Logic-Based Agents[1]

Lecture 1

[1]The slides have been prepared using the textbook material available on the web, and the slides of the previous editions of the course by Prof. Luigia Carlucci Aiello

Summary

- Logical Agents   Russell & Norvig Sect. 7.1–7.4
- Knowledge Representation
- Knowledge-based Agents
- The wumpus world
- Propositional Logic

Symbol hypothesis

The conputer belongs to a class of artificial systems, **symbolic systems**...

Symbols can be used to represent **structures**. . . .

A computer is a symbolic system, where symbols are **created, modified, copied,**. . .

not a quote, but this is argued in:
[H. Simon, The Sciences of the Artificial, 1969, 1994]

Properties of a good representation

A **representation** is made by a set of symbols, stored on a computer together with algorithms that allow to us to use it in order to solve problems.

♢ **Adequacy** criteria by McCarthy & Hayes in 1969: **meta-physics, epistemology, heuristics**

♢ language expressive power (ontology)

♢ level of abstraction

♢ good design (computation)

Representation language

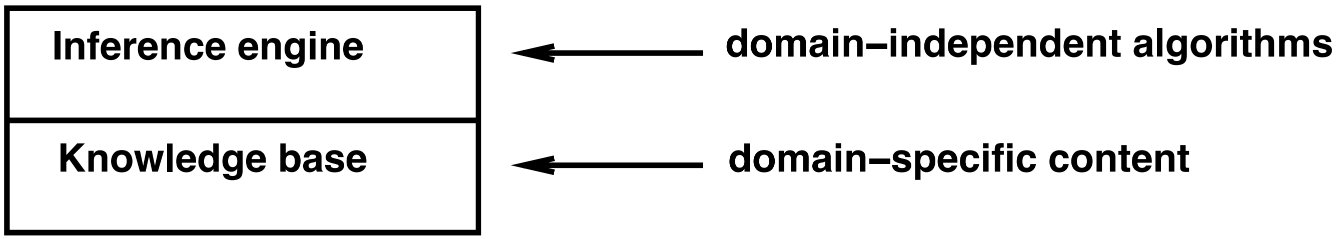
- not a programming language
- not a communication language

Propositional Logic is the basic example:

Key principle: syntax – semantics – reasoning

Possibly "visual", but semantically well defined

Knowledge bases



In a knowledge representation language the set of inference is defined without referring to the algorithm to compute them.

Knowledge bases

Knowledge base = set of sentences in a formal language

Declarative approach to building an agent (or other system):
TELL it what it needs to know

Then ASK it what to do—answers that follow from the KB

Agents can be viewed at the knowledge level
i.e., what they know, regardless of how implemented

Or at the implementation level
i.e., data structures in KB and algorithms that manipulate them

An example

knowledge base	inference
Rain Wind if Rain then TakeUmbrella if Wind then WindJacket if Rain and WithoutUmbrella then Wet	if A is true and $A \Rightarrow B$ then B can be derived

From the knowledge base:

Rain??

Wind??

TakeUmbrella??

WindJacket??

Fog??

Wet??

A simple knowledge-based 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*))
  $t \leftarrow t + 1$
  return *action*

A simple knowledge-based agent

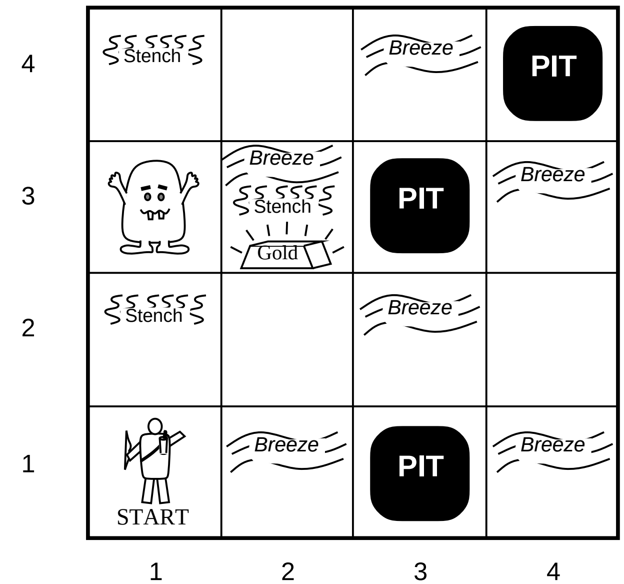
The agent must be able to:

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

Wumpus World PEAS description

Environment

Squares adjacent to wumpus are smelly
Squares adjacent to pits 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



Wumpus World PEAS description

Performance measure

gold +1000, death -1000

-1 per step, -10 for using the arrow

Sensors

Stench, Breeze, Glitter, Bump, Cry

Actuators

Left turn, Right turn, Forward,

Grab, Release, Shoot

Wumpus world characterization

Observable?? No—only local perception

Deterministic?? Yes—outcomes exactly specified

Episodic?? No—sequential at the level of actions

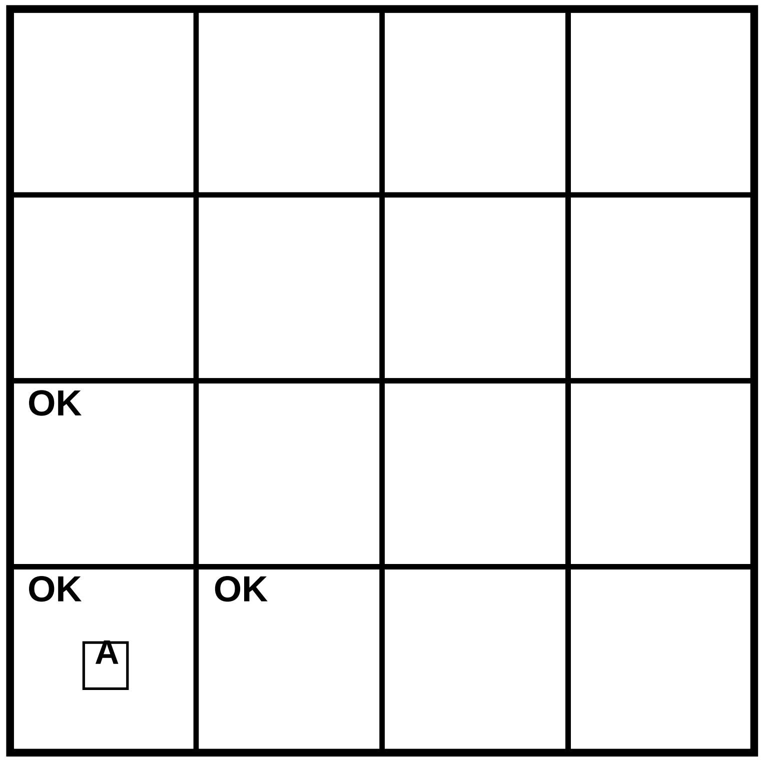
Static?? Yes—Wumpus and Pits do not move

Discrete?? Yes

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

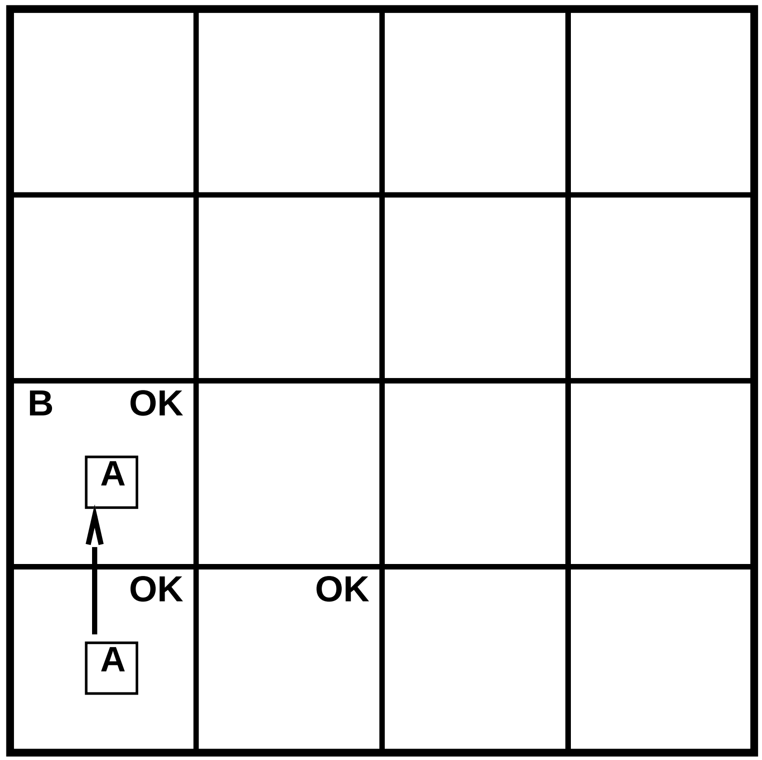
Exploring a wumpus world

Sensors: [None,None,None,None,None]



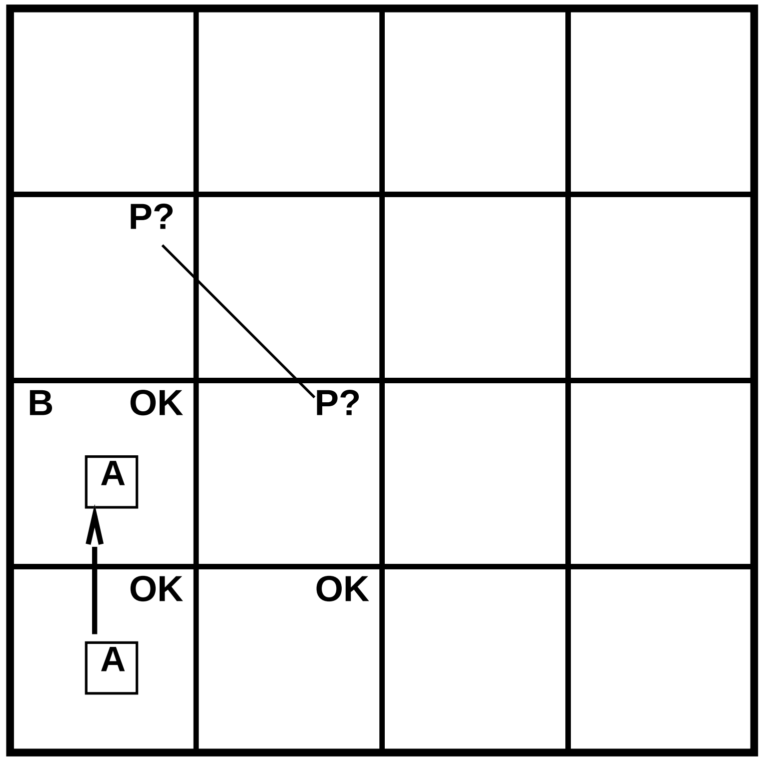
Exploring a wumpus world

Sensors: [None,Breeze,None,None,None]



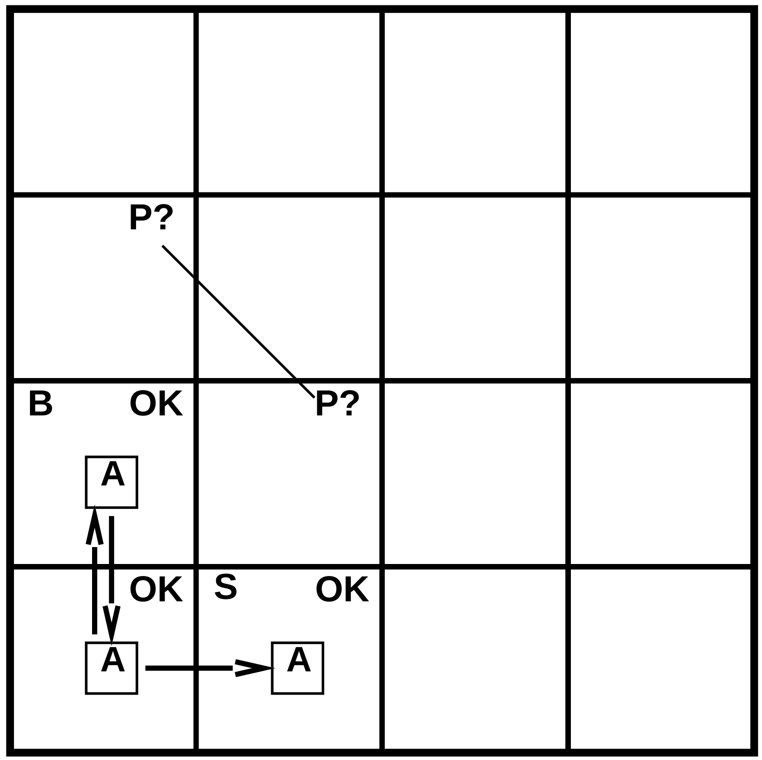
Exploring a wumpus world

Squares adjacent to pits are breezy



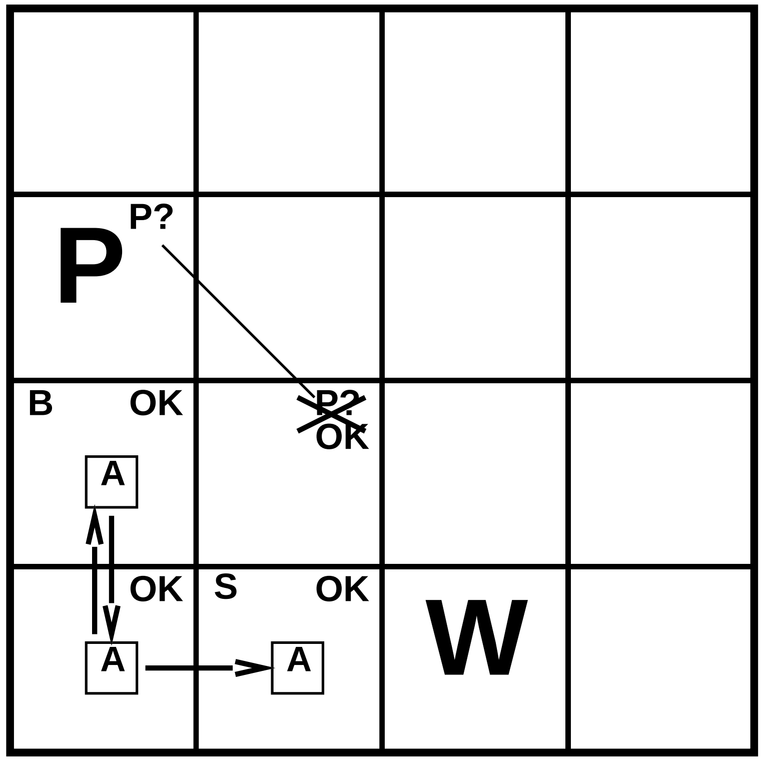
Exploring a wumpus world

Sensors: [Stench,None,None,None,None]

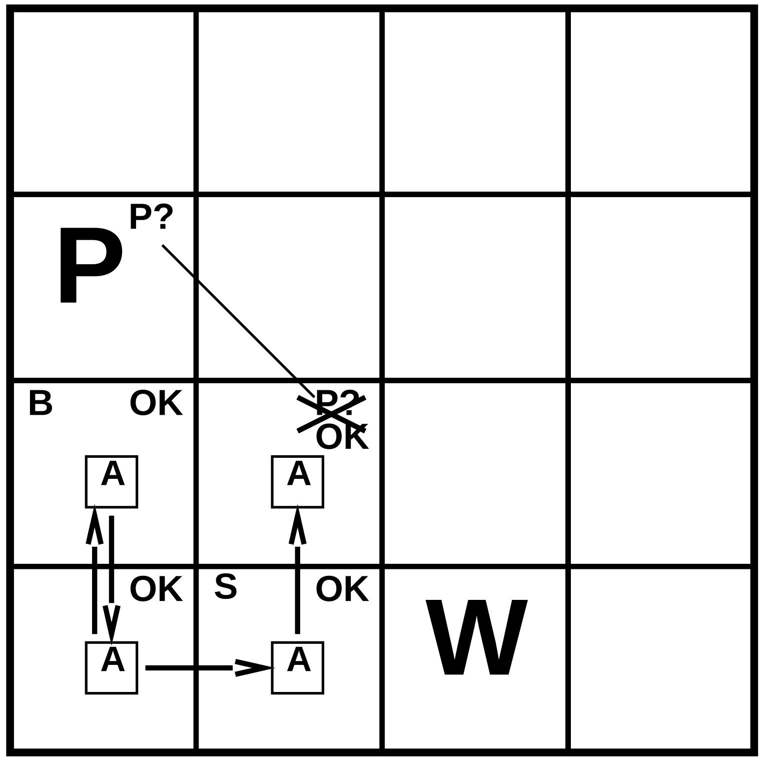


Exploring a wumpus world

Squares adjacent to wumpus are smelly

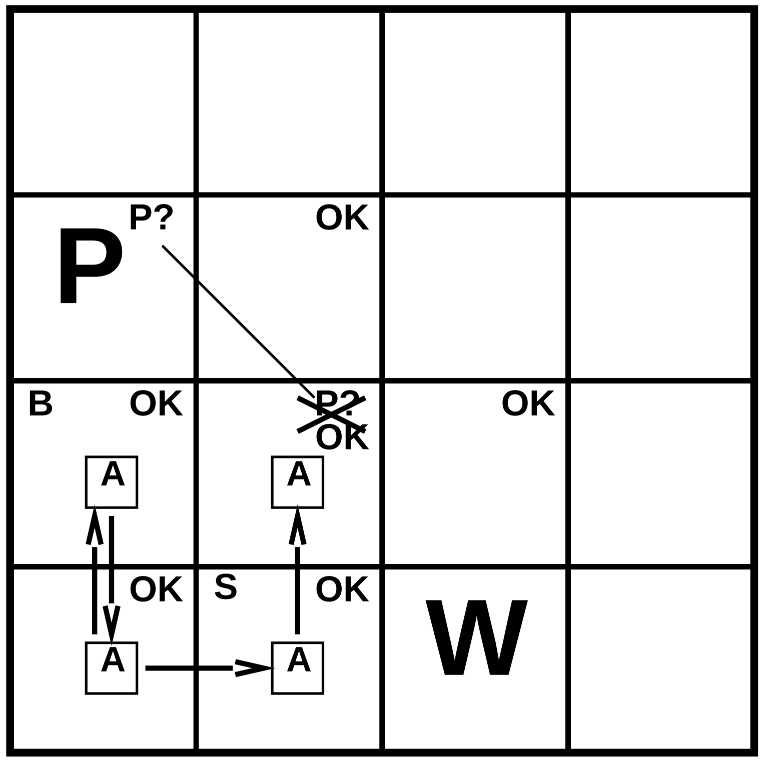


Exploring a wumpus world



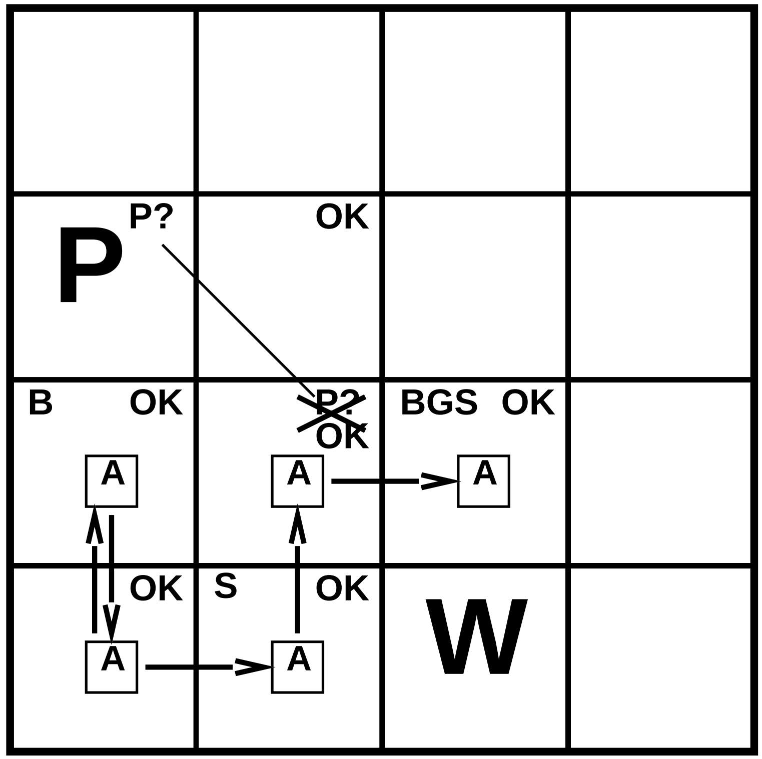
Exploring a wumpus world

Sensors: [None,None,None,None,None]



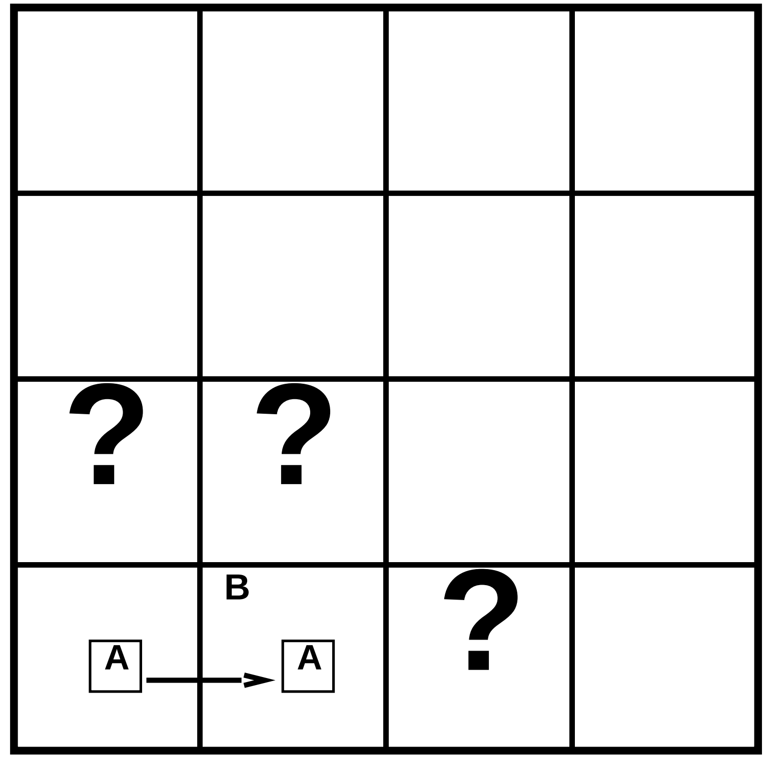
Exploring a wumpus world

Sensors: [None,None,Glitter,None,None]



Entailment in the wumpus world

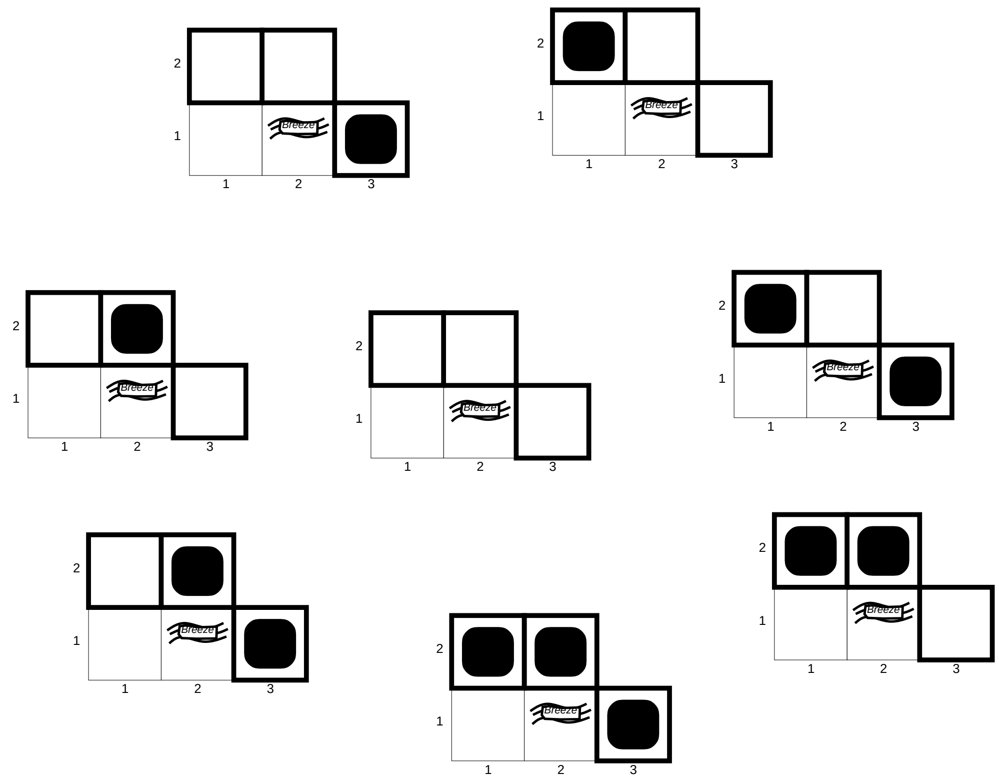
Situation after detecting nothing in [1,1],
moving right, breeze in [2,1]



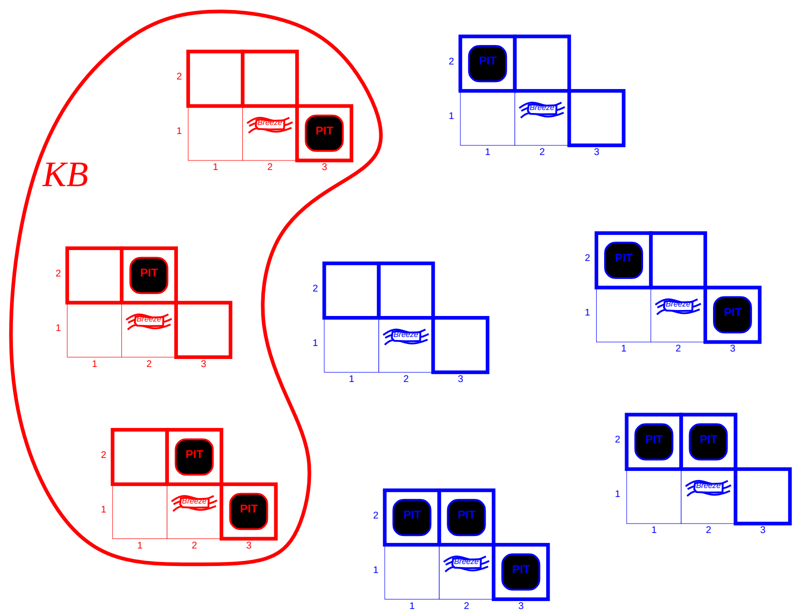
Consider possible cases for ?s assuming only pits

3 Boolean choices $\Rightarrow$ 8 possible cases

Wumpus possible worlds

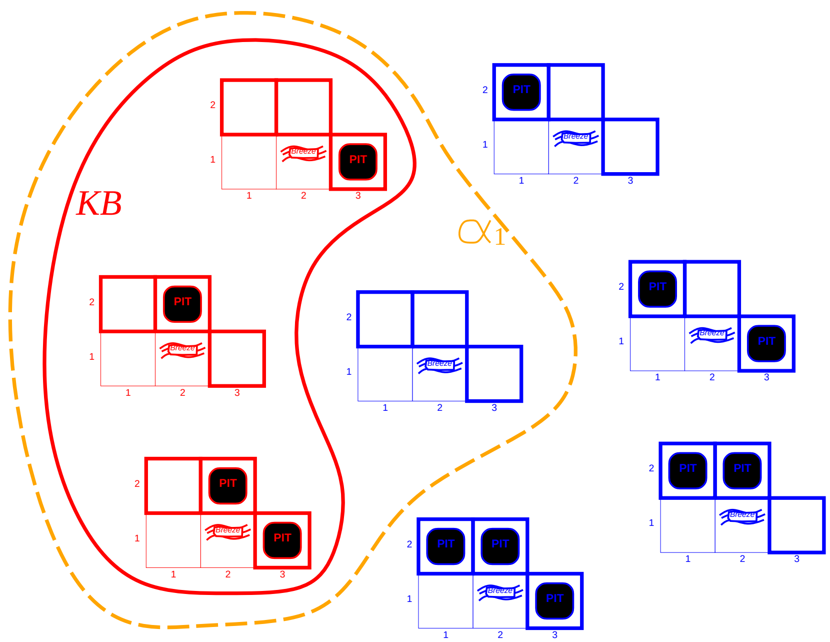


Wumpus models



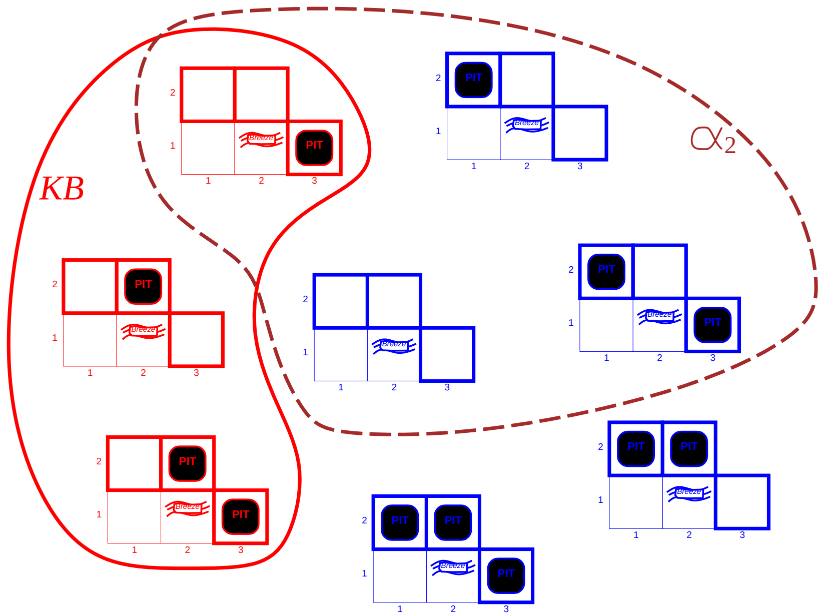
KB = wumpus-world rules + observation $B[1,1]$ and $B[1,2]$
models = possible worlds that are "coherent with KB

Inference by model checking



KB as before and $\alpha_1 =$ "[1,2] is safe"? $\qquad KB \models \alpha_1$

Inference by model checking



KB as before and $\alpha_2 =$ "[2,2] is safe"? $KB \not\models \alpha_2$

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 with "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

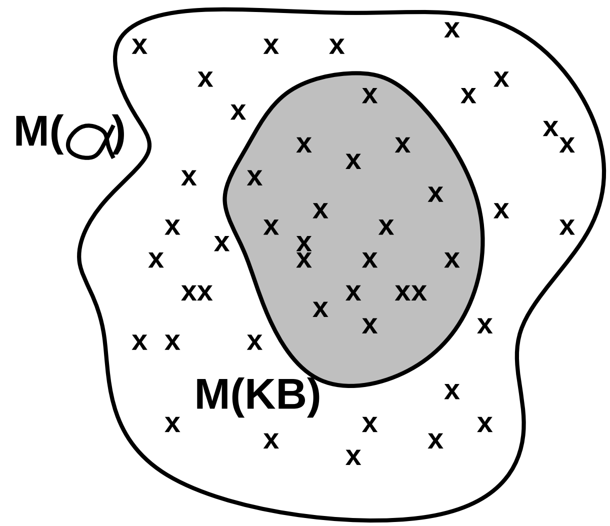
$M(\alpha)$ is the set of all models of α

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

NOTE: Russell and Norvig overload the term **model** by using it also for interpretation structures or possible worlds

Models: example

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



Model Checking

The knowledge that is entailed by a KB can be computed by:

$$KB \models \alpha$$

can be derived by building models and checking whether:

$$M(KB) \subseteq M(\alpha)$$

This approach is referred to as **model checking**

Deduction

Another way of computing the knowledge entailed by a KB is by a **deduction procedure** (proof):

$$KB \vdash_i \alpha$$

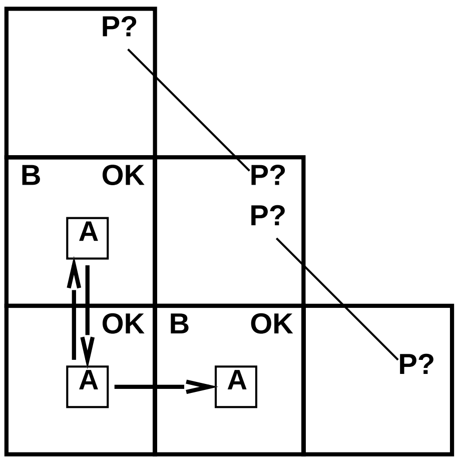
denotes that α can be derived from KB by procedure i

Deduction works on formulae by applying **inference** rules.

Note: the term inference is used for any procedure that can compute logical entailment.

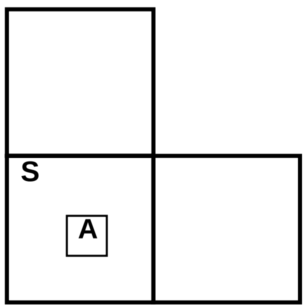
Sometimes there is no certainty

Breeze in (1,2) and (2,1)
$\Rightarrow$ no safe actions



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

Sometimes you can only try



Smell in (1,1)

$\Rightarrow$ cannot move

Can use a strategy of coercion:

shoot straight ahead

wumpus was there $\Rightarrow$ dead $\Rightarrow$ safe

wumpus wasn't there $\Rightarrow$ safe