knowledge-based agent

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

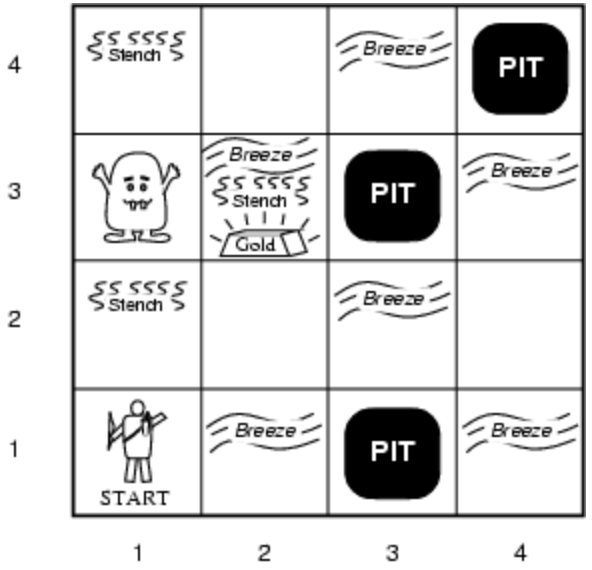
```
function KB-AGENT(percept) returns an action
    static: KB, a knowledge base
            t, a counter, initially 0, indicating time

    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

Wumpus World PEAS description

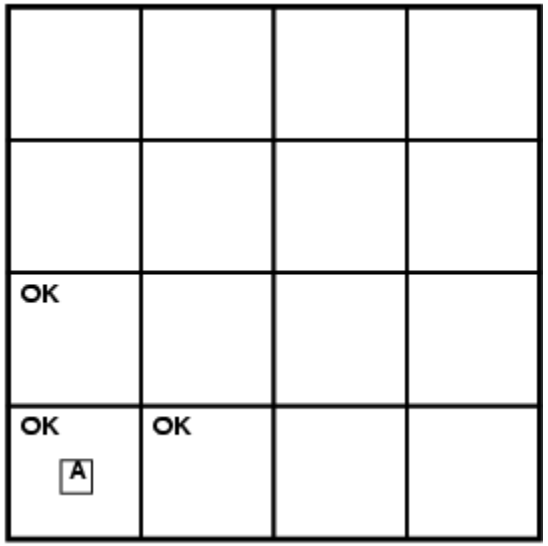
- Performance measure
  - gold +1000, death -1000
  - -1 per step, -10 for using the arrow
- Environment
-
  - Squares adjacent to wumpus are smelly
  -
  - Squares adjacent to pit are breezy
  -
  - Glitter iff gold is in the same square
  -
  - Shooting kills wumpus if you are facing it
  -
  - Shooting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



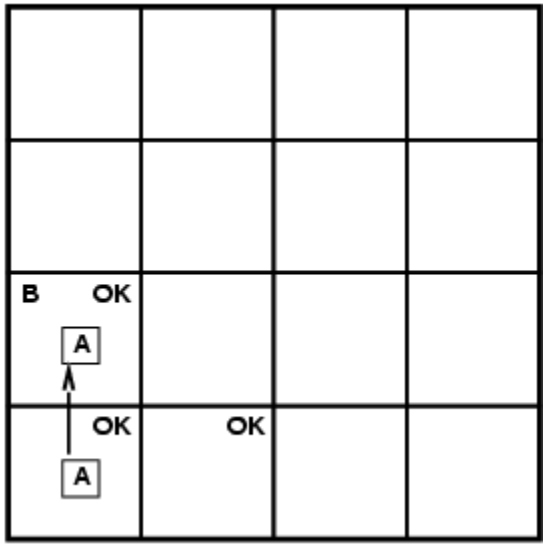
Wumpus world characterization

- Fully 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
-
- Single-agent? Yes – Wumpus is essentially a natural feature

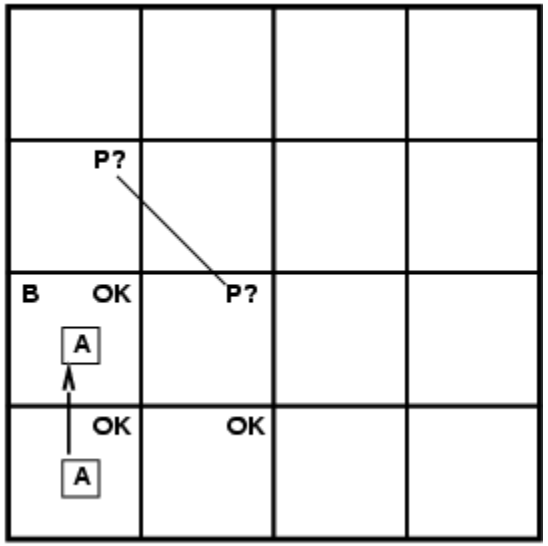
Exploring a wumpus world



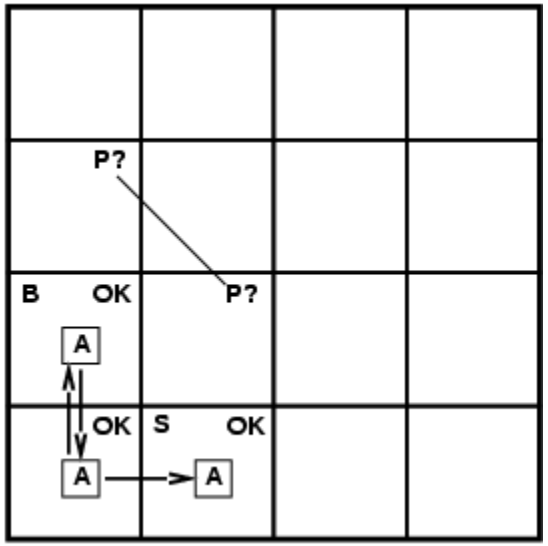
Exploring a wumpus world



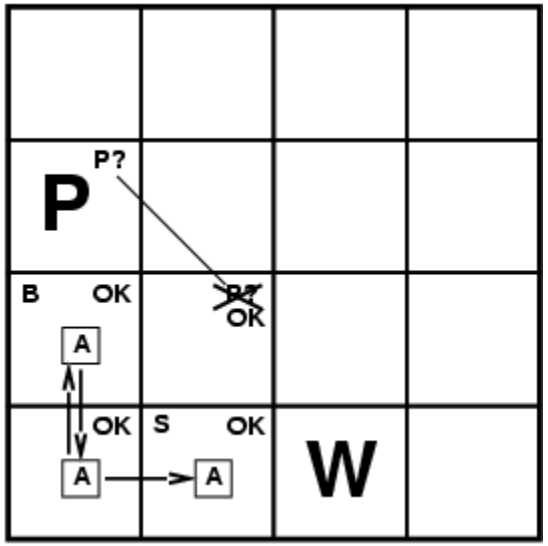
Exploring a wumpus world



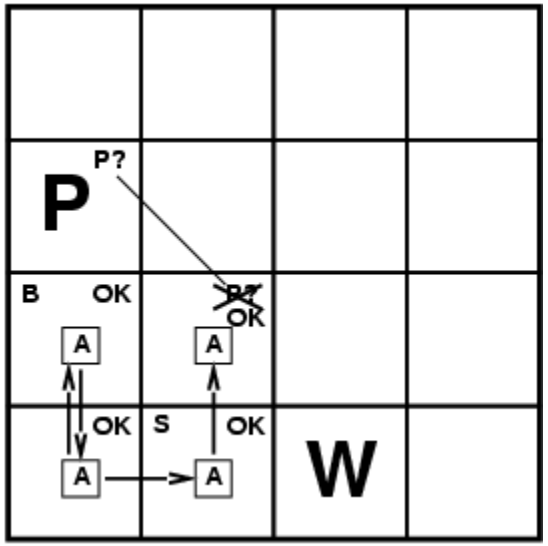
Exploring a wumpus world



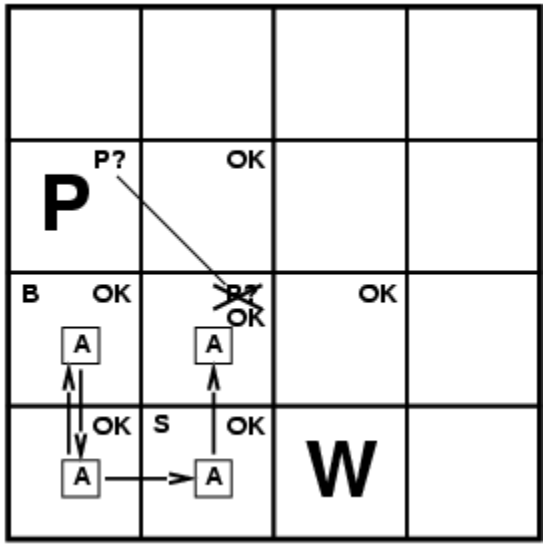
Exploring a wumpus world



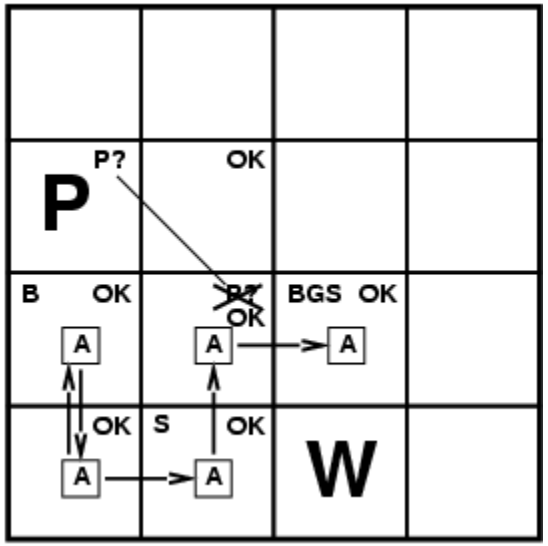
Exploring a wumpus world



Exploring a wumpus world



Exploring a wumpus world



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
  -

Formal languages

Language	Ontology	Epistemology
1- certain languages		
Propositional logic	Facts	True/False/Unknown
First-order logic	Facts, objects, relations	True/False/Unknown
Temporal logic	Facts,objects,relations,time	True/False/Unknown
2- uncertainty languages		
Probability theory	Facts	Degree of belief [0,1]
Fuzzy logic	Facts	Degree of truth [0,1]

Entailment

- Entailment means that one thing follows from another:
-

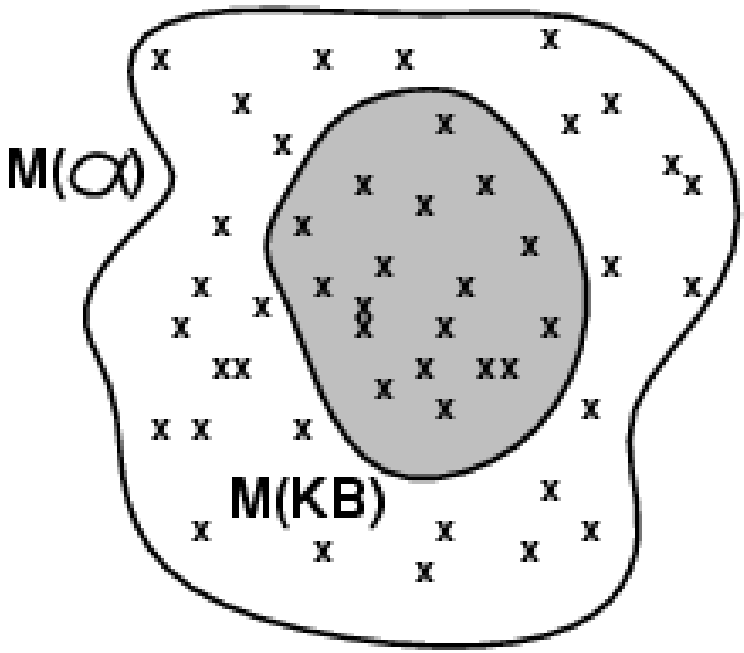
$$KB \models \alpha$$

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

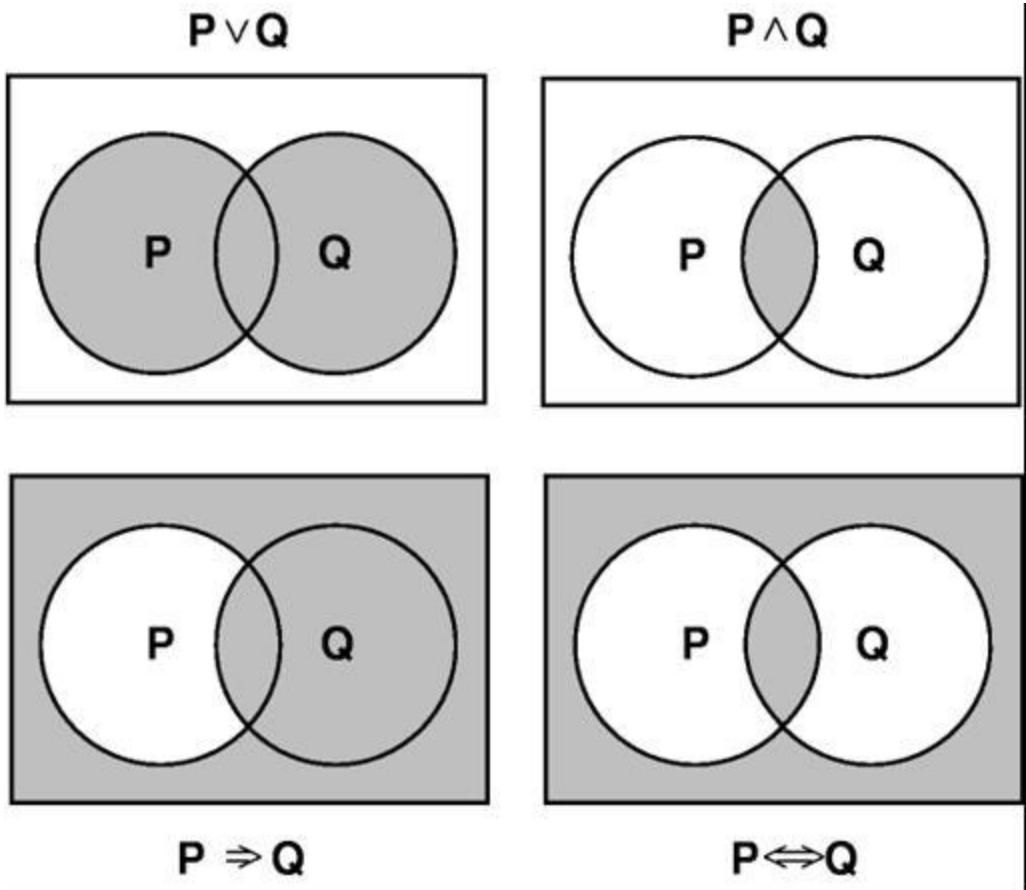
Models

- 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 ⊨ α iff *M(KB)* ⊆ *M(α)*
-
  - E.g. *KB* = Giants won α = Giants won
  -



Models

- We could define the meaning of a sentence by means of set operations on sets of models.
- For example, the set of models of $P \wedge Q$ is the intersection of models of P and models of Q.

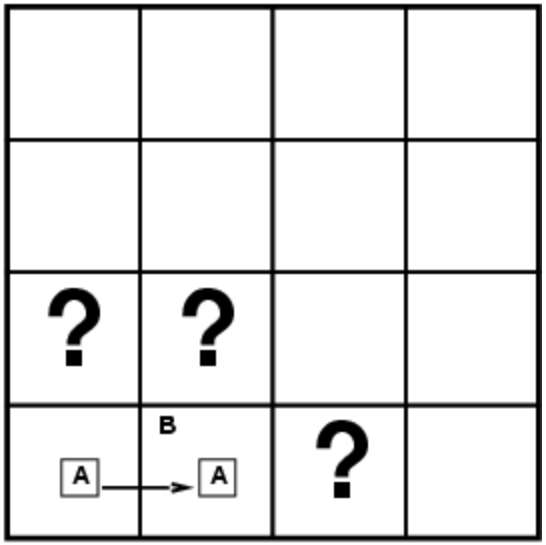


Entailment in the wumpus world

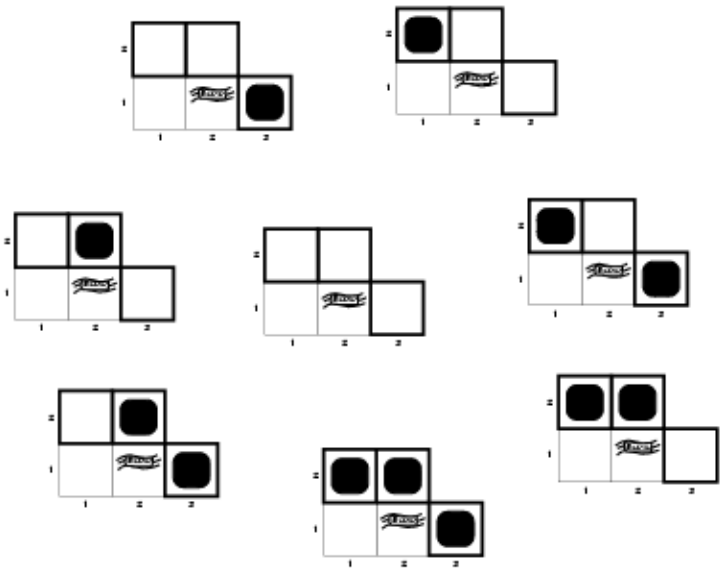
Situation after detecting nothing in [1,1], moving right, breeze in [2,1]

Consider possible models for *KB* assuming only pits

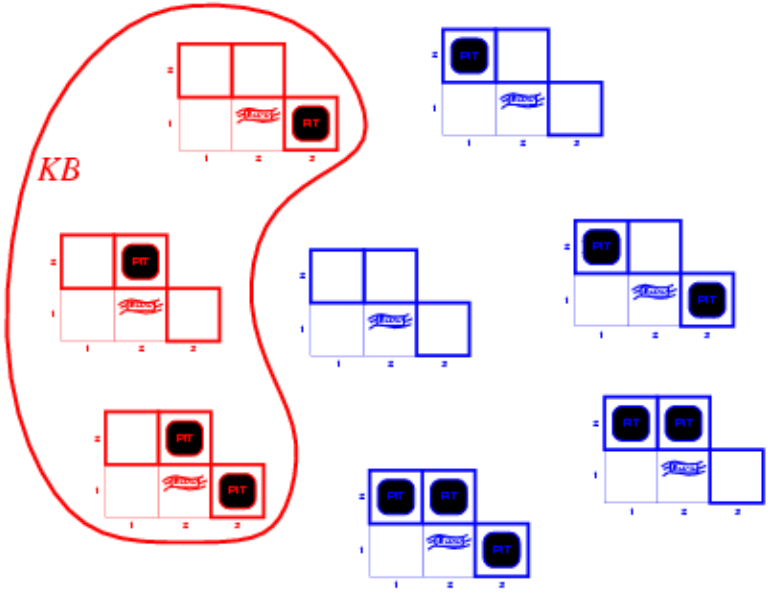
3 Boolean choices $\Rightarrow$ 8 possible models



Wumpus models

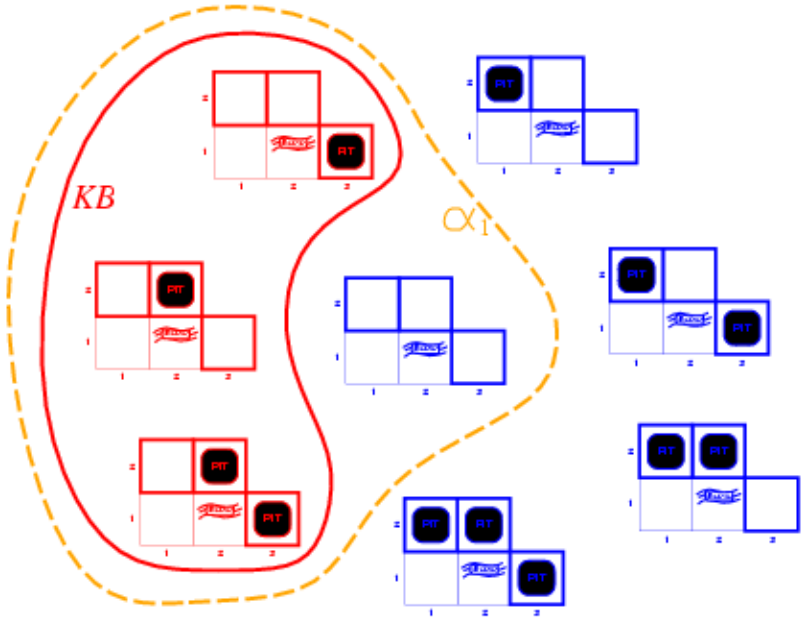


Wumpus models



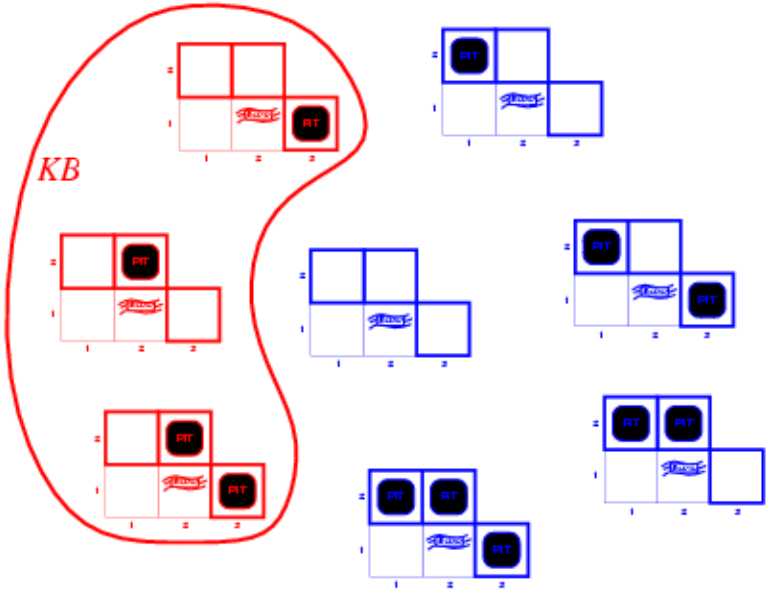
- *KB* = wumpus-world rules + observations
-

Wumpus models



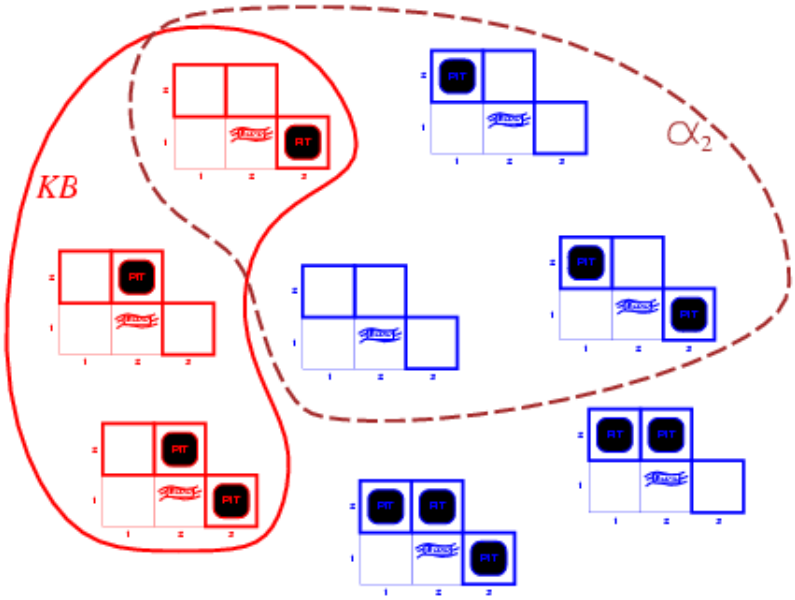
- *KB* = wumpus-world rules + observations
- α_1 = "[1,2] is safe", $KB \models \alpha_1$, proved by model checking
-

Wumpus models



- *KB* = wumpus-world rules + observations

Wumpus models



- *KB* = wumpus-world rules + observations
- α_2 = "[2,2] is safe", $KB \not\models \alpha_2$
-

Inference

- $KB \vdash_i \alpha$ = sentence α can be derived from *KB* by procedure *i*
-
- 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 $KB \vdash_i \alpha$
-
- 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 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: Syntax

$$
\begin{aligned}
\textit{Sentence} &\rightarrow \textit{AtomicSentence} \mid \textit{ComplexSentence} \\
\textit{AtomicSentence} &\rightarrow \textbf{\textit{True}} \mid \textbf{\textit{False}} \mid \textit{Symbol} \\
\textit{Symbol} &\rightarrow \textbf{\textit{P}} \mid \textbf{\textit{Q}} \mid \textbf{\textit{R}} \mid \ldots \\
\textit{ComplexSentence} &\rightarrow \neg\, \textit{Sentence} \\
&\mid (\, \textit{Sentence} \wedge \textit{Sentence}\,) \\
&\mid (\, \textit{Sentence} \vee \textit{Sentence}\,) \\
&\mid (\, \textit{Sentence} \Rightarrow \textit{Sentence}\,) \\
&\mid (\, \textit{Sentence} \Leftrightarrow \textit{Sentence}\,)
\end{aligned}
$$

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.

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

Simple recursive process evaluates an arbitrary sentence, e.g.,

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"
-

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

Logical equivalence

- Two sentences are logically equivalent} iff true in same models: α ≡ ß iff α ╞ β and β ╞ α
-
-

$$
\begin{aligned}
(\alpha \wedge \beta) &\equiv (\beta \wedge \alpha) && \text{commutativity of } \wedge \\
(\alpha \vee \beta) &\equiv (\beta \vee \alpha) && \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) &\equiv (\alpha \wedge (\beta \wedge \gamma)) && \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) &\equiv (\alpha \vee (\beta \vee \gamma)) && \text{associativity of } \vee \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \vee \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) &\equiv (\neg\alpha \vee \neg\beta) && \text{de Morgan} \\
\neg(\alpha \vee \beta) &\equiv (\neg\alpha \wedge \neg\beta) && \text{de Morgan} \\
(\alpha \wedge (\beta \vee \gamma)) &\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) && \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) &\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) && \text{distributivity of } \vee \text{ over } \wedge
\end{aligned}
$$

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