

CS 380: Artificial Intelligence

Lecture 8: Logical Agents

Summary so far:

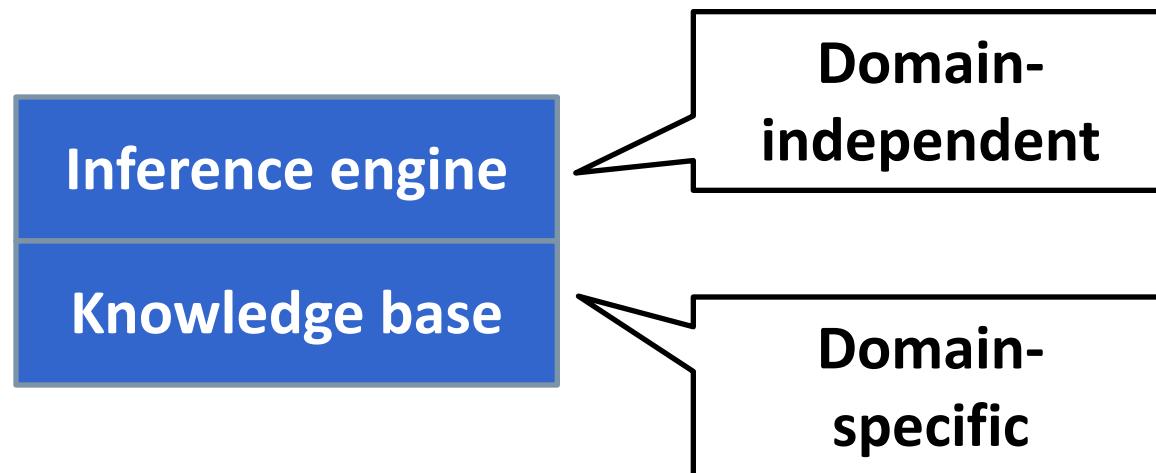
- What is AI?
 - Rational Agents
- Problem solving:
 - Systematic search
 - Uninformed search: DFS, BFS, ID
 - Informed search: heuristics, Greedy Search, A*
 - Local Search: Hill Climbing, Simulated Annealing
 - Adversarial Search: Minimax, alpha-beta, Monte Carlo Tree Search
- **Next: Knowledge Representation + Reasoning**

Knowledge

- Knowledge base:
 - When you think of “knowledge”, what do you think of?

Knowledge

- Knowledge base:
 - For our purposes: A set of **sentences** in a **formal language**
- Knowledge-based agent:



- “Declarative programming”

Declarative Programming

- Tell an agent **WHAT** to do, but not **HOW**.
- Encode all the necessary knowledge, and let the agent *infer* how to achieve the goals.
- For example, instead of programming Quicksort or any other sorting algorithm, we could tell an agent:
 - Here's a set of numbers.
 - Now I want them sorted, which means that for every adjacent numbers a and b , they must satisfy $a \leq b$
 - You (the agent) figure out how ☺
- This is really hard!!

Wumpus World

- There's a cave with a terrible "Wumpus" that eats anyone who enters its room 😞
- The Wumpus can be shot with an arrow if you're facing the right direction 😊
- ... but you only have 1 arrow 😞
- Some rooms have a bottomless pit that traps you 😞
- So why are you poking around Wumpus World?
There's gold in one room!! 😊 😊 😊

Wumpus World PEAS description

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

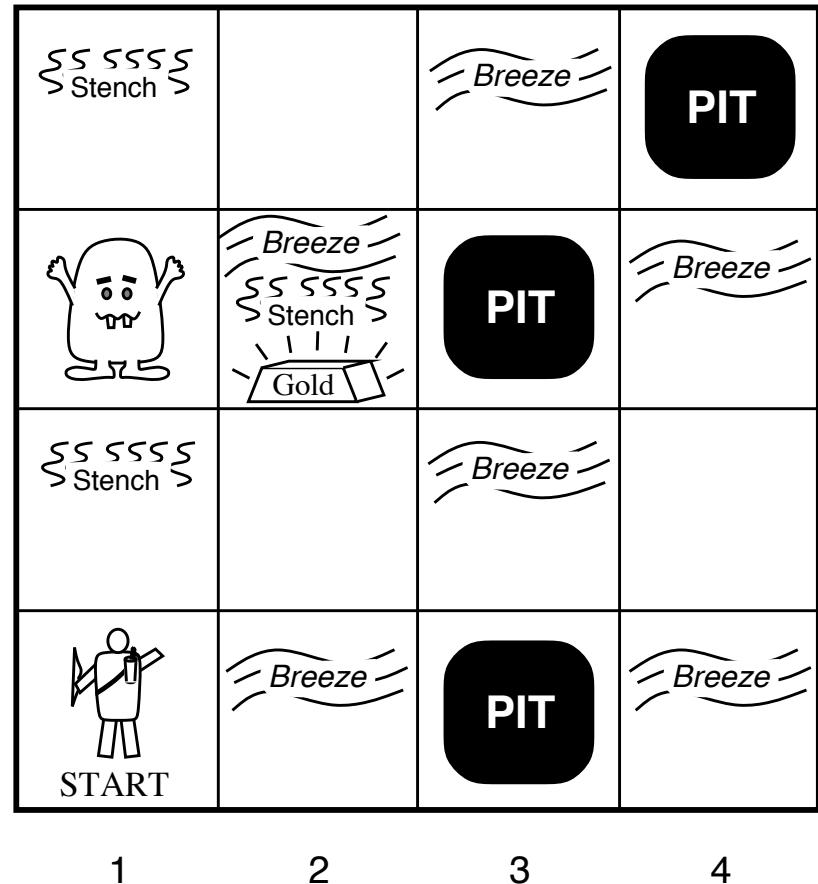
Grabbing picks up gold if in same square

Releasing drops the gold in same square

Actuators Left turn, Right turn,

Forward, Grab, Release, Shoot

Sensors Breeze, Glitter, Smell



Wumpus world characterization

Observable??

Wumpus world characterization

Observable?? No—only **local** perception

Deterministic??

Wumpus world characterization

Observable?? No—only local perception

Deterministic?? Yes—outcomes exactly specified

Episodic??

Wumpus world characterization

Observable?? No—only local perception

Deterministic?? Yes—outcomes exactly specified

Episodic?? No—sequential at the level of actions

Static??

Wumpus world characterization

Observable?? No—only local perception

Deterministic?? Yes—outcomes exactly specified

Episodic?? No—sequential at the level of actions

Static?? Yes—Wumpus and Pits do not move

Discrete??

Wumpus world characterization

Observable?? No—only local perception

Deterministic?? Yes—outcomes exactly specified

Episodic?? No—sequential at the level of actions

Static?? Yes—Wumpus and Pits do not move

Discrete?? Yes

Single-agent??

Wumpus world characterization

Observable?? No—only local perception

Deterministic?? Yes—outcomes exactly specified

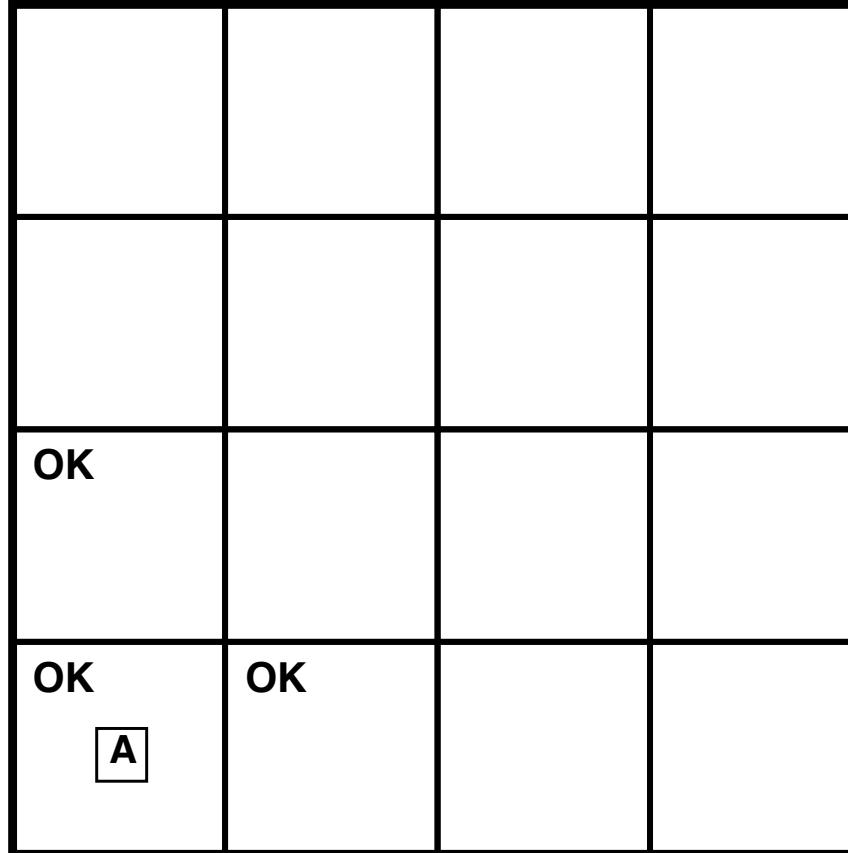
Episodic?? No—sequential at the level of actions

Static?? Yes—Wumpus and Pits do not move

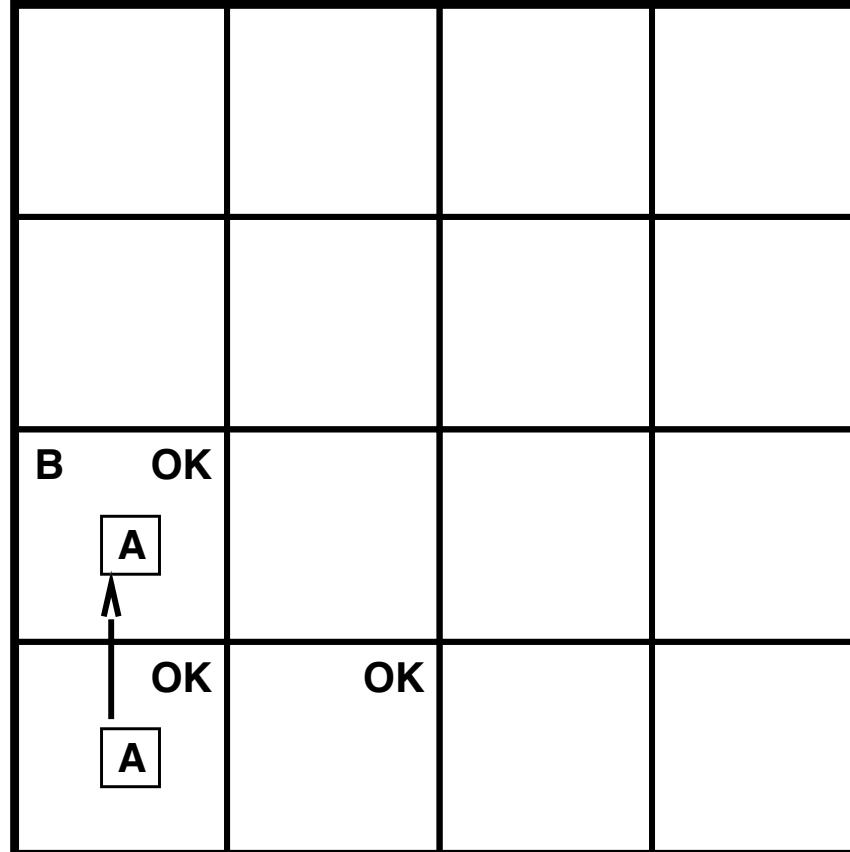
Discrete?? Yes

Single-agent?? Yes—Wumpus is essentially a natural feature

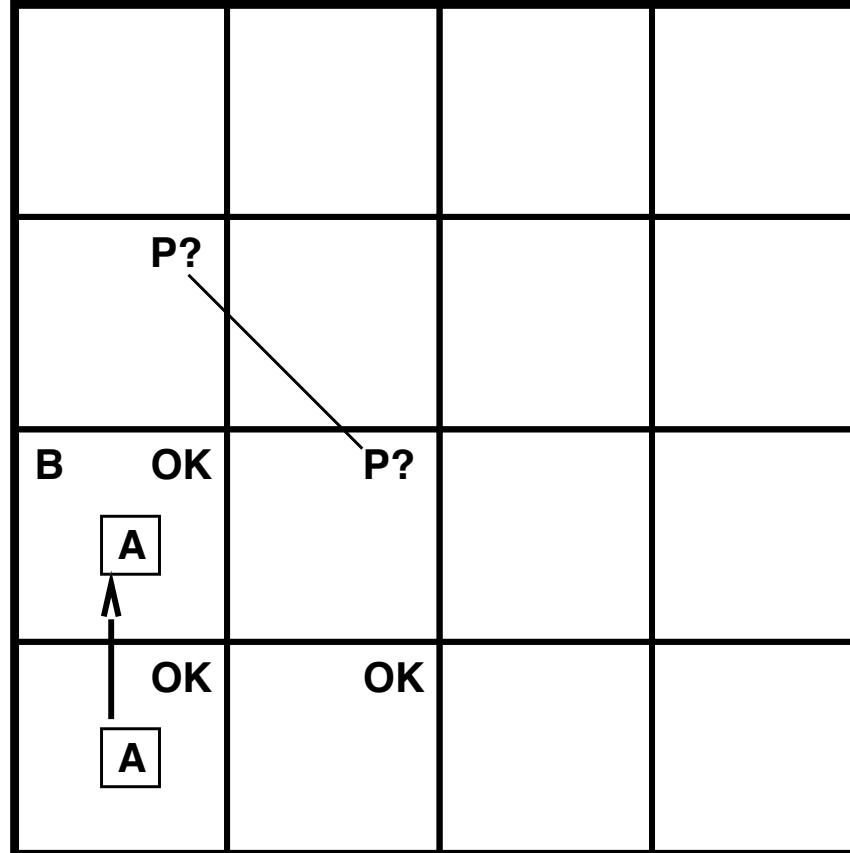
Exploring a wumpus world



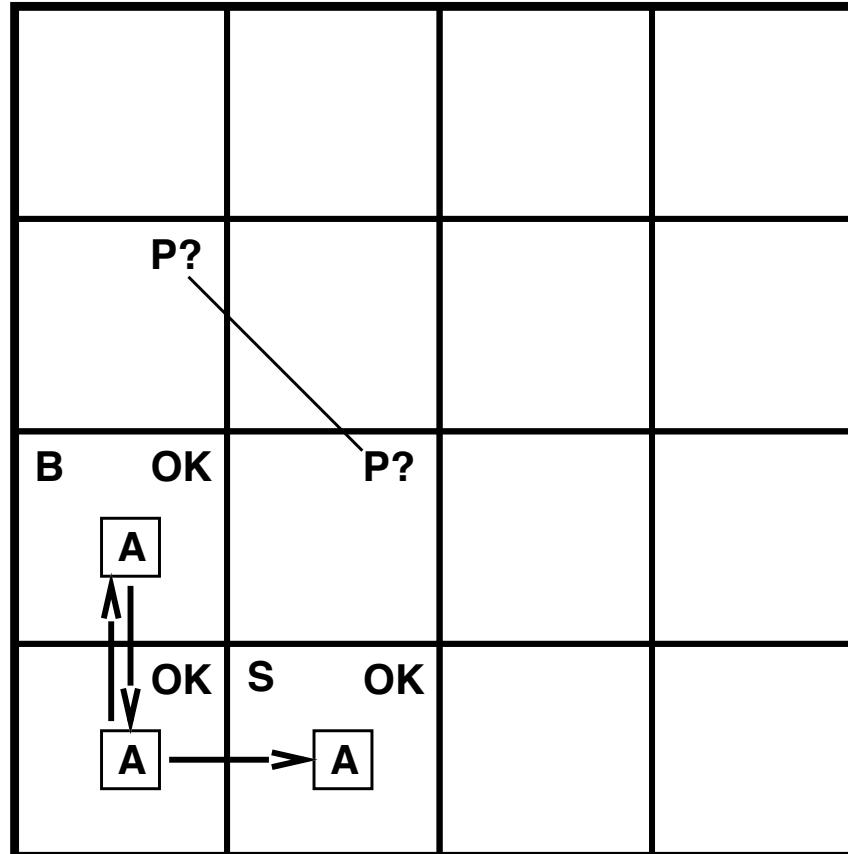
Exploring a wumpus world



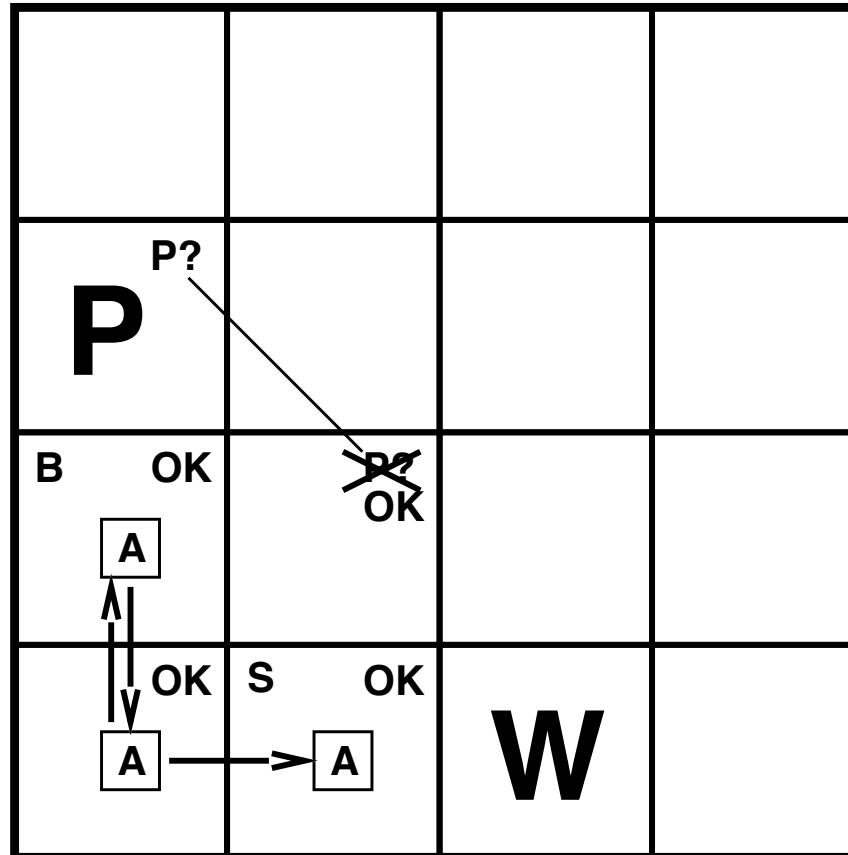
Exploring a wumpus world



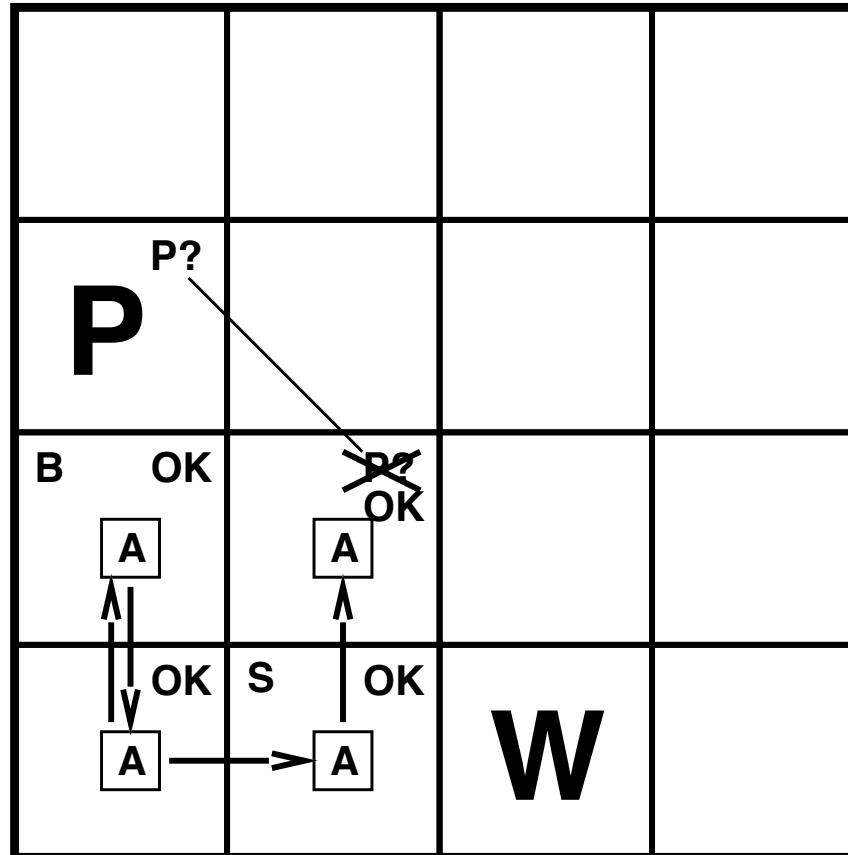
Exploring a wumpus world



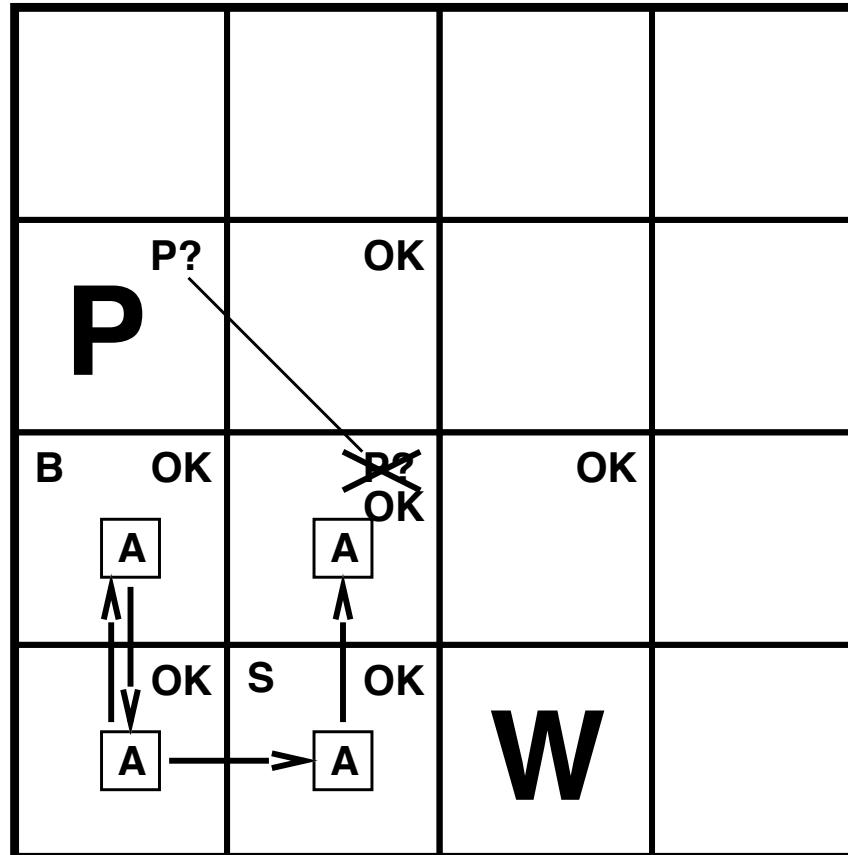
Exploring a wumpus world



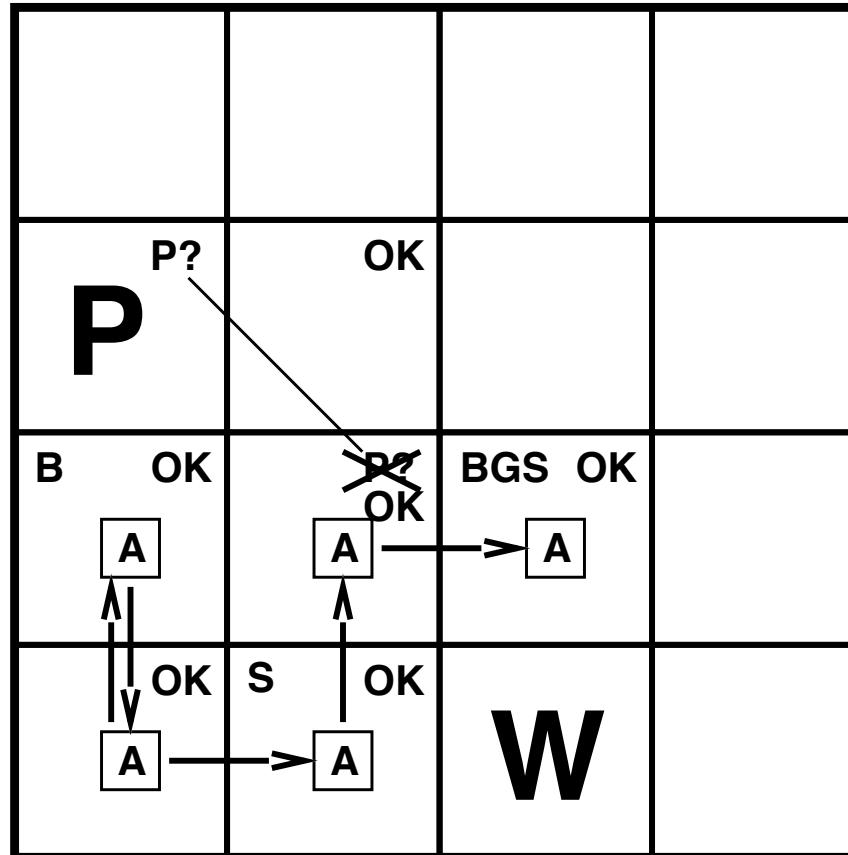
Exploring a wumpus world



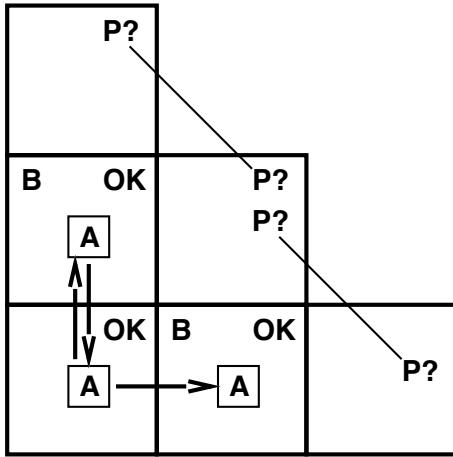
Exploring a wumpus world



Exploring a wumpus world



Other tight spots



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

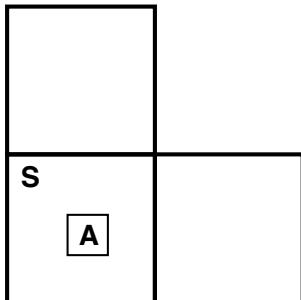
Smell in (1,1)
⇒ cannot move

Can use a strategy of **coercion**:

shoot straight ahead

wumpus was there ⇒ dead ⇒ safe

wumpus wasn't there ⇒ safe



Logic in general

Logics are formal languages for representing information such that conclusions can be drawn

Syntax defines the sentences in the language

Semantics define the “meaning” of sentences;
i.e., define truth of a sentence in a world

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 $x = 0, y = 6$

Entailment

Entailment means that 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

E.g., the KB containing “the Giants won” and “the Reds won”
entails “Either the Giants won or the Reds won”

E.g., $x + y = 4$ entails $4 = x + y$

Entailment is a relationship between sentences (i.e., **syntax**)
that is based on **semantics**

Note: brains process **syntax** (of some sort)

Models

Logicians typically think in terms of **models**, which are formally structured worlds with respect to which truth can be evaluated

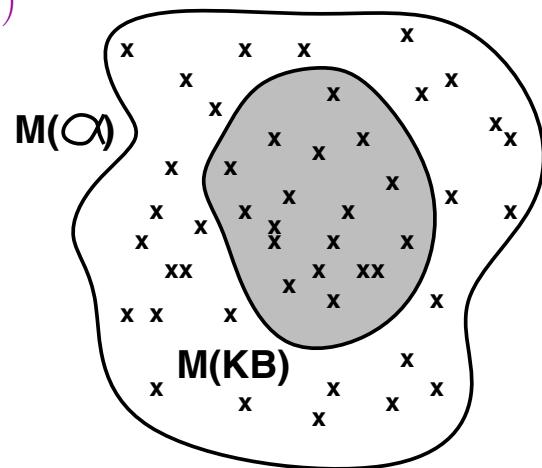
We say m is a model of a sentence α if α is true in m

$M(\alpha)$ is the set of all models of α

Then $KB \models \alpha$ if and only if $M(KB) \subseteq M(\alpha)$

E.g. $KB =$ Giants won and Reds won

$\alpha =$ Giants won

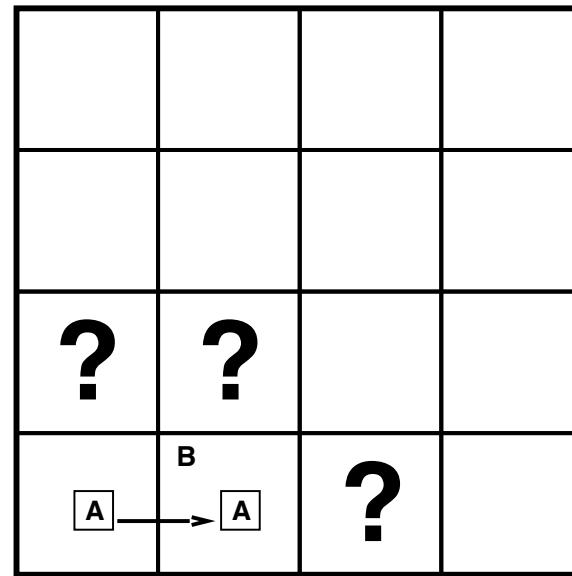


Entailment in the wumpus world

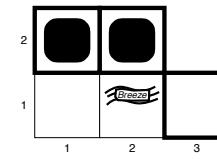
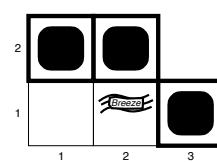
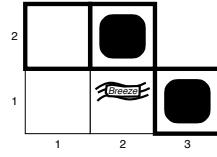
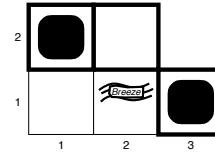
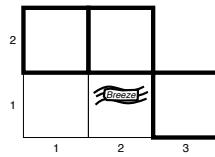
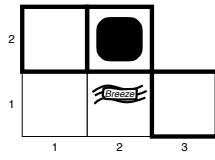
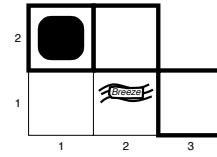
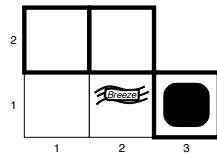
Situation after detecting nothing in [1,1],
moving right, breeze in [2,1]

Consider possible models for ?s
assuming only pits

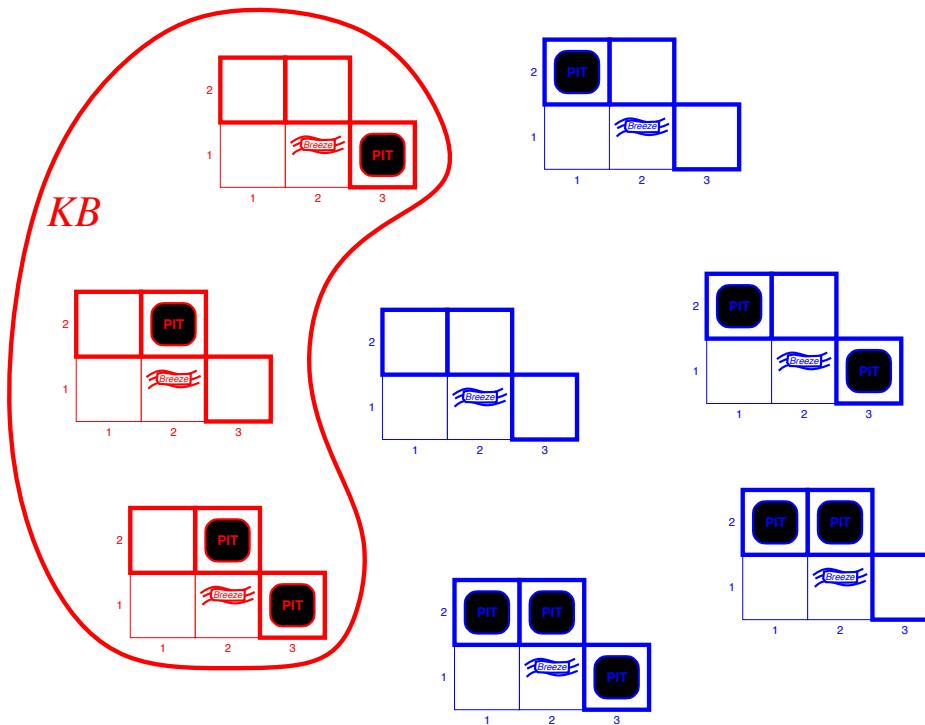
3 Boolean choices \Rightarrow 8 possible models



Wumpus models

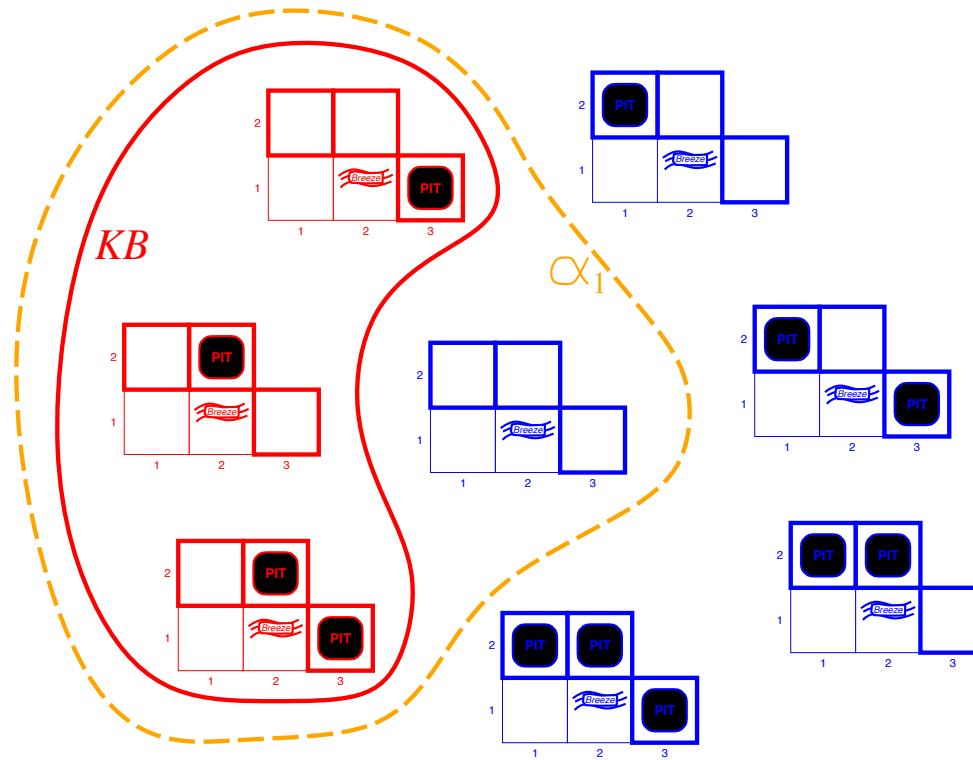


Wumpus models



KB = wumpus-world rules + observations

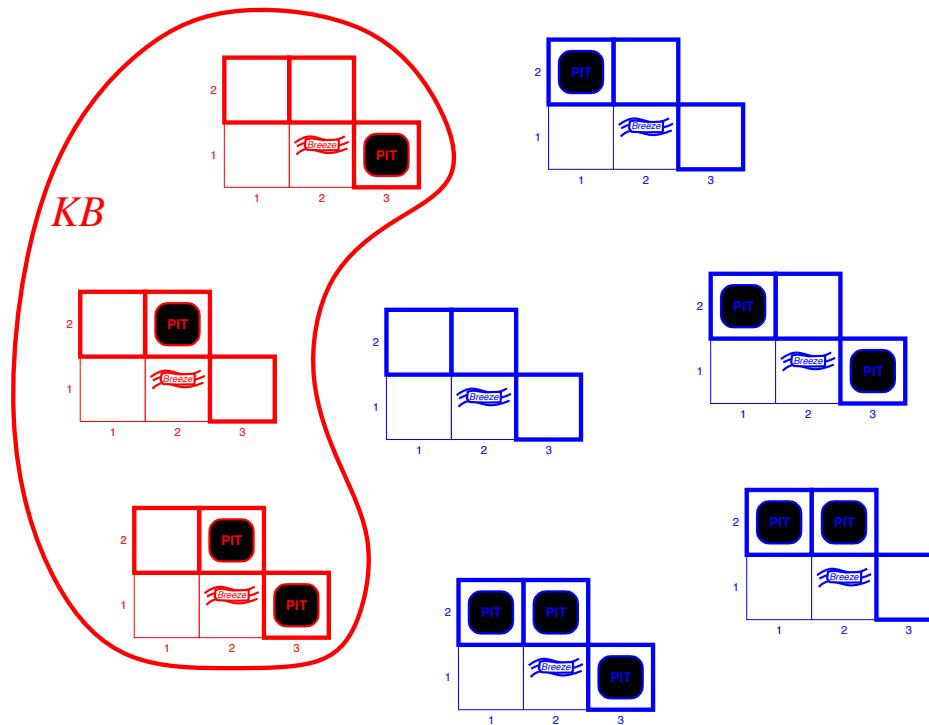
Wumpus models



$KB = \text{wumpus-world rules} + \text{observations}$

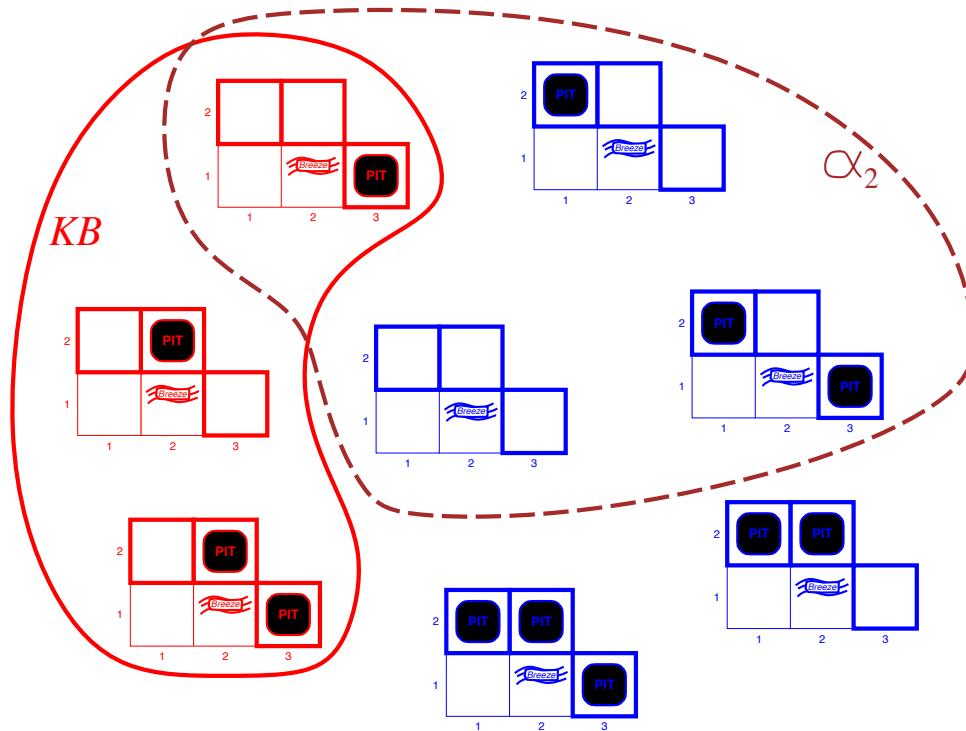
$\alpha_1 = "[1,2] \text{ is safe}" , KB \models \alpha_1$, proved by model checking

Wumpus models



KB = wumpus-world rules + observations

Wumpus models



KB = wumpus-world rules + observations

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

Transitioning...

- We'll continue with Wumpus World in the next lecture
- But now, let's take a look at LISP!