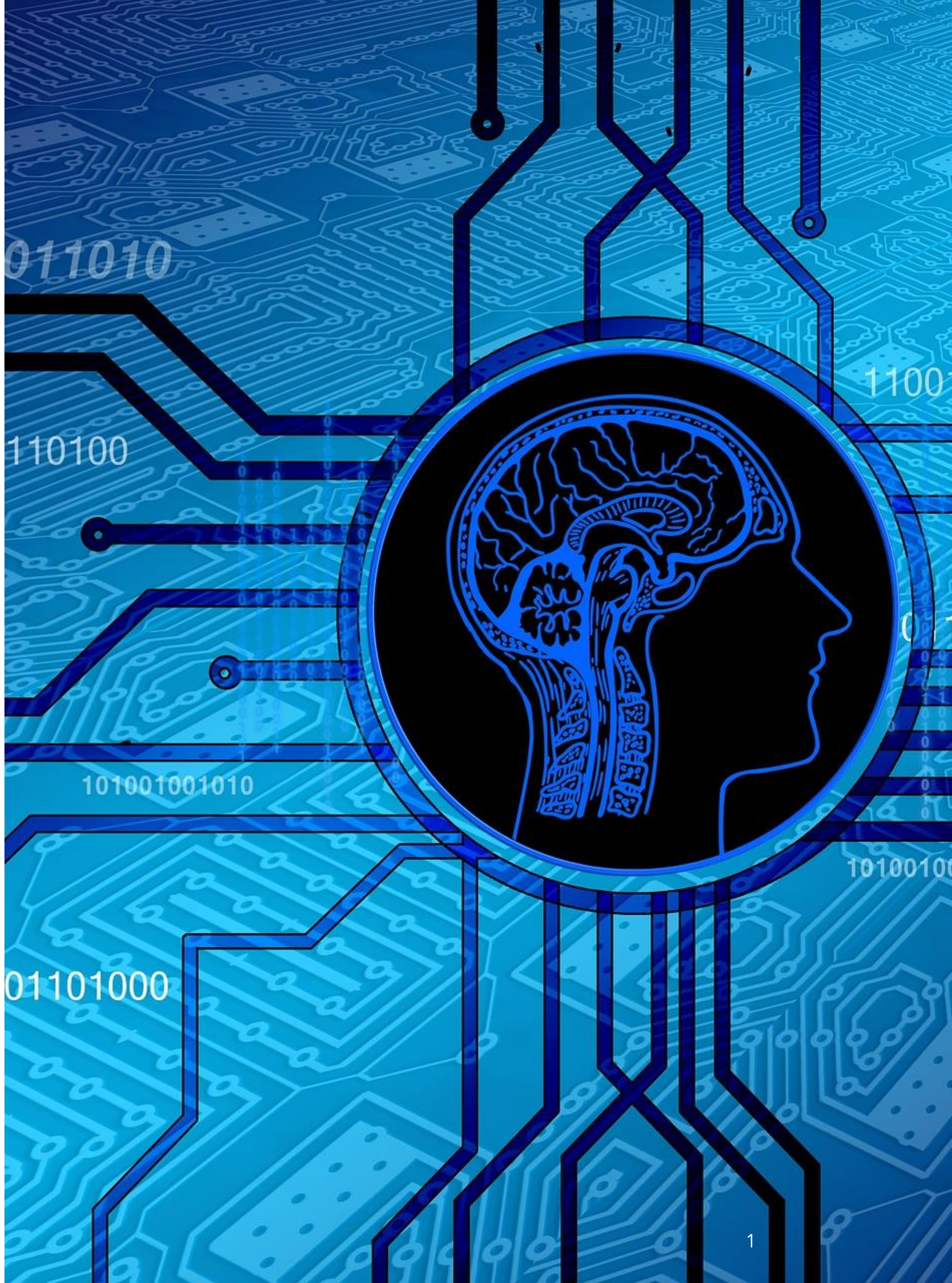
Logical Agents

Petros Papapanagiotou

Informatics 2D: Reasoning and Agents
Lecture 9



Knowledge bases

Inference engine	←	Domain-independent algorithms
Knowledge base	←	Domain-specific content

Knowledge base (KB) = set of sentences in a formal language

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

Then it can `Ask` itself what to do - answers should 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

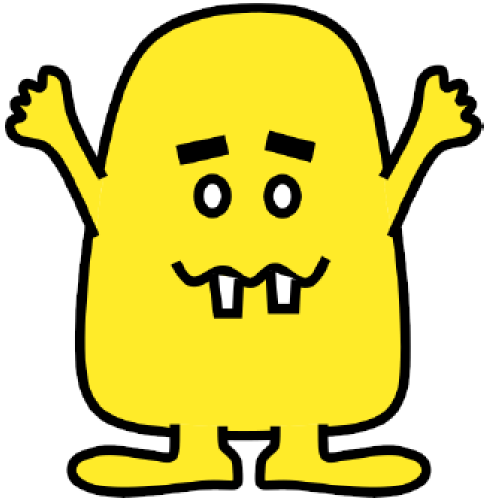
A simple knowledge-based agent

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

function KB-AGENT(*percept*) **returns** an *action*
persistent: *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*



Wumpus World

https://en.wikipedia.org/wiki/Hunt_the_Wumpus

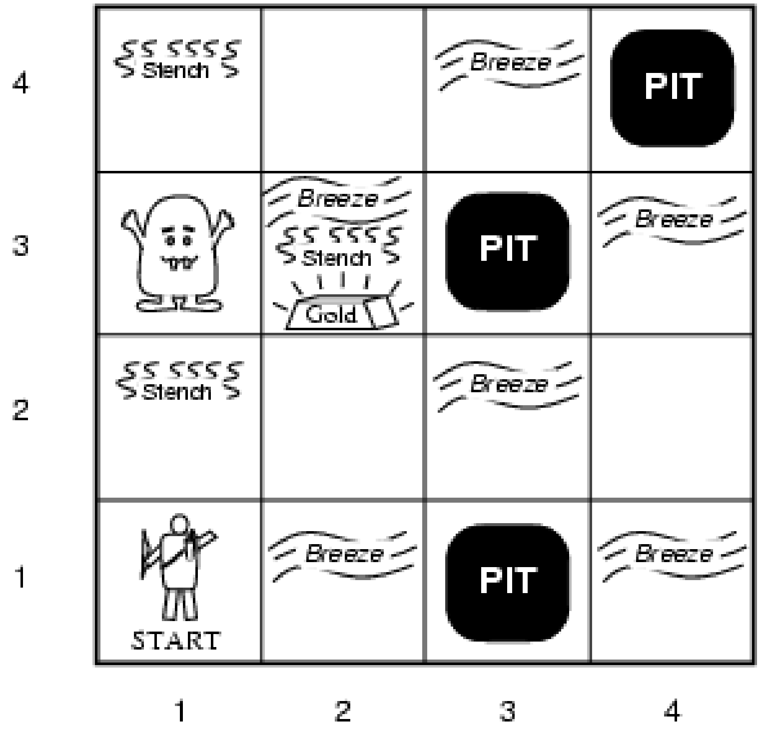
Wumpus World

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

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

Sensors: Stench, Breeze, Glitter, Bump, Scream

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



Wumpus World Environment Characterization

Observable	
Deterministic	
Episodic	
Static	
Discrete	
Single-agent	

Wumpus World Environment Characterization

Observable	• No - only local perception
Deterministic	
Episodic	
Static	
Discrete	
Single-agent	

Wumpus World Environment Characterization

Observable	• No - only local perception
Deterministic	• Yes – outcomes exactly specified
Episodic	
Static	
Discrete	
Single-agent	

Wumpus World Environment Characterization

Observable	• No – only local perception
Deterministic	• Yes – outcomes exactly specified
Episodic	• No – sequential at the level of actions
Static	
Discrete	
Single-agent	

Wumpus World Environment 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	
Single-agent	

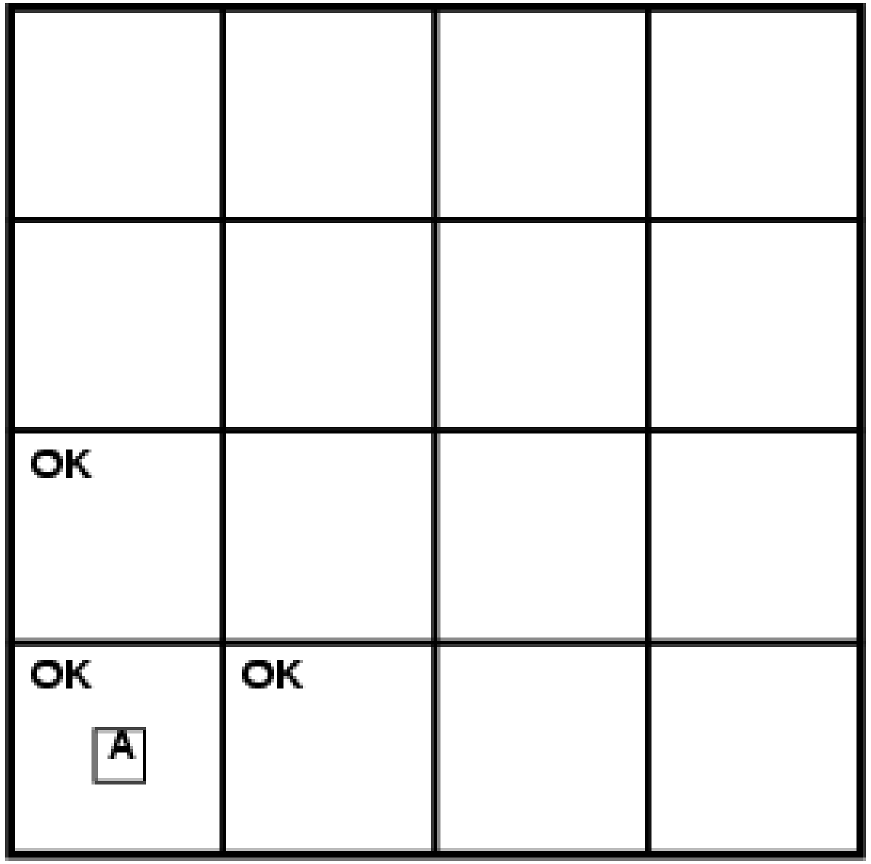
Wumpus World Environment Characterization

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	

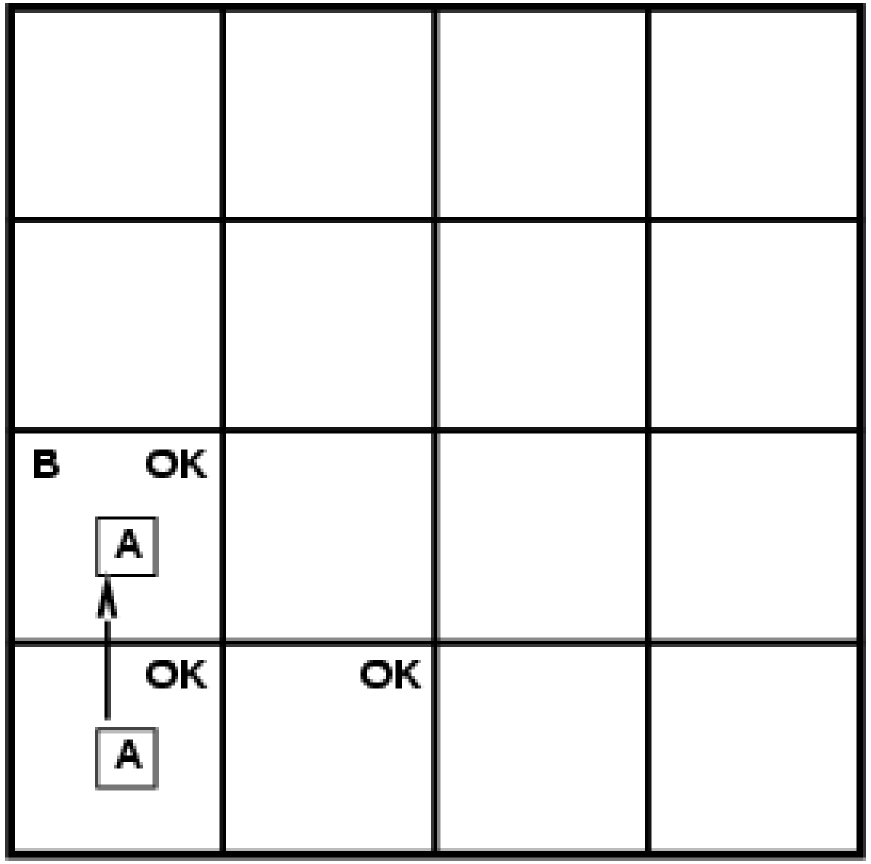
Wumpus World Environment Characterization

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

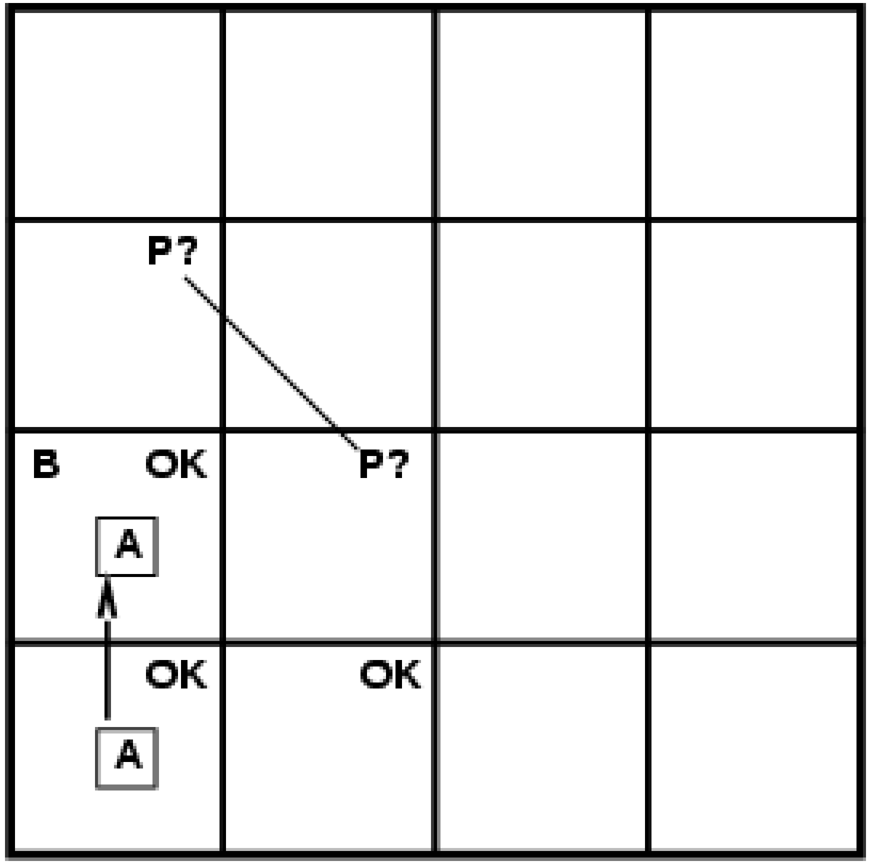
Exploring a wumpus world



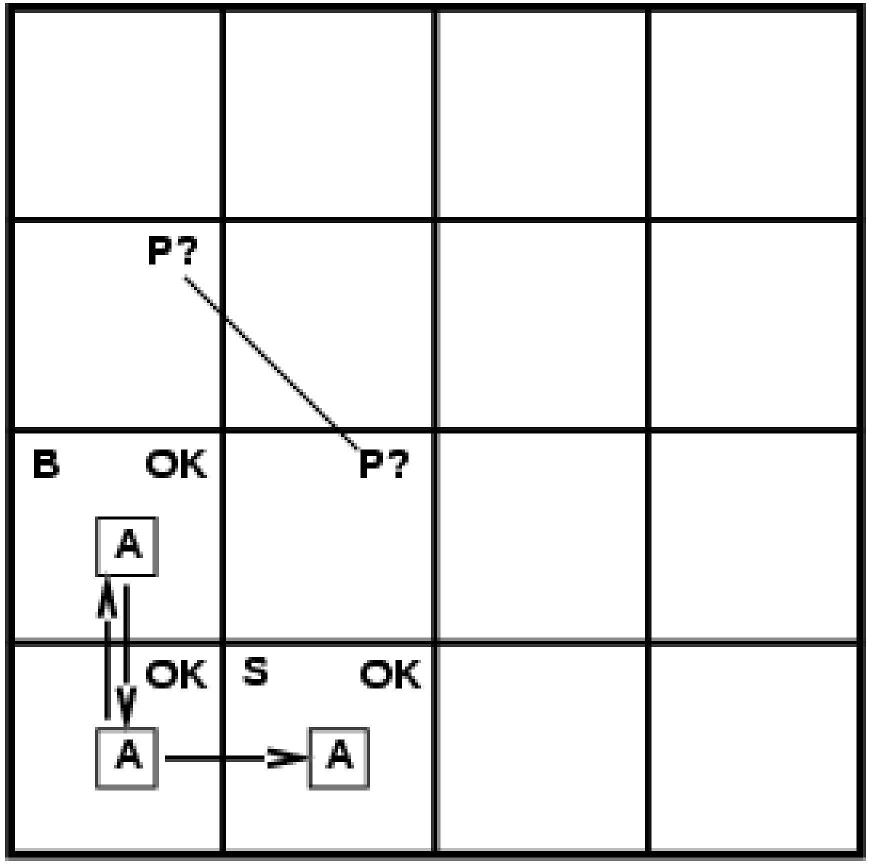
Exploring a wumpus world



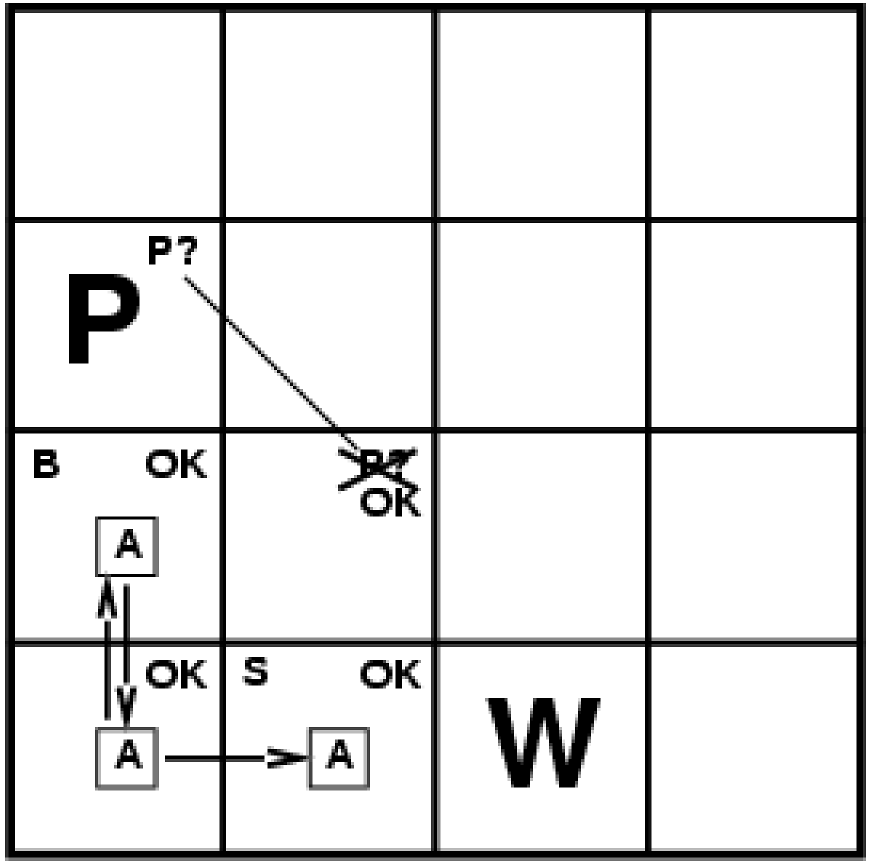
Exploring a wumpus world



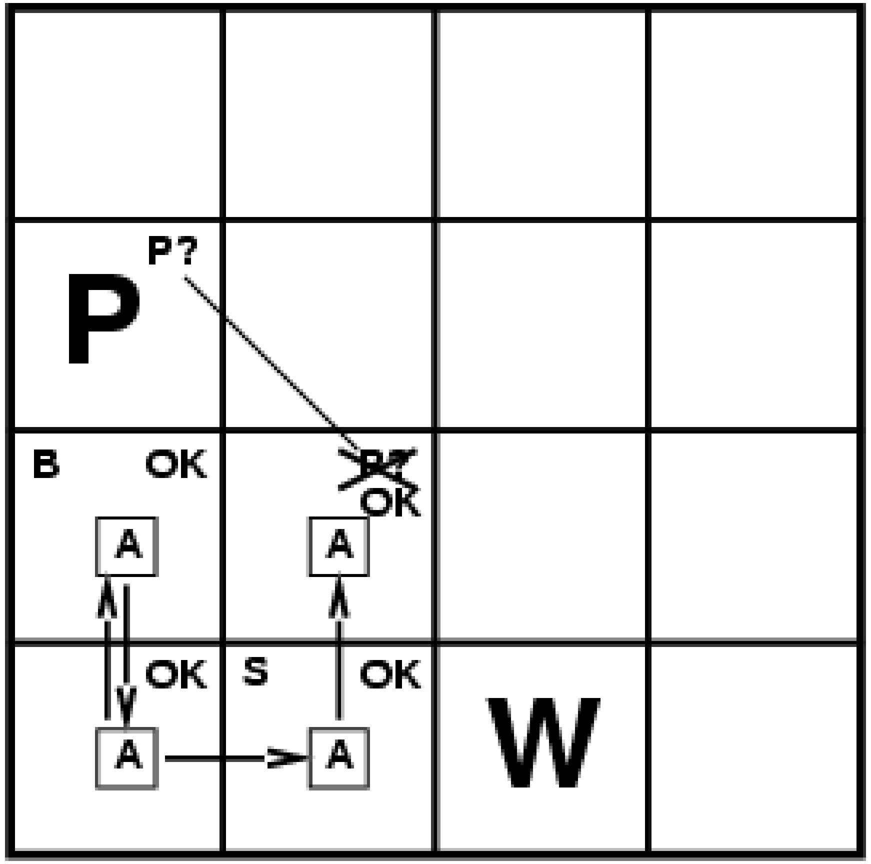
Exploring a wumpus world



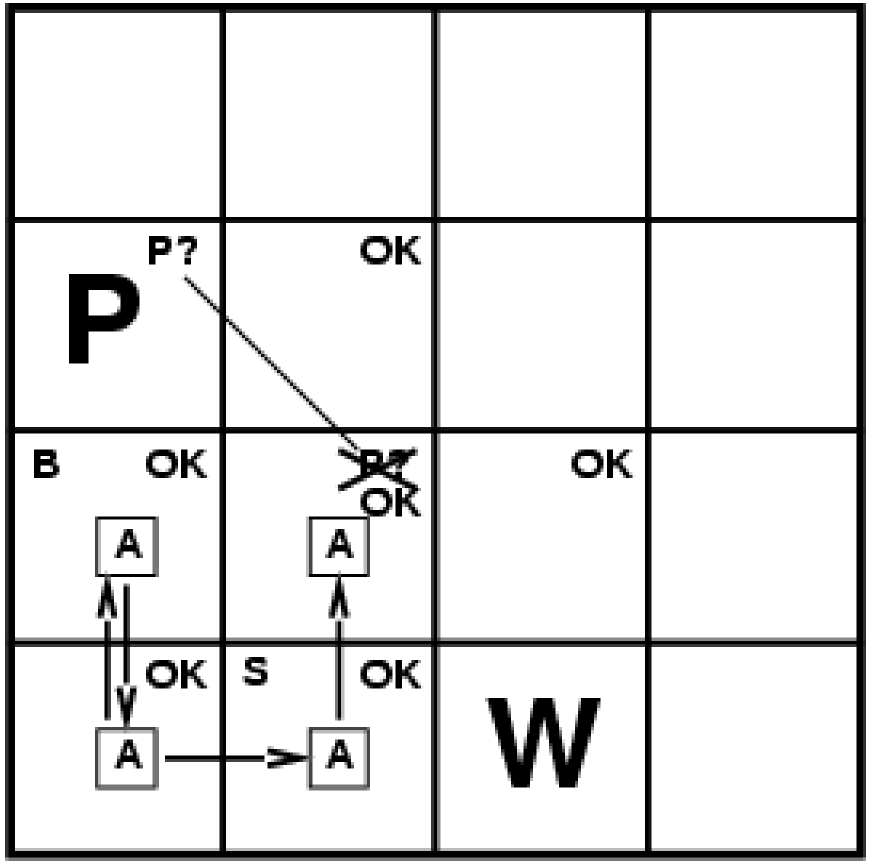
Exploring a wumpus world



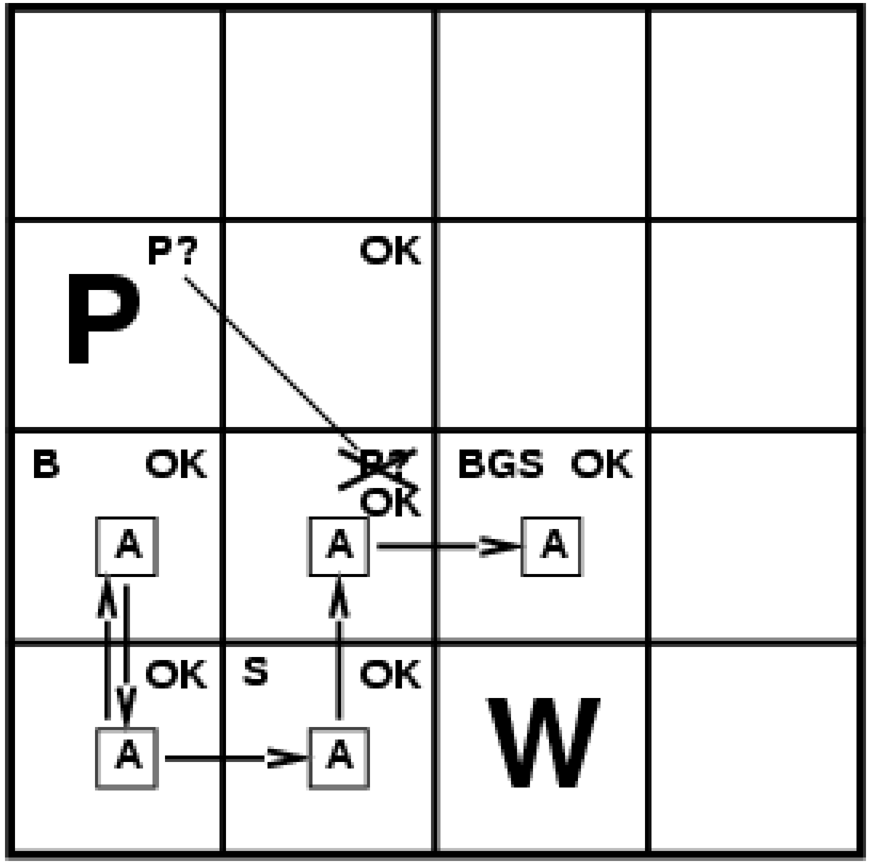
Exploring a wumpus world



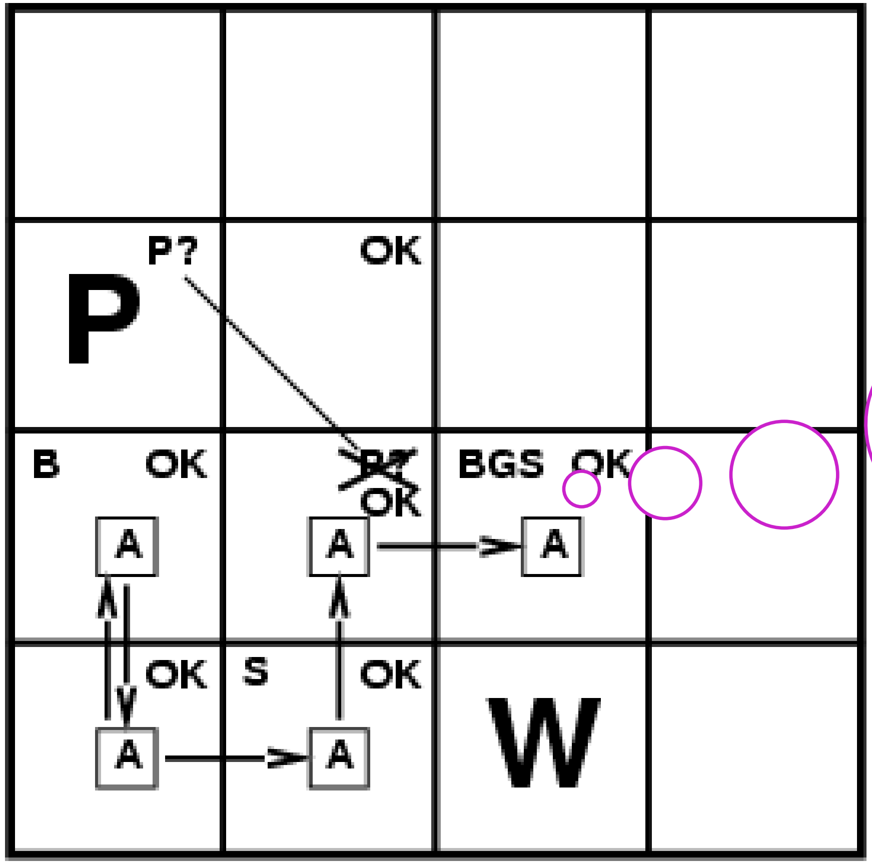
Exploring a wumpus world



Exploring a wumpus world



Exploring a wumpus world



Logical agents apply inference to a knowledge base to derive new information and make decisions.

Logic

Logic in general

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

Syntax defines the sentences in the language

Semantics defines the *meaning* of sentences; 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. x+y = 4 entails 4 = x+y
- e.g. the KB containing "*Celtic won*" and "*Hearts won*" entails "*Either Celtic won or Hearts won*"
- Entailment is a relationship between sentences (*syntax*) that is based on *semantics*

Models

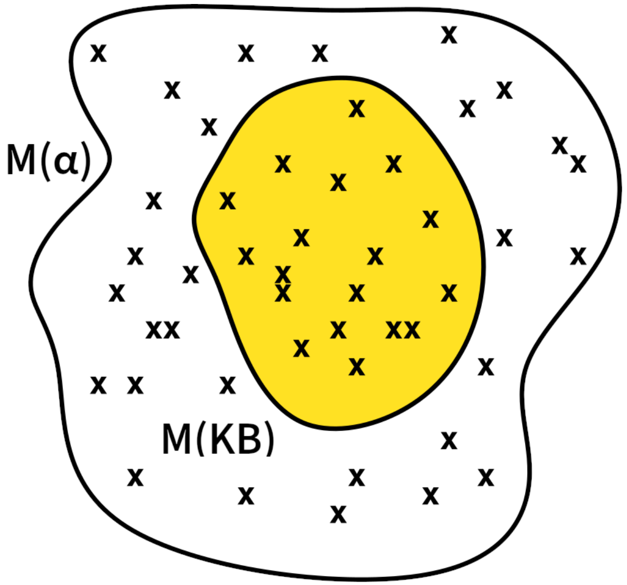
Logicians typically think in terms of models, which are formally structured worlds with respect to which truth can be evaluated

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

M(α) is the set of all models of α.

KB ⊨ α iff *M(KB)* ⊆ *M(α)*

The *stronger* an assertion, the fewer models it has.



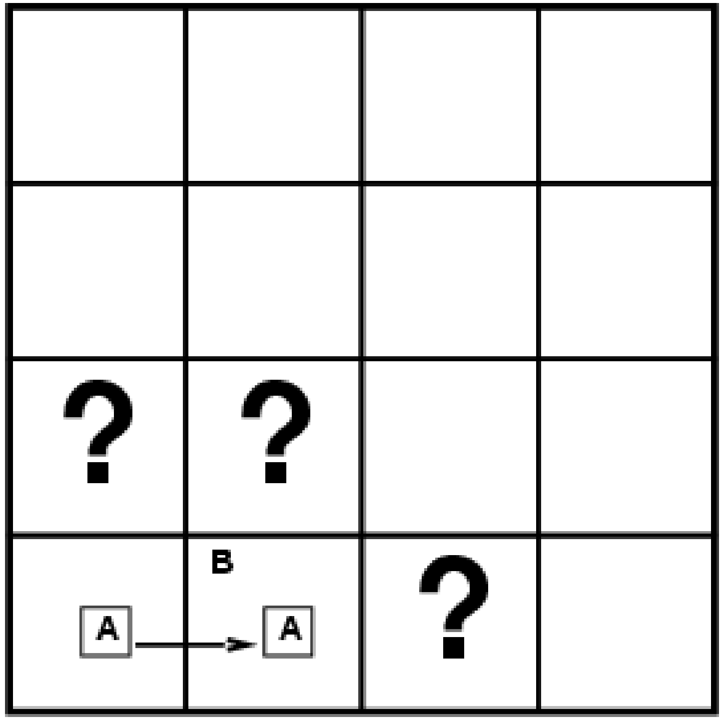
Entailment in the wumpus world

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

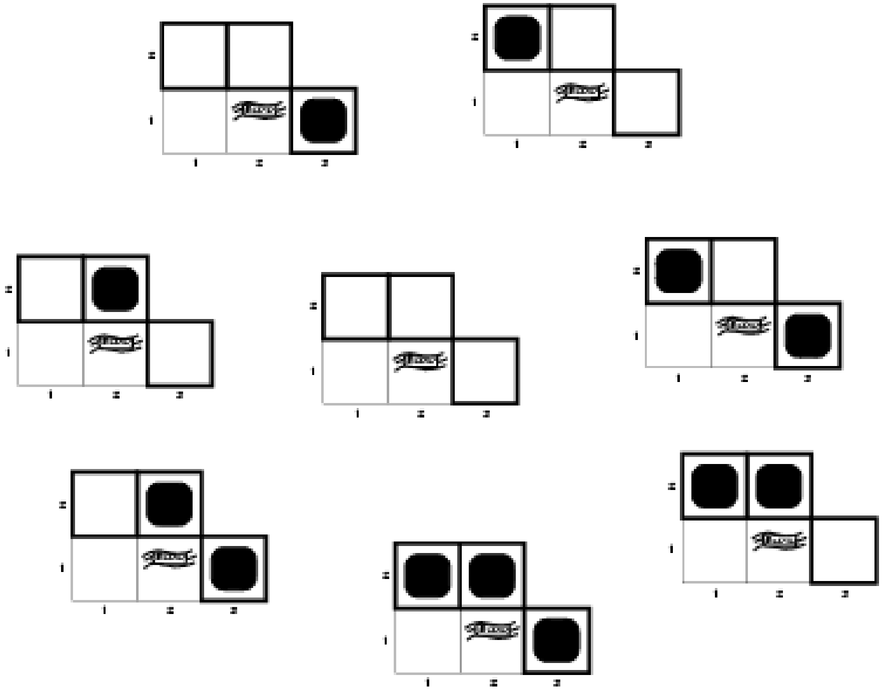
Consider possible models for *KB* assuming only pits

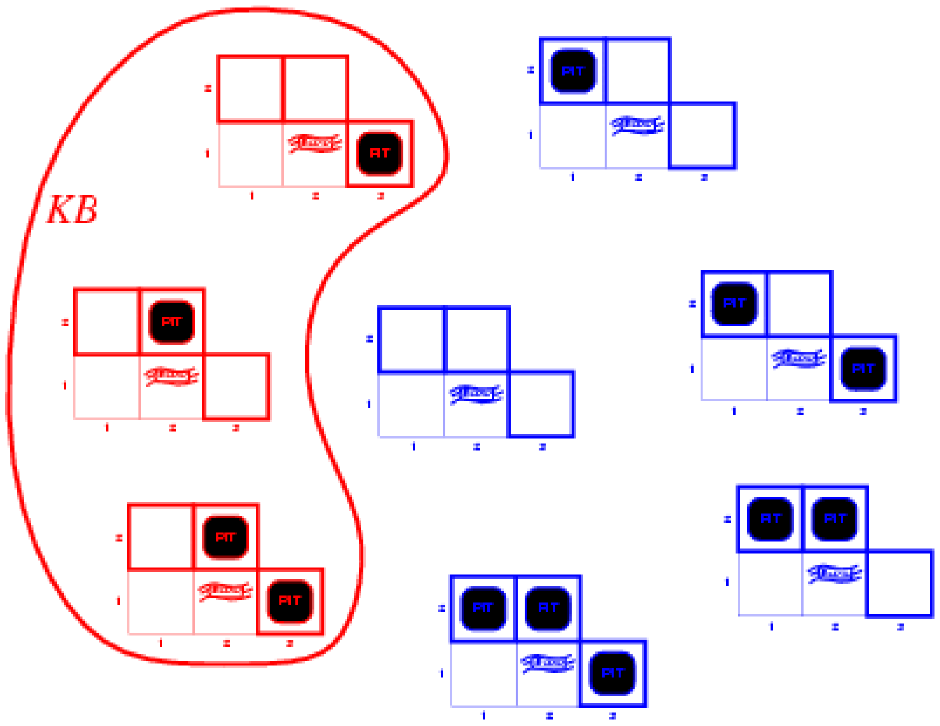
3 Boolean choices → 8 possible models

What are these 8 models?



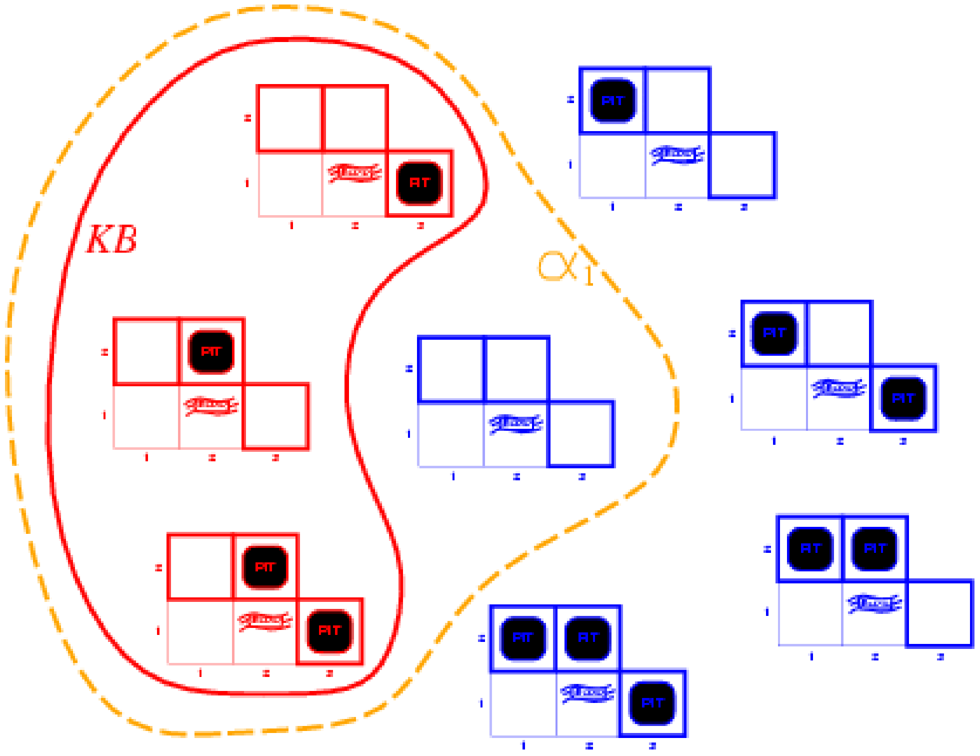
Wumpus models





Wumpus models

KB = wumpus-world rules + observations



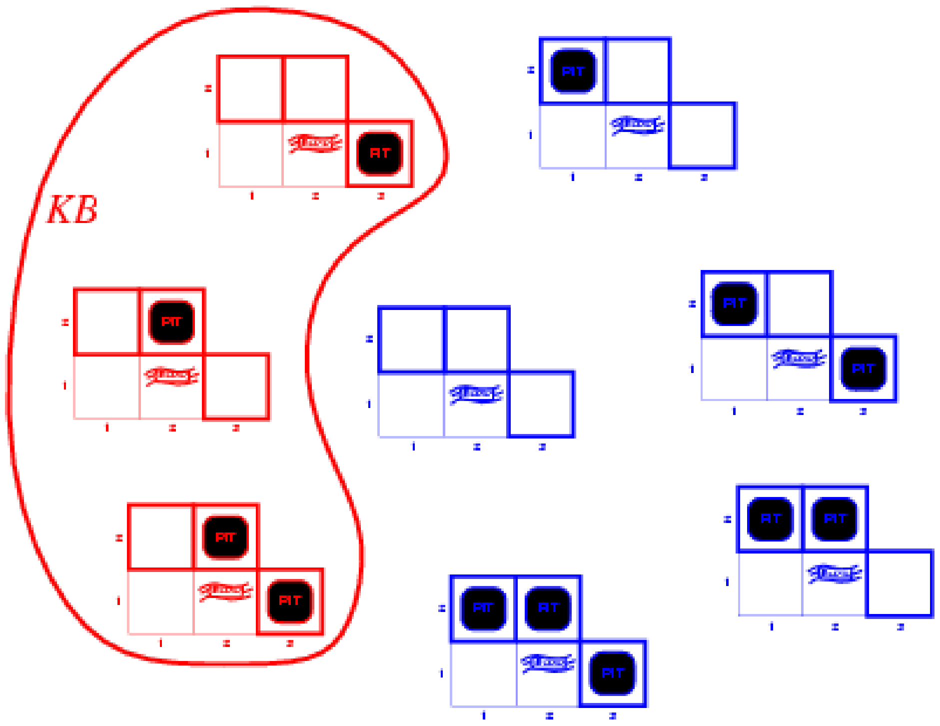
Wumpus models

KB = wumpus-world rules + observations

α_1 = "[1,2] has no pit"

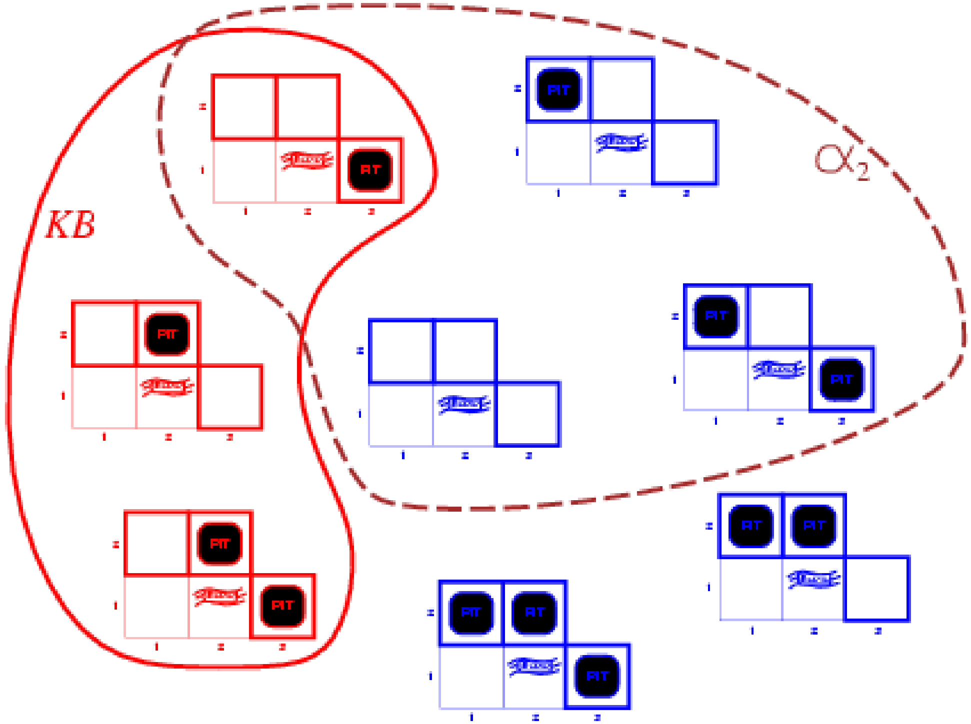
$KB \vDash \alpha_1$, proved by model checking

- In every model in which KB is true, α_1 is also true



Wumpus models

KB = wumpus-world rules + observations



Wumpus models

KB = wumpus-world rules + observations

α_2 = "[2,2] has no pit"

$KB \nvDash \alpha_2$

- In some models in which KB is true, α_2 is false

Inference

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

Soundness

- i is sound if whenever $KB \vdash_i \alpha$, it is also true that $KB \vDash \alpha$

Completeness

- i is complete if whenever $KB \vDash \alpha$, it is also true that $KB \vdash_i \alpha$

Preview: we will define first-order logic:

- expressive enough to say almost anything of interest,
- sound and complete inference procedure exists.

But first…

Propositional logic

P	$\neg$	$\wedge$
$\vee$	$\Rightarrow$	$\Leftrightarrow$

Propositional logic: Syntax

Propositional logic is the simplest logic – illustrates basic ideas

- The proposition symbols P_1, P_2 etc are sentences
- If S is a sentence, $\neg S$ is a sentence [negation]
- If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence [conjunction]
- If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence [disjunction]
- If S_1 and S_2 are sentences, $S_1 \Rightarrow S_2$ is a sentence [implication]
- If S_1 and S_2 are sentences, $S_1 \Leftrightarrow S_2$ is a sentence [biconditional]

Propositional logic: Semantics

Each model specifies true/false for each proposition symbol

e.g.	$P_{1,2}$	$P_{2,2}$	$P_{3,1}$
	false	*true*	*false*

With these symbols, 8 possible models

- can be enumerated automatically!

Propositional logic: Semantics

Rules for evaluating truth with respect to a model *m*:

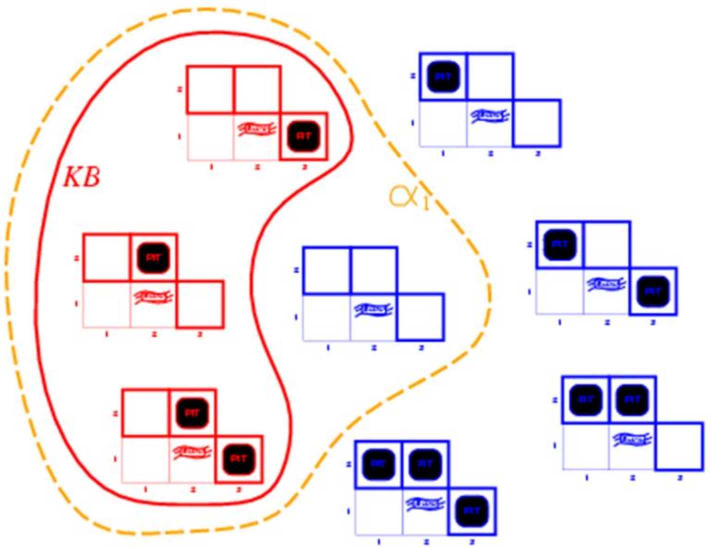
$\neg S$	is true	iff	S is false
$S_1 \land S_2$	is true	iff	S_1 is true and S_2 is true
$S_1 \lor S_2$	is true	iff	S_1 is true or S_2 is true
$S_1 \Rightarrow S_2$	is true	iff	S_1 is false or S_2 is true
i.e.	is false	iff	S_1 is true and S_2 is false
$S_1 \Leftrightarrow S_2$	is true	iff	$S_1 \Rightarrow S_2$ is true and $S_2 \Rightarrow S_1$ is true

Simple recursive process evaluates an arbitrary sentence:

$$\neg P_{1,2} \land (P_{2,2} \lor P_{3,1}) = \mathit{true} \land (\mathit{true} \lor \mathit{false}) = \mathit{true} \land \mathit{true} = \mathit{true}$$

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

Truth tables for connectives



Wumpus world sentences

Let $P_{i,j}$ be true if there is a pit in [i, j].

Let $B_{i,j}$ be true if there is a breeze in [i, j].

$$\neg P_{1,1} \qquad \neg B_{1,1} \qquad B_{2,1}$$

"Pits cause breezes in adjacent squares"

$B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})$

$B_{2,1} \Leftrightarrow (P_{1,1} \lor P_{2,2} \lor P_{3,1})$

α_1 = "[1,2] has no pit"

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	KB	α_1
false	*false*	*false*	*false*	*false*	*false*	*false*	*false*	*true*
false	*false*	*false*	*false*	*false*	*false*	*true*	*false*	*true*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*false*	*false*	*false*	*false*	*false*	*false*	*true*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*false*	*true*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*false*

Truth tables for inference

function TT-ENTAILS?(KB, α) **returns** *true* or *false*
  inputs: KB, the knowledge base, a sentence in propositional logic
    α, the query, a sentence in propositional logic

  symbols ← a list of the proposition symbols in KB and α
  return TT-CHECK-ALL($KB, \alpha, symbols, \{\,\}$)

---

function TT-CHECK-ALL($KB, \alpha, symbols, model$) **returns** *true* or *false*
  if EMPTY?(*symbols*) **then**
    if PL-TRUE?($KB, model$) **then return** PL-TRUE?($\alpha, model$)
    else return *true* // *when KB is false, always return true*
  else do
    P ← FIRST(*symbols*)
    rest ← REST(*symbols*)
    return (TT-CHECK-ALL($KB, \alpha, rest, model \cup \{P = true\}$)
      and
      TT-CHECK-ALL($KB, \alpha, rest, model \cup \{P = false\}$))

Inference by enumeration

Depth-first enumeration of all models is sound and complete

PL-TRUE?

- returns true if a sentence holds within a model

For *n* symbols

- Time complexity is $O(2^n)$
- Space complexity is $O(n)$

Logical equivalence

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha) \equiv \alpha$	double-negation elimination
$(\alpha \longrightarrow \beta) \equiv (\neg\beta \longrightarrow \neg\alpha)$	contraposition
$(\alpha \longrightarrow \beta) \equiv (\neg\alpha \vee \beta)$	implication elimination
$(\alpha \leftrightarrow \beta) \equiv ((\alpha \longrightarrow \beta) \wedge (\beta \longrightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$	de Morgan
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$	de Morgan
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

Two sentences are logically equivalent iff true in the same models:

$$\alpha \equiv \beta \text{ iff } \alpha \models \beta \text{ and } \beta \models \alpha$$

Validity and satisfiability

A sentence is **valid** if it is true in *all models*

- *true*, $A \lor \neg A$, $A \Rightarrow A$, $(A \land (A \Rightarrow B)) \Rightarrow B$

Validity is connected to inference via the **Deduction Theorem**

- $KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

A sentence is **satisfiable** if it is true in *some model*

- e.g., $A \lor B$, C

A sentence is **unsatisfiable** if it is true in *no models*

- e.g., $A \land \neg A$

Satisfiability is connected to inference via the following:

- $KB \models \alpha$ if and only if $(KB \land \neg\alpha)$ is unsatisfiable
- prove α by *reductio ad absurdum*

Proof methods

APPLICATION OF INFERENCE RULES

- Legitimate (sound) generation of new sentences from old
- **Proof** = a sequence of inference rule applications
  - Can use inference rules as operators in a standard search algorithm!
- Typically require transformation of sentences into a **normal form**
- Example: resolution

MODEL CHECKING

- truth table enumeration
  - (always exponential in n)
- improved backtracking
  - e.g. DPLL
- heuristic search in model space
  - (sound but incomplete)
  - e.g. min-conflicts-like hill-climbing algorithms

Summary

Logical agents apply inference to a knowledge base to derive new information and make decisions

Basic concepts of logic:

- syntax: formal structure of sentences
- semantics: truth of sentences wrt models
- entailment: necessary truth of one sentence given another
- inference: deriving sentences from other sentences
- soundness: derivations produce only entailed sentences
- completeness: derivations can produce all entailed sentences

Wumpus world requires the ability to represent partial and negated information, reason by cases, etc.

Does propositional logic have enough expressive power?