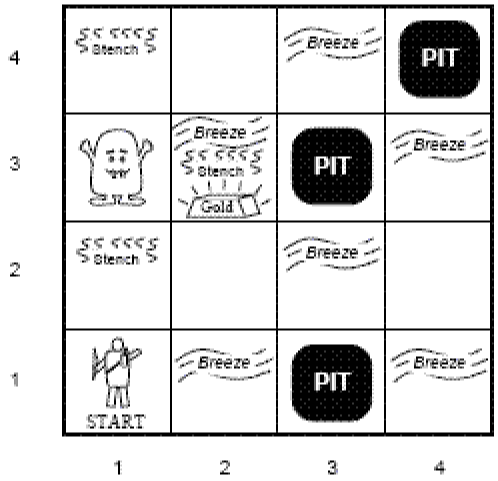
LOGICAL AGENTS

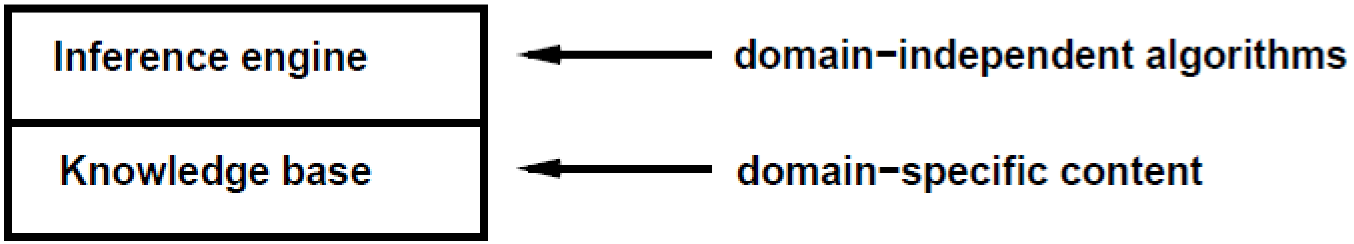


Outline

- Knowledge-based agents
- Wumpus world
- Logic in general - models and entailment
- Propositional (Boolean) logic
- Equivalence, validity, satisfiability

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 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

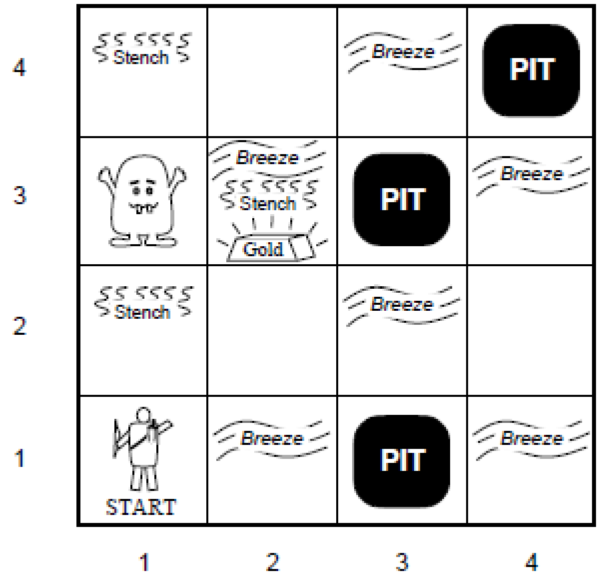
- The agent must be able to:
  - Represent states, actions, etc.
  - Incorporate new percepts
  - Update internal representations of the world
  - Deduce hidden properties of the world
  - Deduce appropriate actions

```
function KB-AGENT(percept) returns an action
    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
```

Wumpus World PEAS Description

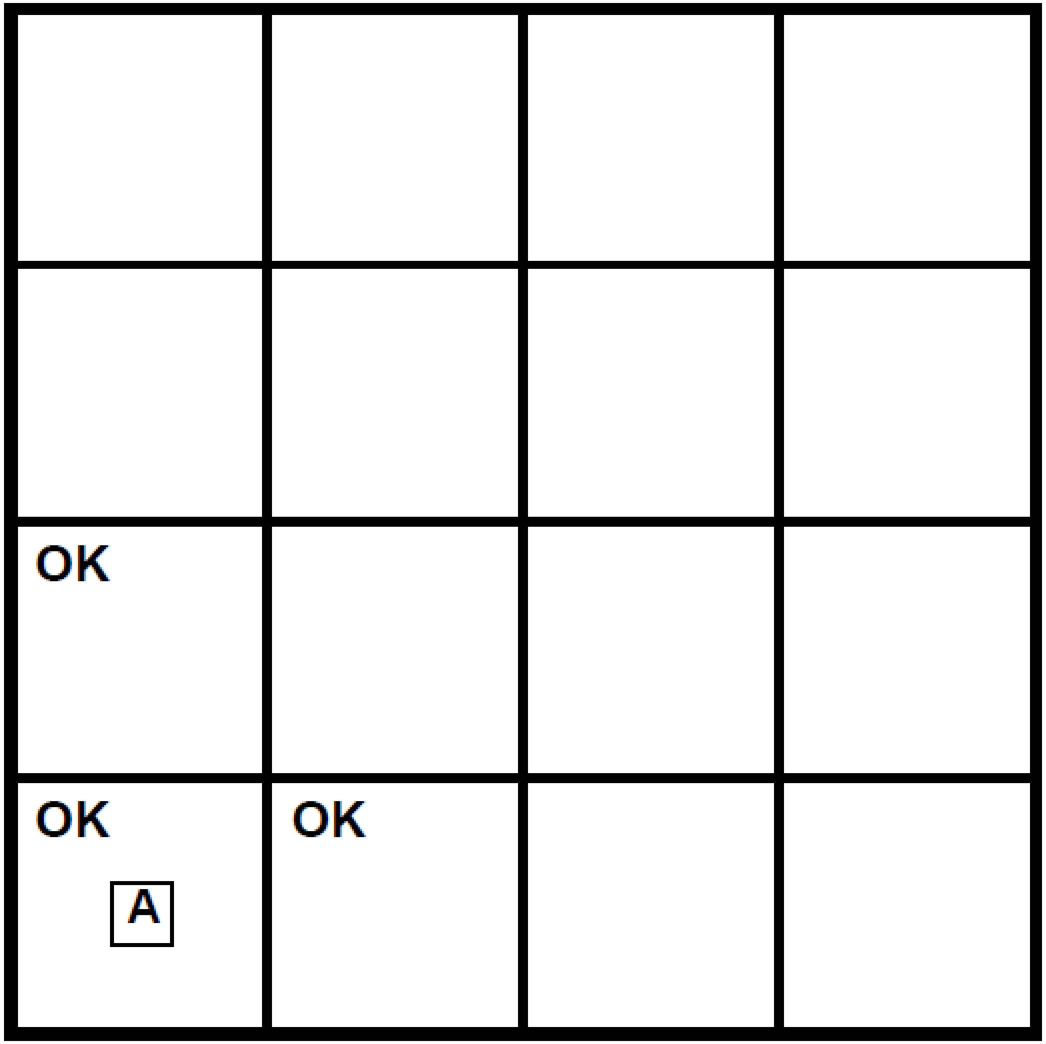
- Performance measure
  - gold +1000, death -1000
  - -1 per step, -10 for using the arrow
- Environment
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy
  - Glitter if gold is in the same square
  - Shooting kills wumpus if you are facing it
  - Shooting uses up the only 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

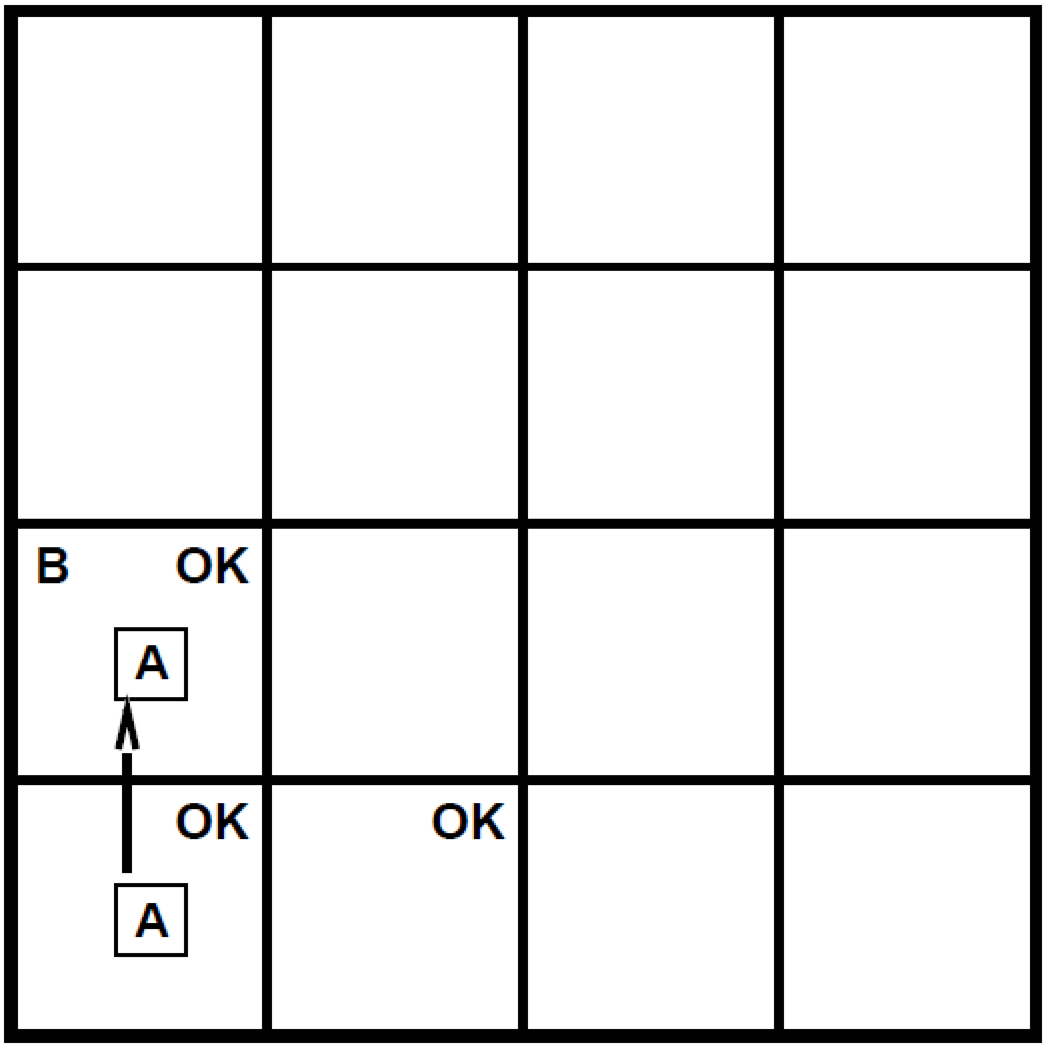


Wumpus World Characterization

- Observable??
  - 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

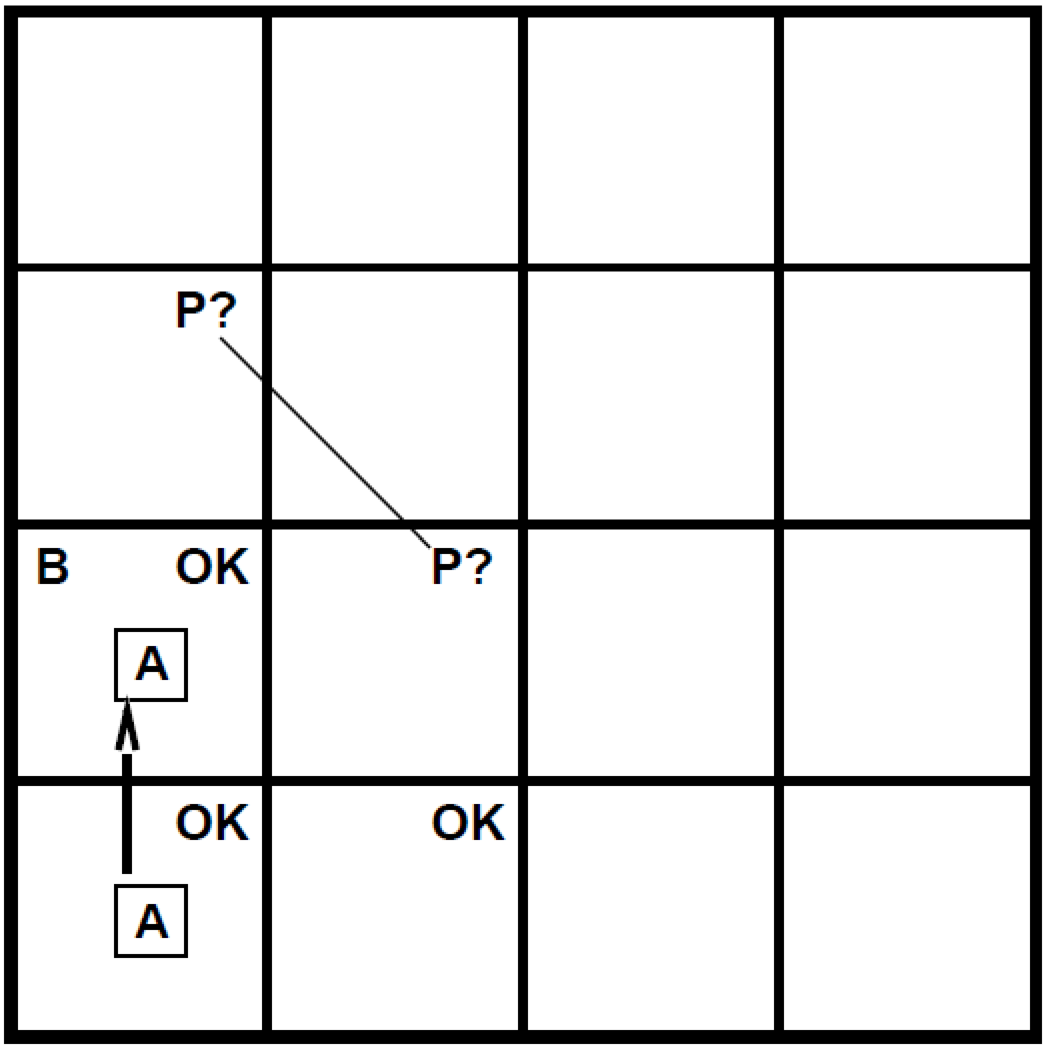
Exploring a Wumpus World



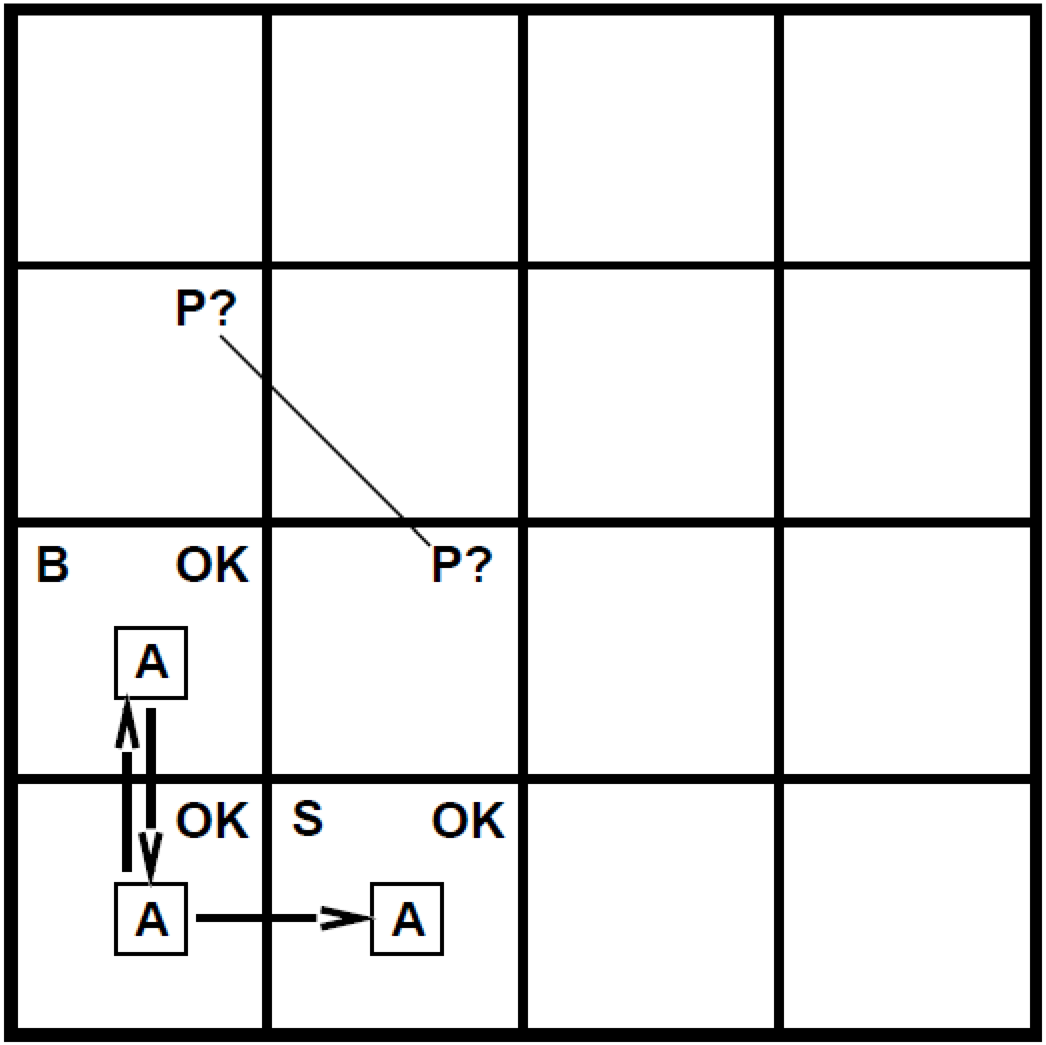


Exploring a Wumpus World

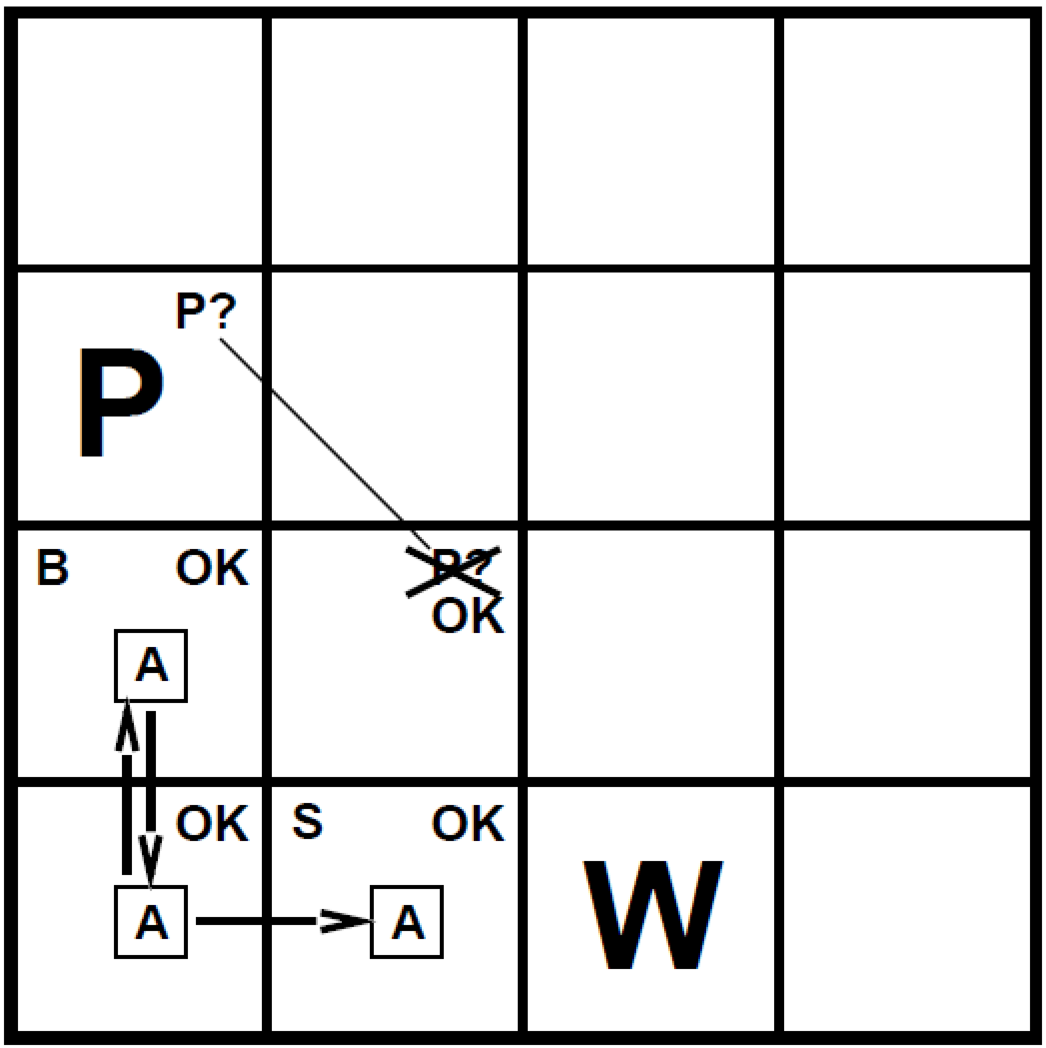
Exploring a Wumpus World



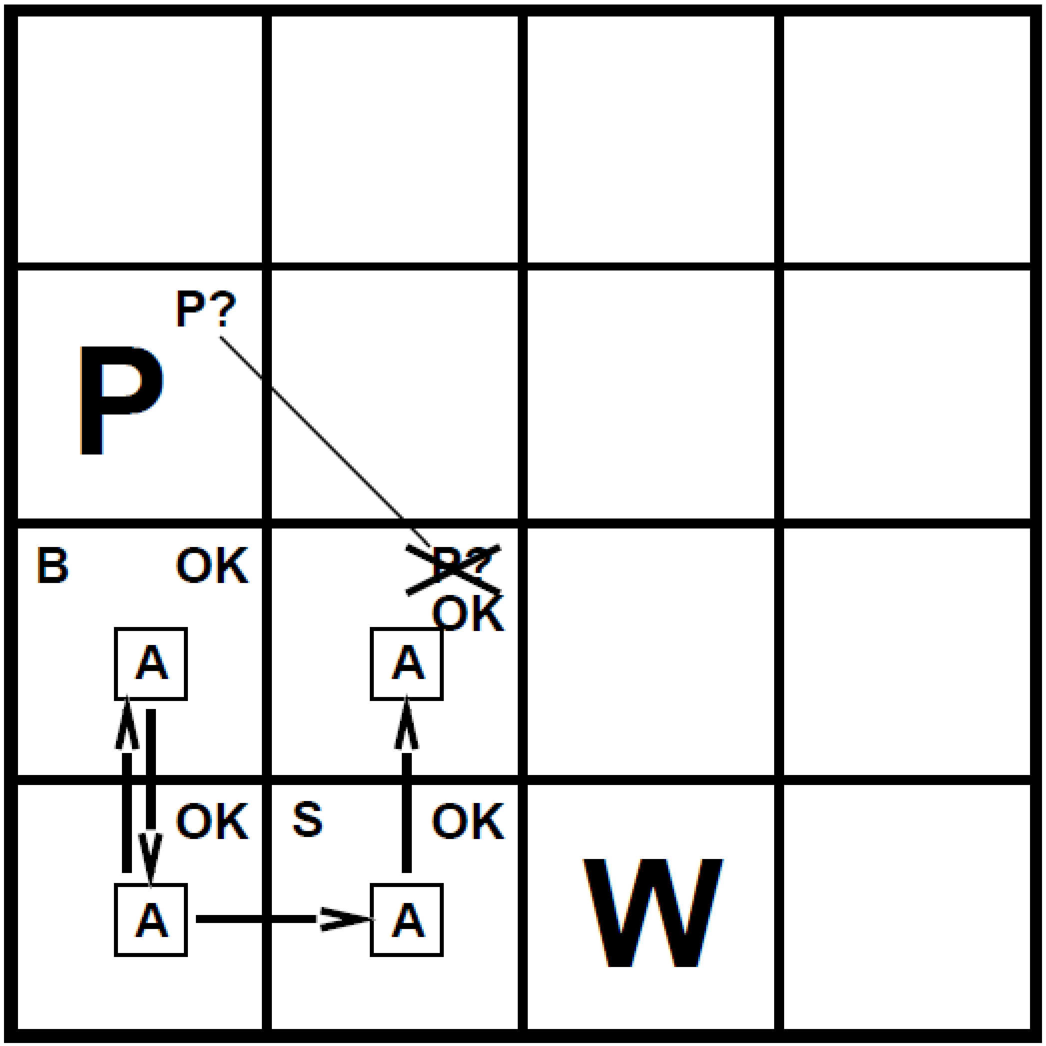
Exploring a Wumpus World



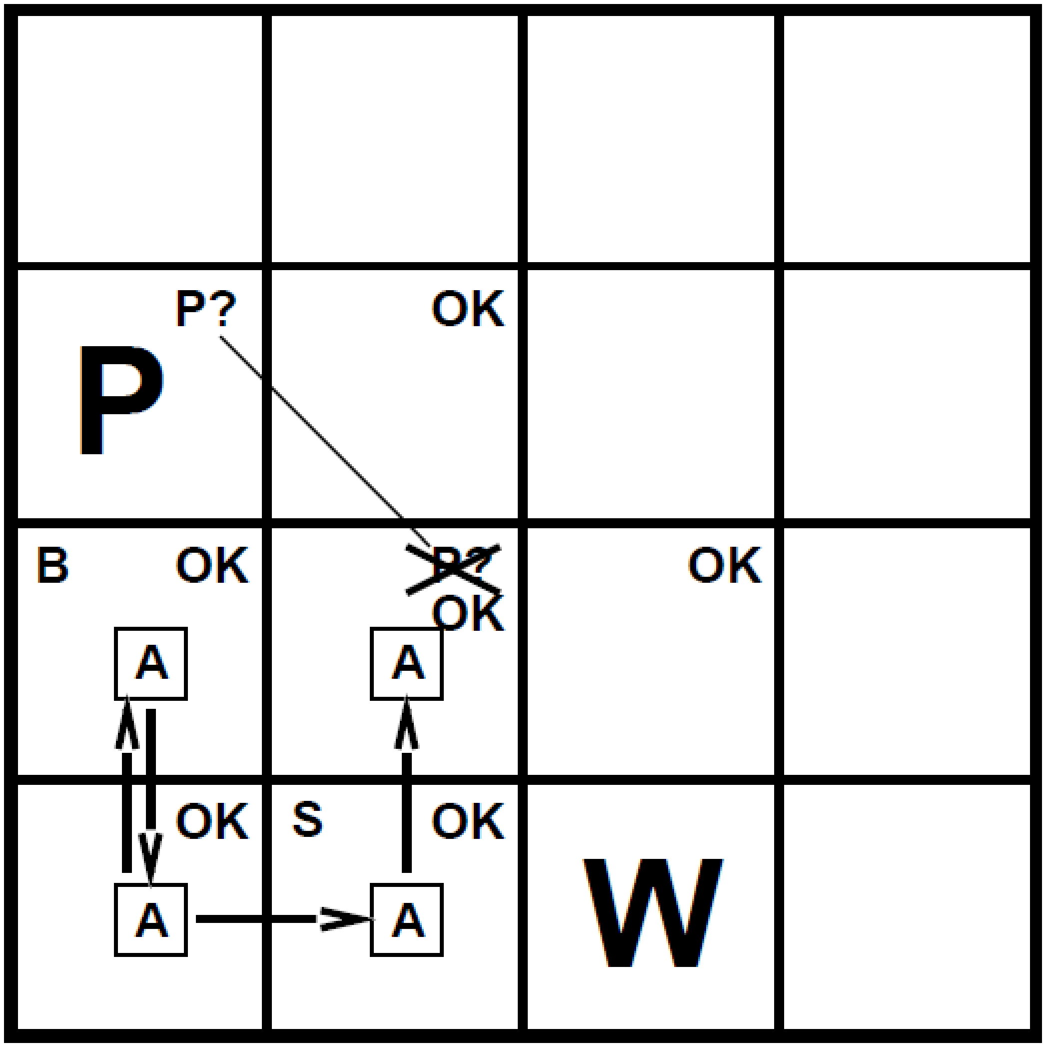
Exploring a Wumpus World



Exploring a Wumpus World

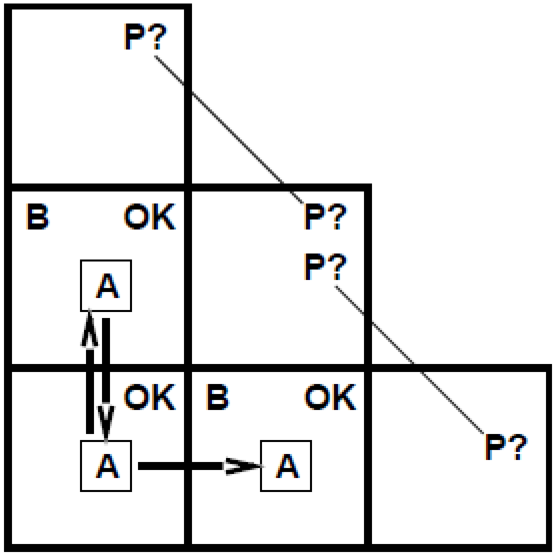


Exploring a Wumpus World

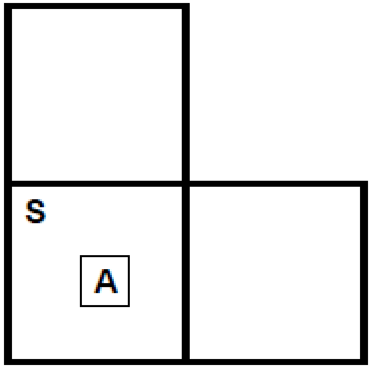


Other Tight Spots

- 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 >= y is a sentence; x2 + y > is not a sentence
  - x + 2 >= y is true iff the number x + 2 is no less than the number y
  - x + 2 >= y is true in a world where x=7, y =1
  - x + 2 >= 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
- Entailment is a relationship between sentences (i.e., syntax) that is based on semantics

Models

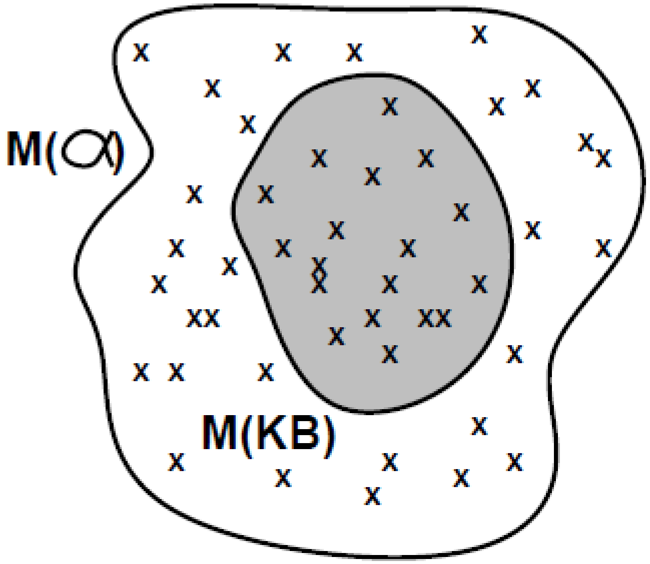
- Logicians typically think in terms of models, which are formally structured worlds with respect to which truth can be evaluated
- We say m is a model of a sentence α if α is true in m
- M(α) is the set of all models of α
- Then

$$KB \models \alpha$$

- if and only if

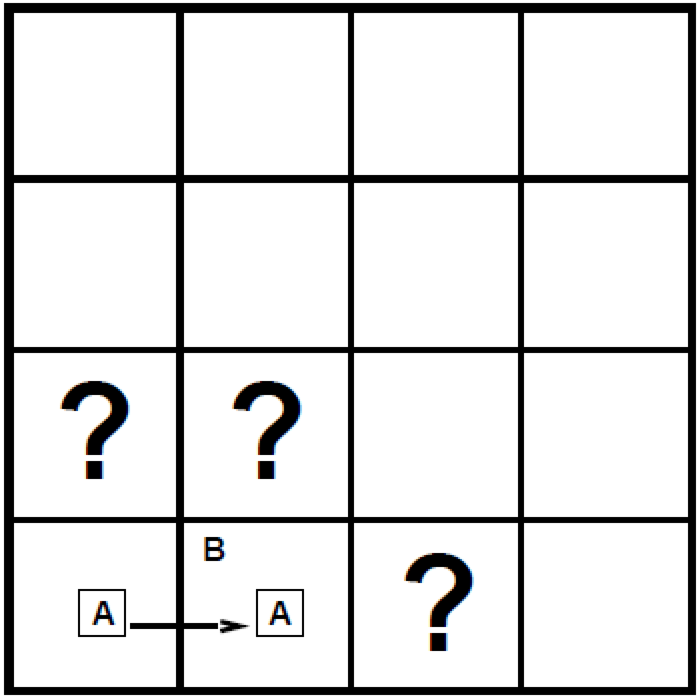
$$M(KB) \subseteq M(\alpha)$$

- E.g. KB = Giants won and Reds won
- α = Giants won

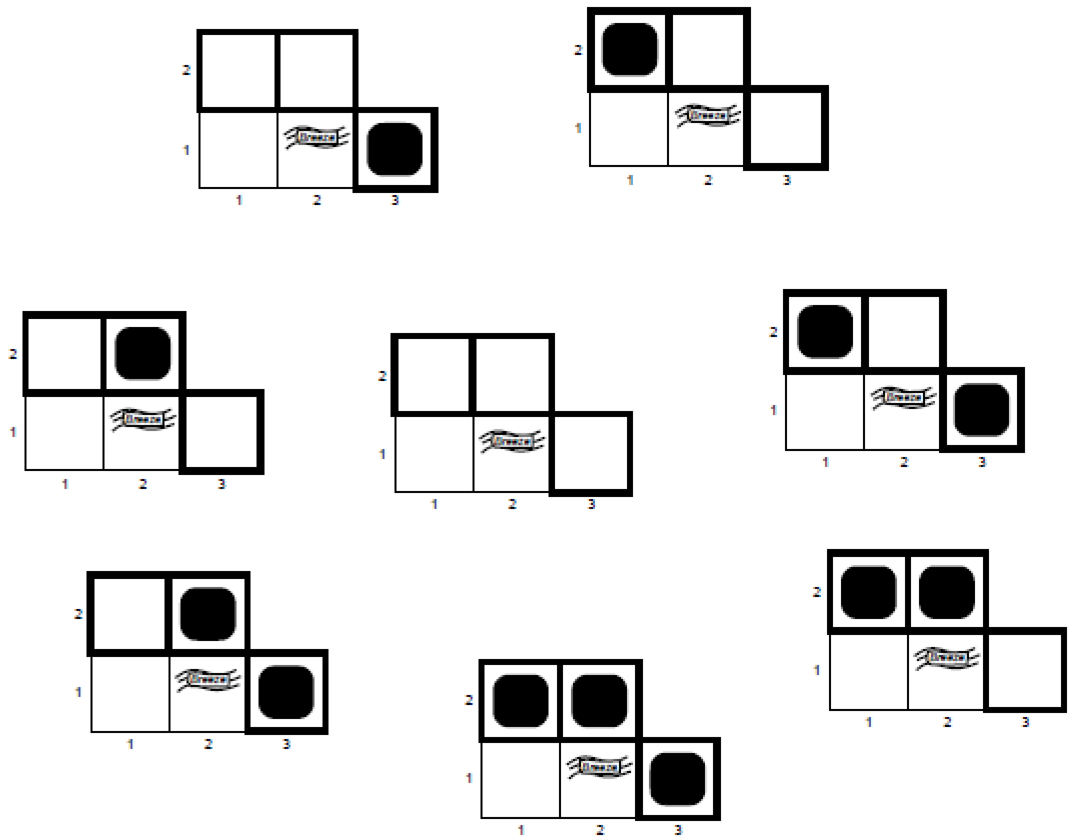


Entailment in the Wumpus World

- Situation after detecting nothing in [1,1], moving right, breeze in [2,1]
- Consider possible models for ?s
- assuming only pits, 3 Boolean choices => 8 possible models

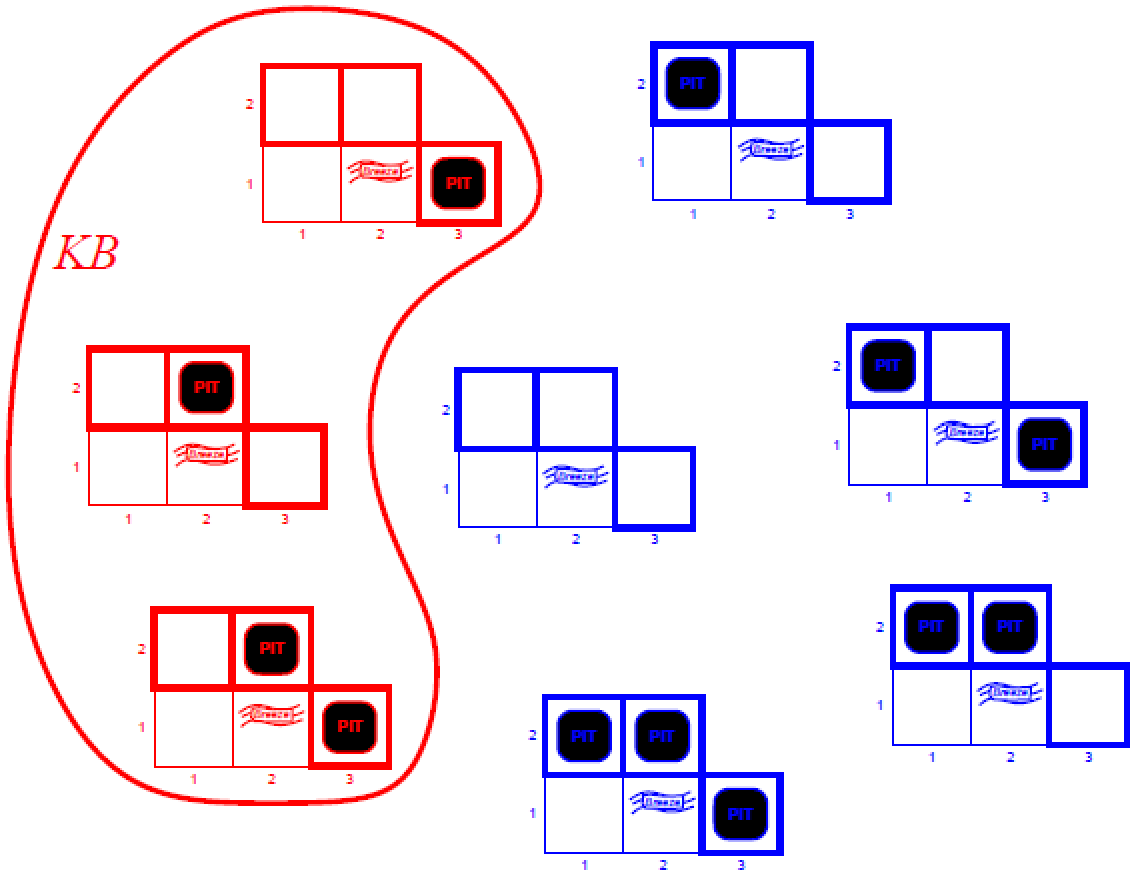


Wumpus Models



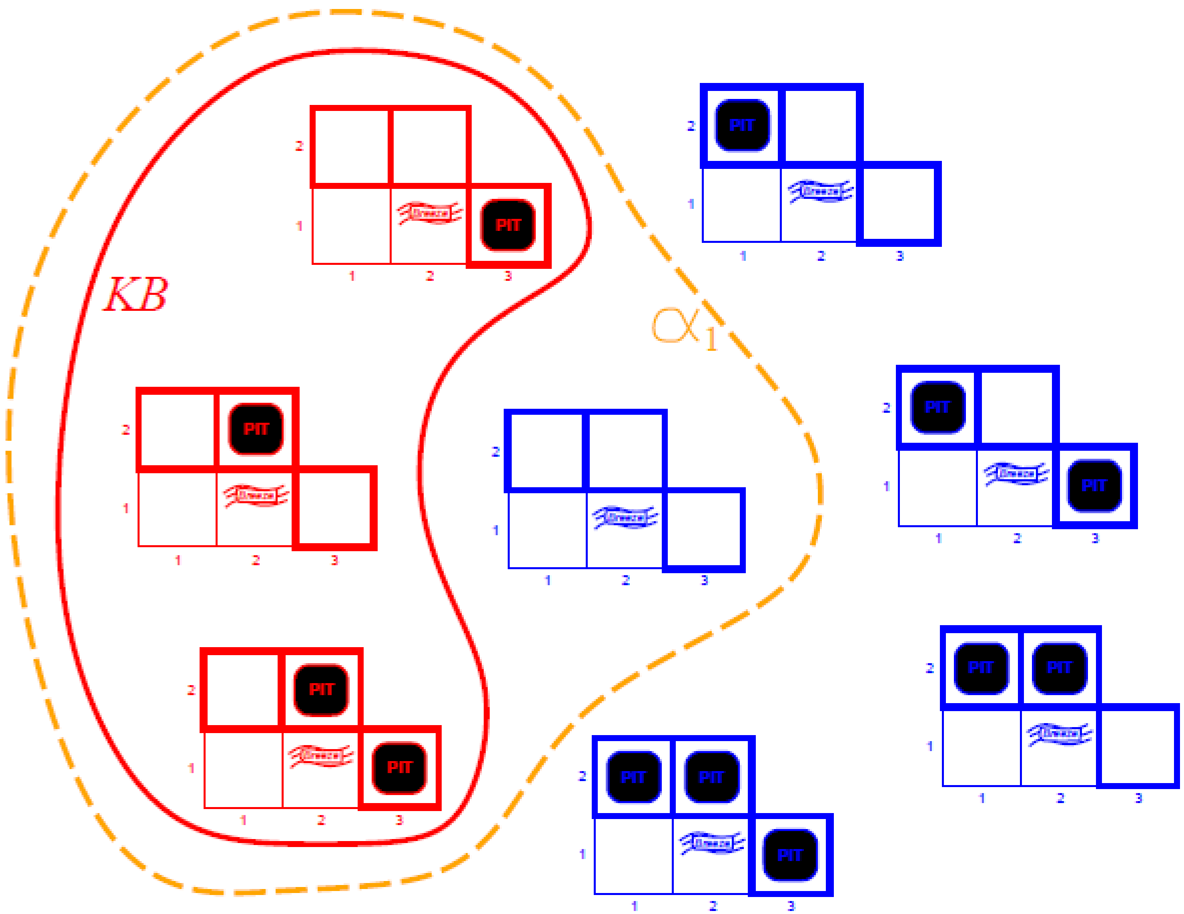
Wumpus Models

KB = wumpus-world rules + observations



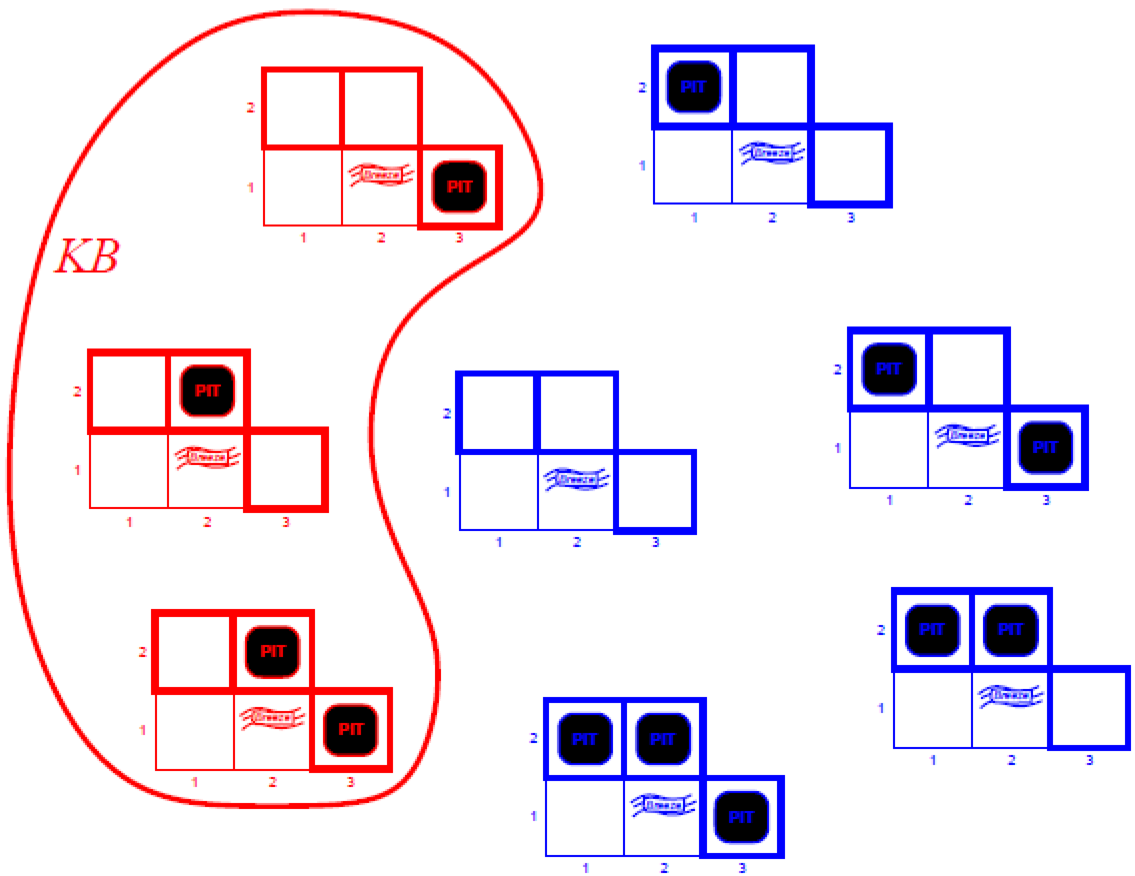
- KB = wumpus-world rules + observations
- α1 = “[1,2] is safe”, KB |= α1, proved by model checking

Wumpus Models



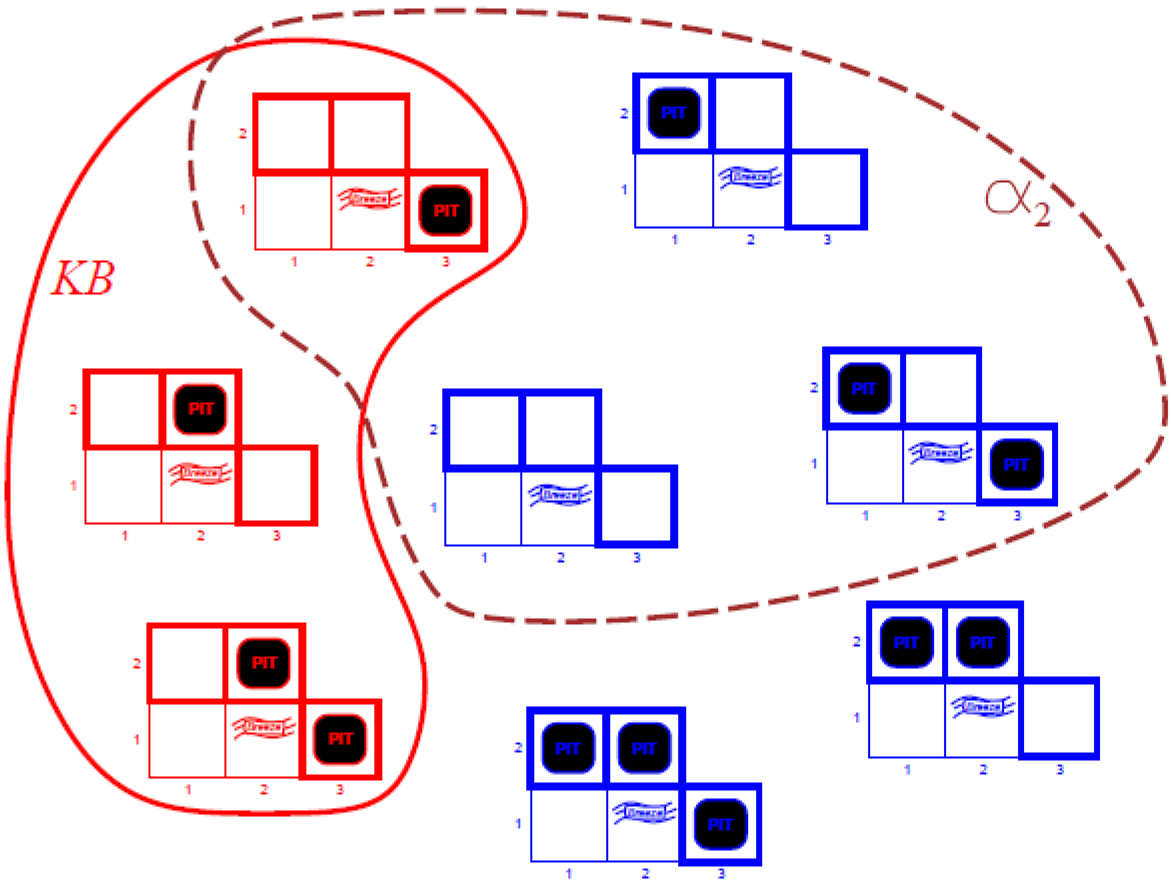
Wumpus Models

KB = wumpus-world rules + observations



Wumpus models

- KB = wumpus-world rules + observations
- α2 = “[2,2] is safe”, KB |≠ α2



Inference

$KB \vdash_i \alpha$

- means 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.

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 S1 and S2 are sentences, $S1 \wedge S2$ is a sentence (conjunction)
- If S1 and S2 are sentences, $S1 \vee S2$ is a sentence (disjunction)
- If S1 and S2 are sentences, $S1 \Rightarrow S2$ is a sentence (implication)
- If S1 and S2 are sentences, $S1 \Leftrightarrow S2$ is a sentence (biconditional)

Propositional Logic: Semantics

- Each model species true/false for each proposition symbol
- E.g. $\begin{matrix} P_{1,2} & P_{2,2} & P_{3,1} \\ true & true & false \end{matrix}$
- (With these symbols, 8 possible models, can be enumerated automatically.)
- Rules for evaluating truth with respect to a model m:

$\neg S$	is true iff	S	is false		
$S_1 \wedge S_2$	is true iff	S_1	is true **and**	S_2	is true
$S_1 \vee S_2$	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

$P_{1,2} \wedge (\neg P_{2,2} \vee \neg P_{3,1}) = true \wedge (false \vee true) = true \wedge true = true$

- Simple recursive process evaluates an arbitrary sentence

Truth Tables for Connectives

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*

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].
  - $\neg P_{1,1}$
  - $\neg B_{1,1}$
  - $B_{2,1}$
- "Pits cause breezes in adjacent squares"
  - $B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})$
  - $B_{2,1} \Leftrightarrow (P_{1,1} \lor P_{2,2} \lor P_{3,1})$
- "A square is breezy if and only if there is an adjacent pit"

Truth Tables for Inference

- Enumerate rows (different assignments to symbols), if KB is true in row, check that α is too

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	*false*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*false*	*false*
false	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*false*	*true*	*false*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*false*	*true*	*true*	*false*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

Inference by Enumeration

- Depth-first enumeration of all models is sound and complete
- $O(2^n)$ for n symbols; problem is co-NP-complete

```
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, α, symbols, [ ])
-----------------------------------------------------------------
function TT-CHECK-ALL(KB, α, symbols, model) returns true or false
   if EMPTY?(symbols) then
      if PL-TRUE?(KB, model) then return PL-TRUE?(α, model)
      else return true
   else do
      P ← FIRST(symbols); rest ← REST(symbols)
      return TT-CHECK-ALL(KB, α, rest, EXTEND(P, true, model)) and
             TT-CHECK-ALL(KB, α, rest, EXTEND(P, false, model))
```

Logical Equivalence

- Two sentences are logically equivalent if true in same models:

$$\alpha \equiv \beta$$

- if and only if

$$\alpha \models \beta$$

- and

$$\beta \models \alpha$$

$$
\begin{array}{rcll}
(\alpha \land \beta) & \equiv & (\beta \land \alpha) & \text{commutativity of } \land \\
(\alpha \lor \beta) & \equiv & (\beta \lor \alpha) & \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) & \equiv & (\alpha \land (\beta \land \gamma)) & \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) & \equiv & (\alpha \lor (\beta \lor \gamma)) & \text{associativity of } \lor \\
\neg(\neg\alpha) & \equiv & \alpha & \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg\beta \Rightarrow \neg\alpha) & \text{contraposition} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg\alpha \lor \beta) & \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) & \equiv & ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) & \text{biconditional elimination} \\
\neg(\alpha \land \beta) & \equiv & (\neg\alpha \lor \neg\beta) & \text{De Morgan} \\
\neg(\alpha \lor \beta) & \equiv & (\neg\alpha \land \neg\beta) & \text{De Morgan} \\
(\alpha \land (\beta \lor \gamma)) & \equiv & ((\alpha \land \beta) \lor (\alpha \land \gamma)) & \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) & \equiv & ((\alpha \lor \beta) \land (\alpha \lor \gamma)) & \text{distributivity of } \lor \text{ over } \land
\end{array}
$$

Validity and Satisfiability

- A sentence is valid if it is true in all models,
  - e.g., True, $A \vee \neg A$, $A \Rightarrow A$, $(A \wedge (A \Rightarrow B)) \Rightarrow B$
- 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., $A \vee B$, C
- A sentence is unsatisfiable if it is true in no models
  - e.g., $A \wedge \neg A$
- Satisfiability is connected to inference via the following:
  - $KB \models \alpha$ if and only if $(KB \wedge \neg\alpha)$ is unsatisfiable
- i.e., prove by reductio ad absurdum