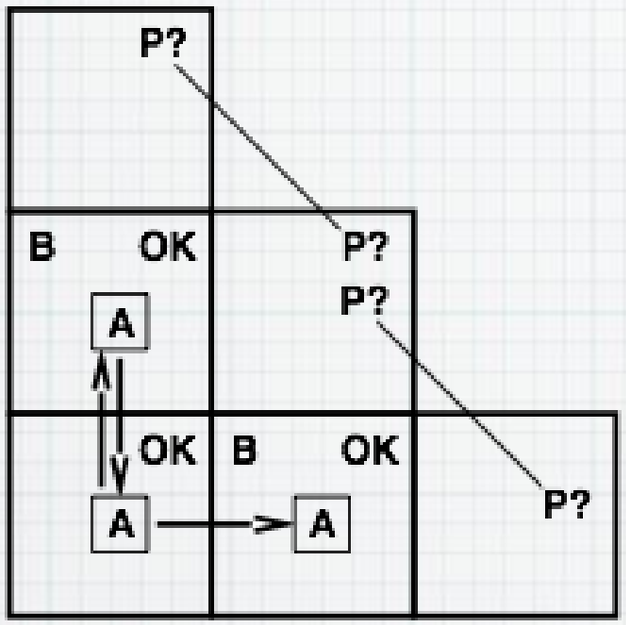
Exploring in Wumpus World



Percept [3,2]: Breeze, Glitter, Smelly

Action: grab gold!

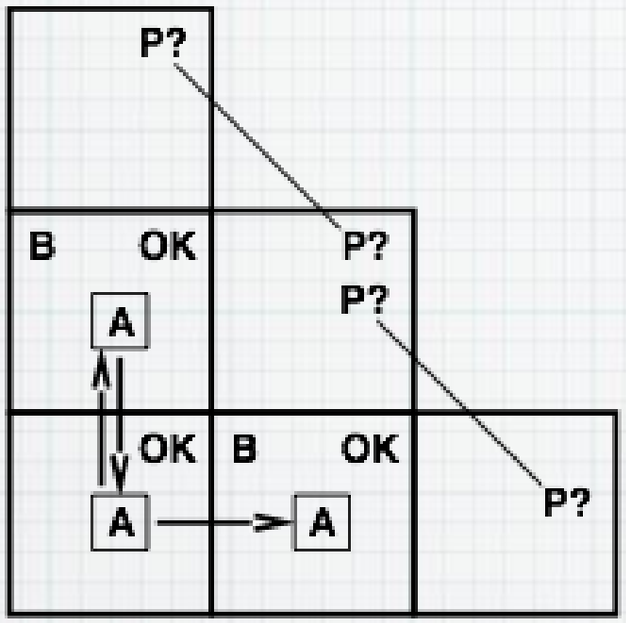
Harder Decisions



Breeze in (1,2) and (2,1)

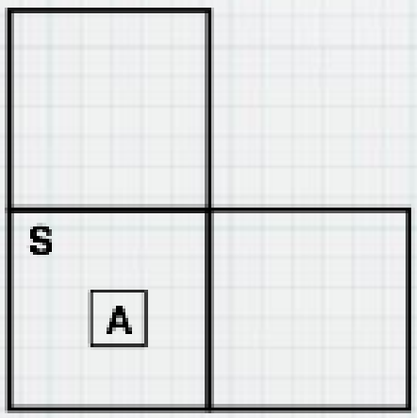
$\Rightarrow$ no safe actions

Harder Decisions



Breeze in (1,2) and (2,1)
⇒ no safe actions

Assuming pits uniformly distributed,
(2,2) has pit w/ prob 0.86, vs. 0.31



Smell in (1,1)
⇒ cannot move

Logic in General

Representation & Reasoning

Logic In General

- Logics: formal languages for representing information s.t. conclusions can be drawn
- Syntax: defines sentences in the language
- Semantics: defines meaning of sentence (defines truth of a sentence in a world)

Logic In General

E.g., the language of arithmetic

$x + 2 \geq y$ is a sentence; $x2 + y >$ is not a sentence

Logic In General

E.g., the language of arithmetic

$x + 2 \geq y$ is a sentence; $x2 + y >$ is not a sentence

$x + 2 \geq y$ is true iff the number $x + 2$ is no less than the number y

Logic In General

E.g., the language of arithmetic

$x + 2 \geq y$ is a sentence; $x2 + y >$ is not a sentence

$x + 2 \geq y$ is true iff the number $x + 2$ is no less than the number y

$x + 2 \geq y$ is true in a world where $x = 7,\ y = 1$

Logic In General

E.g., the language of arithmetic

$x+2 \geq y$ is a sentence; $x2+y >$ is not a sentence

$x+2 \geq y$ is true iff the number $x+2$ is no less than the number y

$x+2 \geq y$ is true in a world where $x=7,\ y=1$
$x+2 \geq y$ is false in a world where $x=0,\ y=6$

Logic In General

E.g., the language of arithmetic

$x + 2 \geq y$ is a sentence; $x2 + y >$ is not a sentence

$x + 2 \geq y$ is true iff the number $x + 2$ is no less than the number y

$x + 2 \geq y$ is true in a world where $x = 7,\ y = 1$
$x + 2 \geq y$ is false in a world where $x = 0,\ y = 6$

Particular instantiation of variables, a possible world, is called a model

Entailment

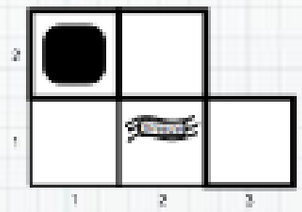
- Entailment: one thing follows from another

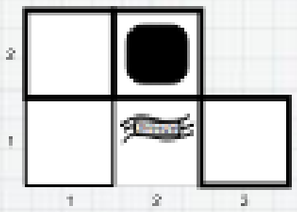
$$KB \models \alpha$$

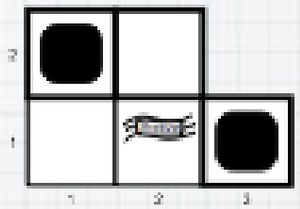
Knowledge base KB entails sentence α
if and only if
α is true in all worlds where KB is true

- Example, KB with sentence: “x+y=4”
  - entails sentence: “4=x+y”
  - and sentence: “y=4-x”

Entailment in Wumpus

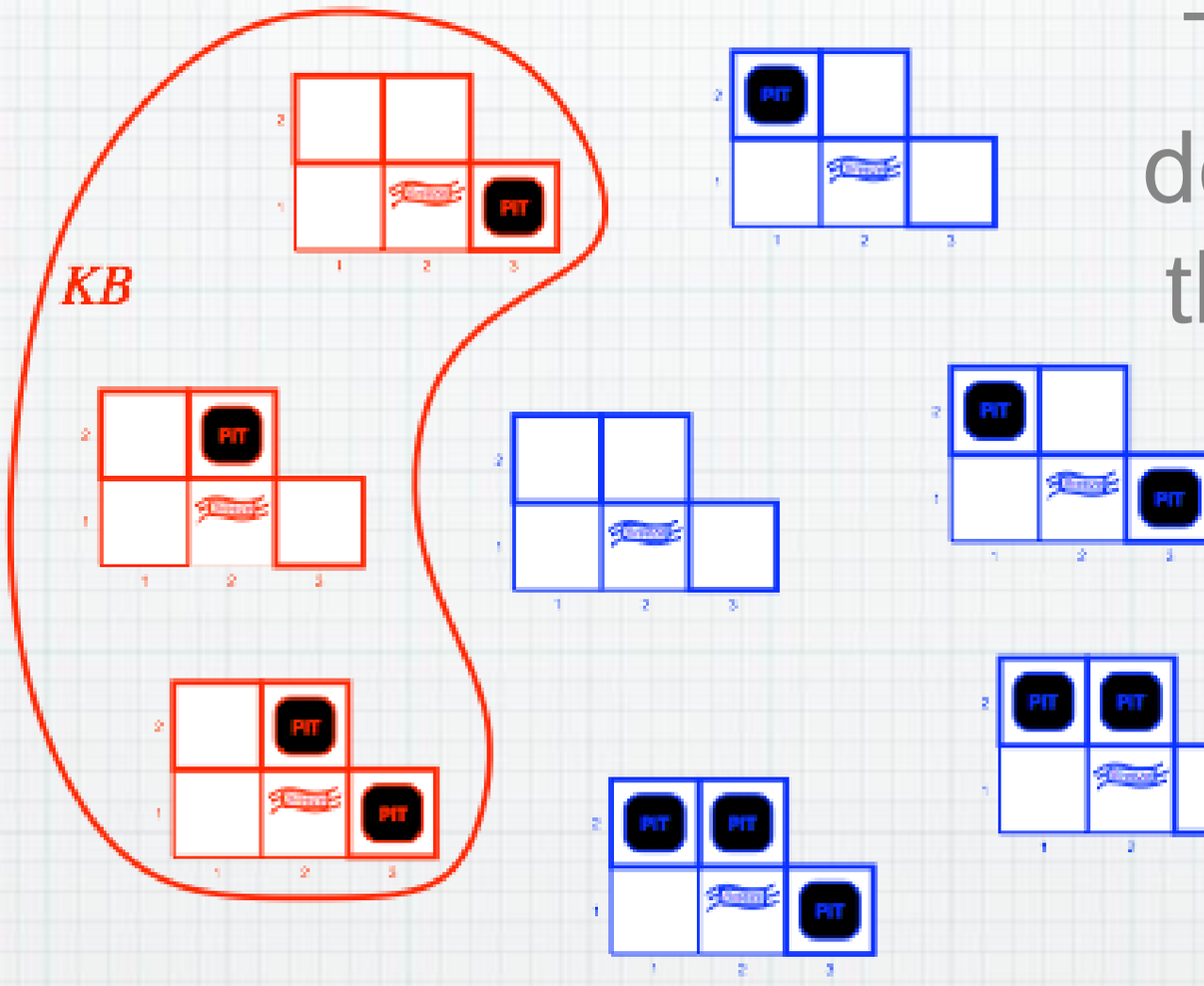






Wumpus Models

These models are consistent with the rules and our percepts so far

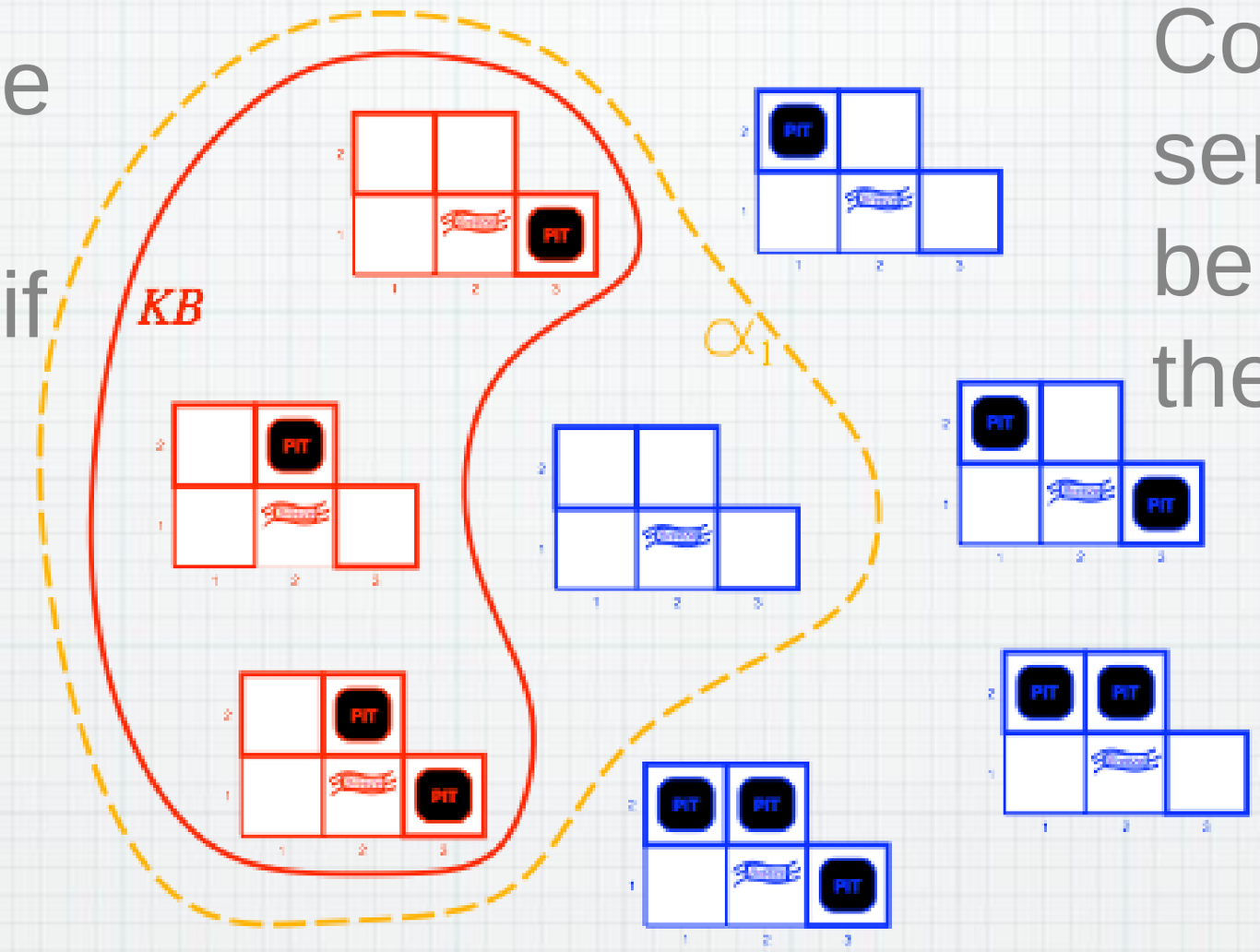


These models don’t agree with the rules about “Breezes”

KB = wumpus-world rules + observations

Model Checking

Find all models where query is true…check if KB is within this set



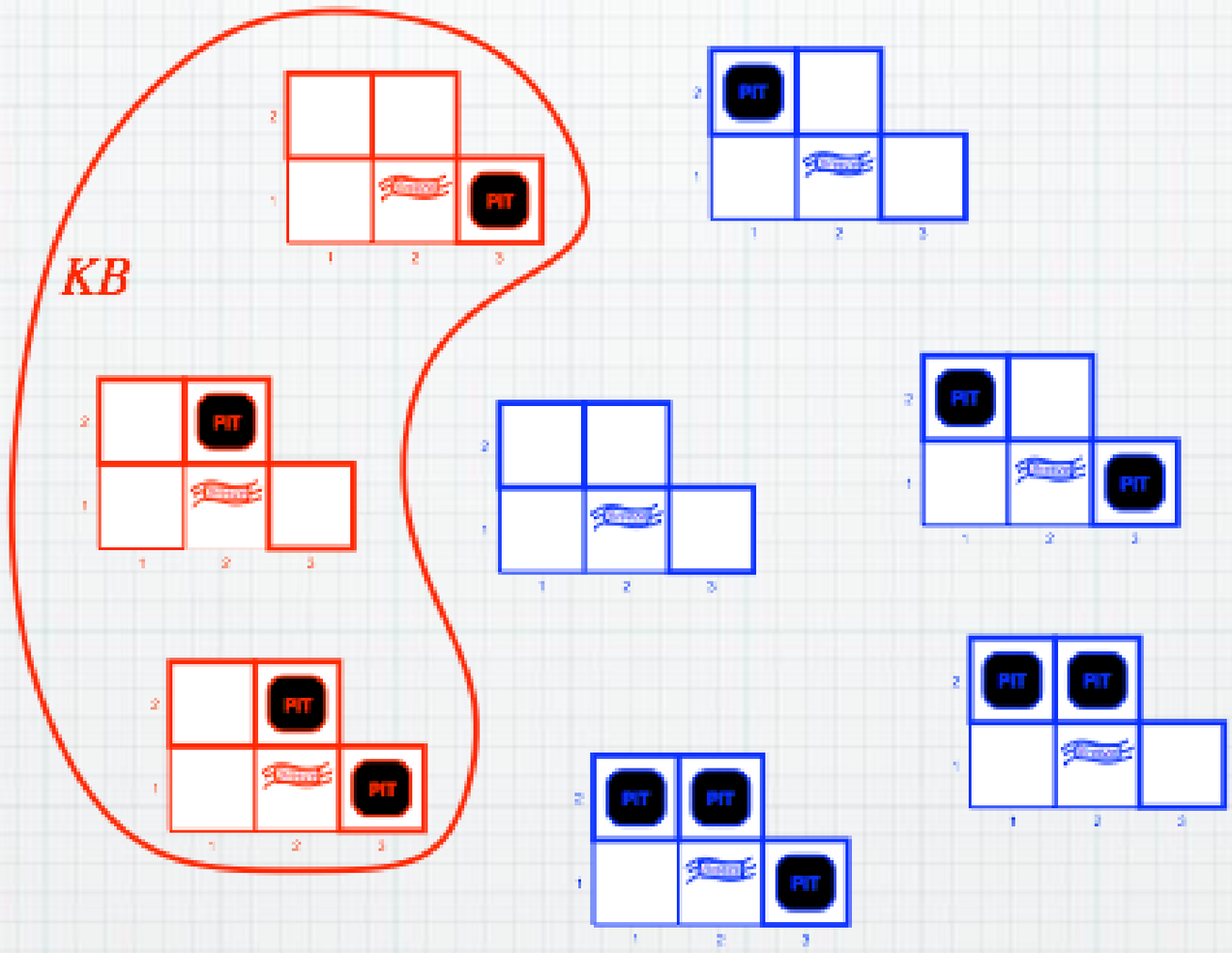
Conclude: sentence can be added to the KB

KB = wumpus-world rules + observations

α_1 = "[1,2] is safe", $KB \models \alpha_1$, proved by model checking

Model Checking

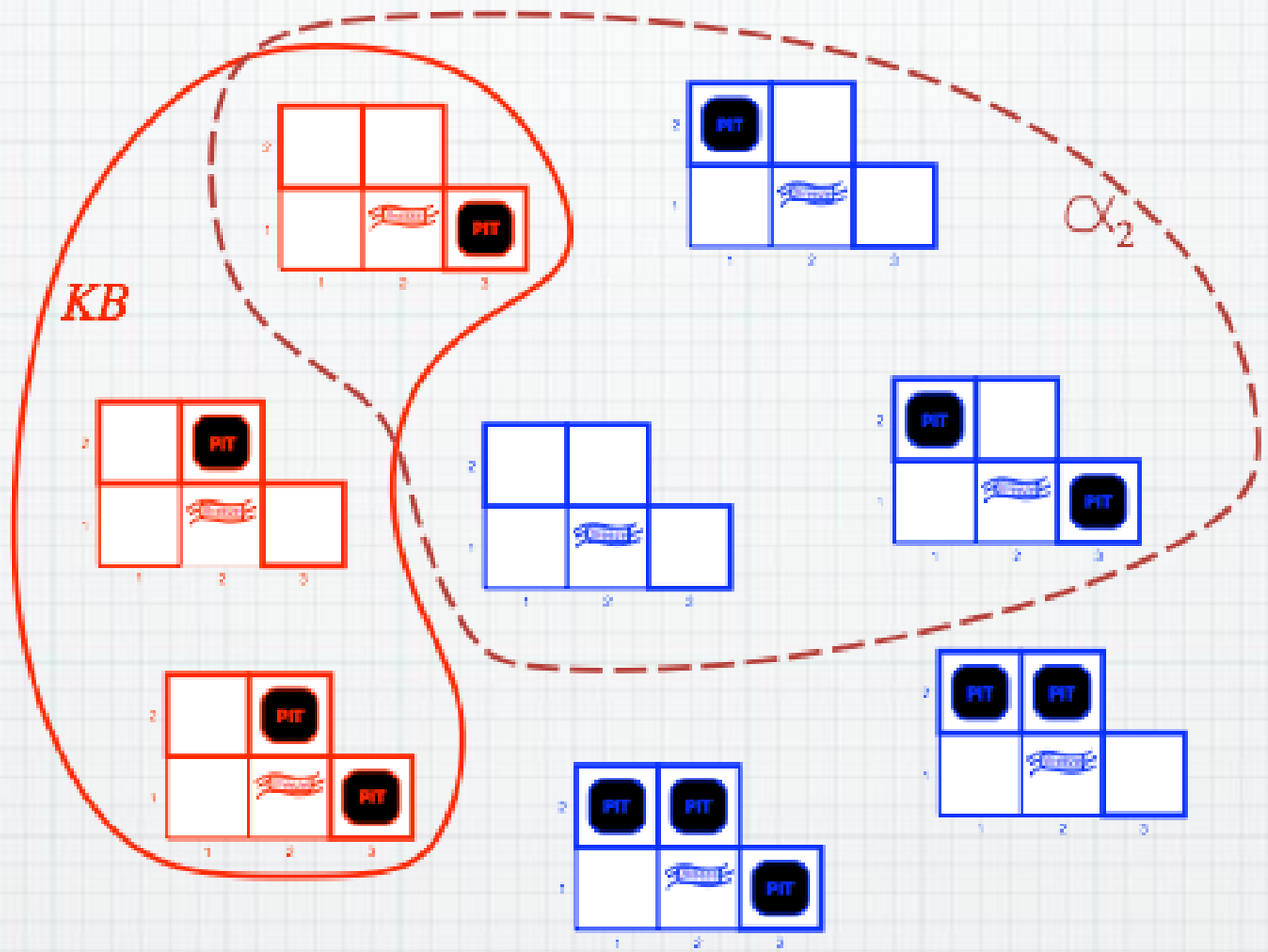
Find all models where query is true…check if KB is within this set



KB = wumpus-world rules + observations

Model Checking

Find all models where query is true…check if KB is within this set



Conclude: sentence no consistent with the KB

KB = wumpus-world rules + observations

α_2 = "[2,2] is safe", $KB \not\models \alpha_2$

Inference

$KB \vdash_i \alpha$ = sentence α can be derived from KB by procedure i

Inference

$KB \vdash_i \alpha$ = sentence α can be derived from KB by procedure i

Consequences of KB are a haystack; α is a needle.
Entailment = needle in haystack; inference = finding it

Inference

$KB \vdash_i \alpha$ = sentence α can be derived from KB by procedure i

Consequences of KB are a haystack; α is a needle.
Entailment = needle in haystack; inference = finding it

Soundness: i is sound if
whenever $KB \vdash_i \alpha$, it is also true that $KB \models \alpha$

Inference

$KB \vdash_i \alpha$ = sentence α can be derived from KB by procedure i

Consequences of KB are a haystack; α is a needle.
Entailment = needle in haystack; inference = finding it

Soundness: i is sound if
whenever $KB \vdash_i \alpha$, it is also true that $KB \models \alpha$

Completeness: i is complete if
whenever $KB \models \alpha$, it is also true that $KB \vdash_i \alpha$

Inference

$KB \vdash_i \alpha =$ sentence α can be derived from KB by procedure i

Consequences of KB are a haystack; α is a needle.
Entailment = needle in haystack; inference = finding it

Soundness: i is sound if
whenever $KB \vdash_i \alpha$, it is also true that $KB \models \alpha$

Completeness: i is complete if
whenever $KB \models \alpha$, it is also true that $KB \vdash_i \alpha$

Preview: we will define a logic (first-order logic) which is expressive enough to say almost anything of interest, and for which there exists a sound and complete inference procedure.

That is, the procedure will answer any question whose answer follows from what is known by the KB.

Questions?

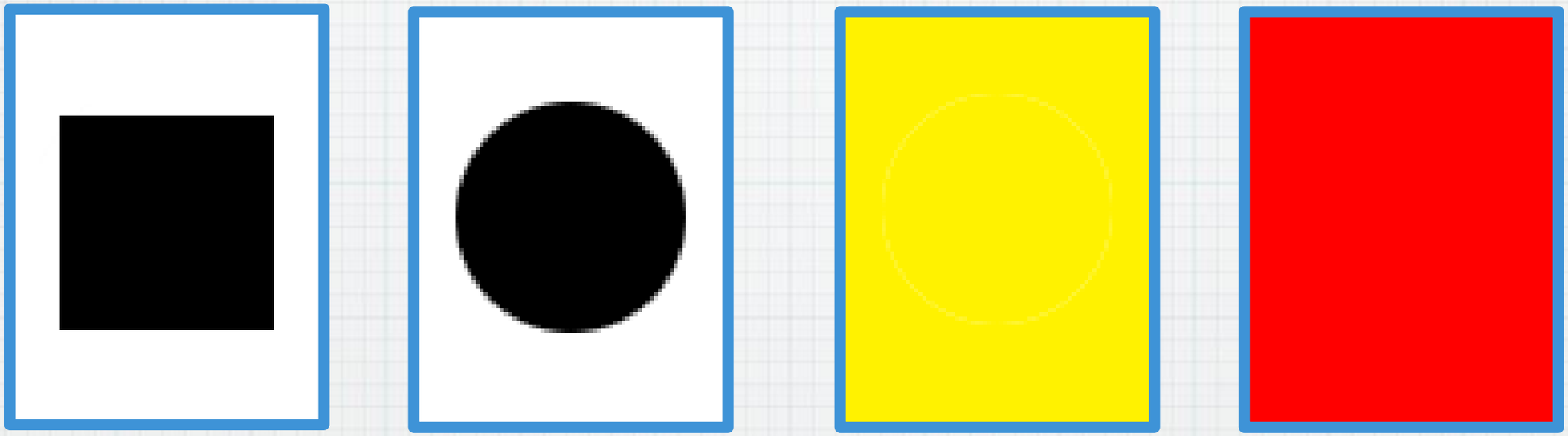
Extra: The logical agent called YOU

Extra: The logical agent called YOU

* Imagine that you are the quality control technician at a factory that prints cards that are used in signaling exercises.
* You see cards coming down the conveyor belt and you have to turn them over to check whether they follow a certain rule.

If a card has a circle on one side, then it has the color yellow on the other side.

If a card has a circle on one side, then it has the color yellow on the other side.



* You got laid off. But, you managed to get a new job at a local bar as the bouncer.
* You see people, and you see drinks, and your job is to make sure that everyone is following the law.

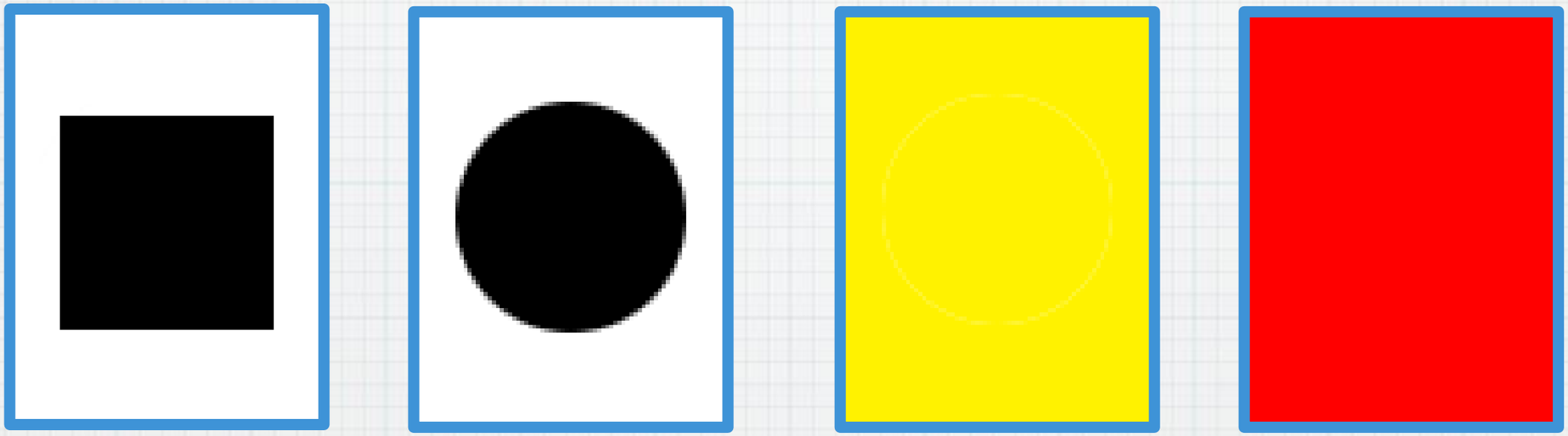
If a person drinks an alcoholic drink, then they must be over the age of 21 years old.

If a person drinks an alcoholic drink, then they must be over the age of 21 years old.

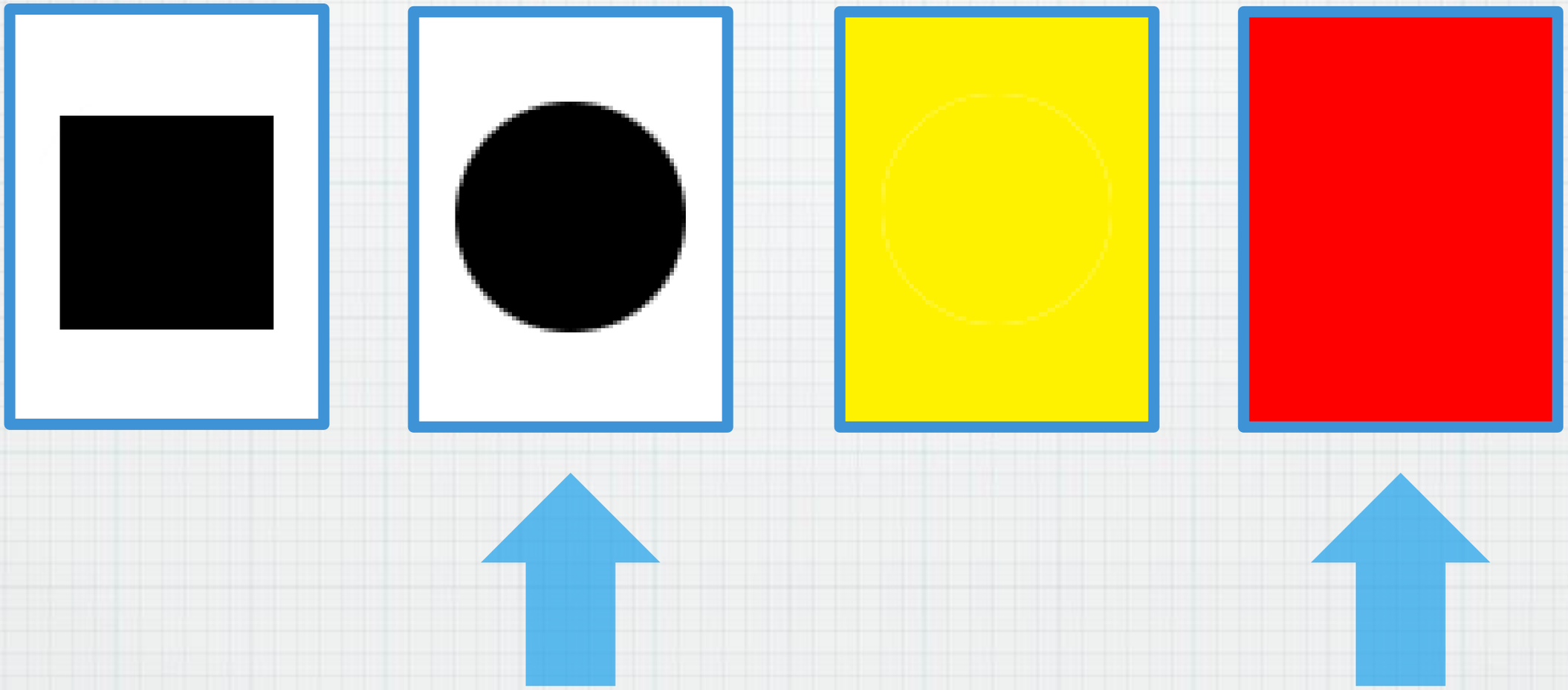
22 years old

17 years old

If a card has a circle on one side, then it has the color yellow on the other side.



If a card has a circle on one side, then it has the color yellow on the other side.



If a person drinks an alcoholic drink, then they must be over the age of 21 years old.

22 years old

17 years old

If a person drinks an alcoholic drink, then they must be over the age of 21 years old.

22 years old

17 years old

Why?

Why?

* This task is called the **Wason Selection Task.**