

Introduction to Artificial Intelligence

Lecture 7 – Logical reasoning

CS/CNS/EE 154
Andreas Krause

Announcements

- Project milestone code out
- Due Nov 3

Logical agents

- Want rational agents that perform well in variety of environments
- Engineering principle:
 - Decouple problem specific properties and problem independent algorithms
- Logical agents use
 - **formal languages** that allow to succinctly represent many different environments
 - **knowledge base** to encode **problem-specific** known facts
 - **problem independent inference algorithms** to deduce new facts

Example: Wumpus world (PEAS)

- *Performance measure*

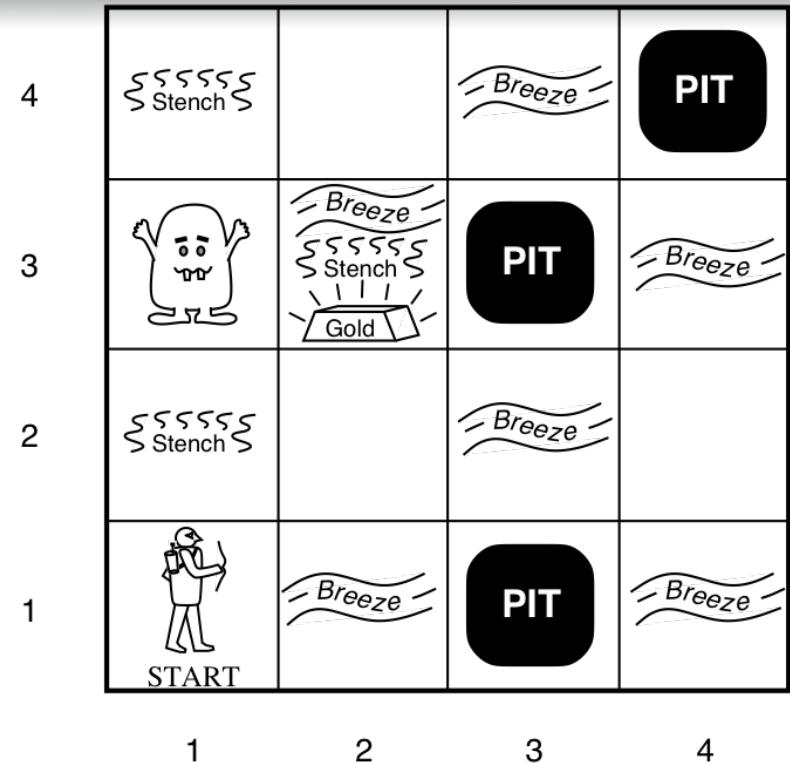
- Gold +1000; death -1000
- -1 per step; -10 for arrow

- *Environment*

- Squares adj. to wumpus smelly
- Squares adj. to pit are breezy
- Glitter if gold on same square
- Shooting kills wumpus if facing it
- Shooting uses up only arrow
- Grabbing picks up gold if in same square
- Releasing drops gold in same square
- Dead if eaten by wumpus or fallen into pit

- *Actions:* Turn left, right; Forward; Grab; Release; Shoot

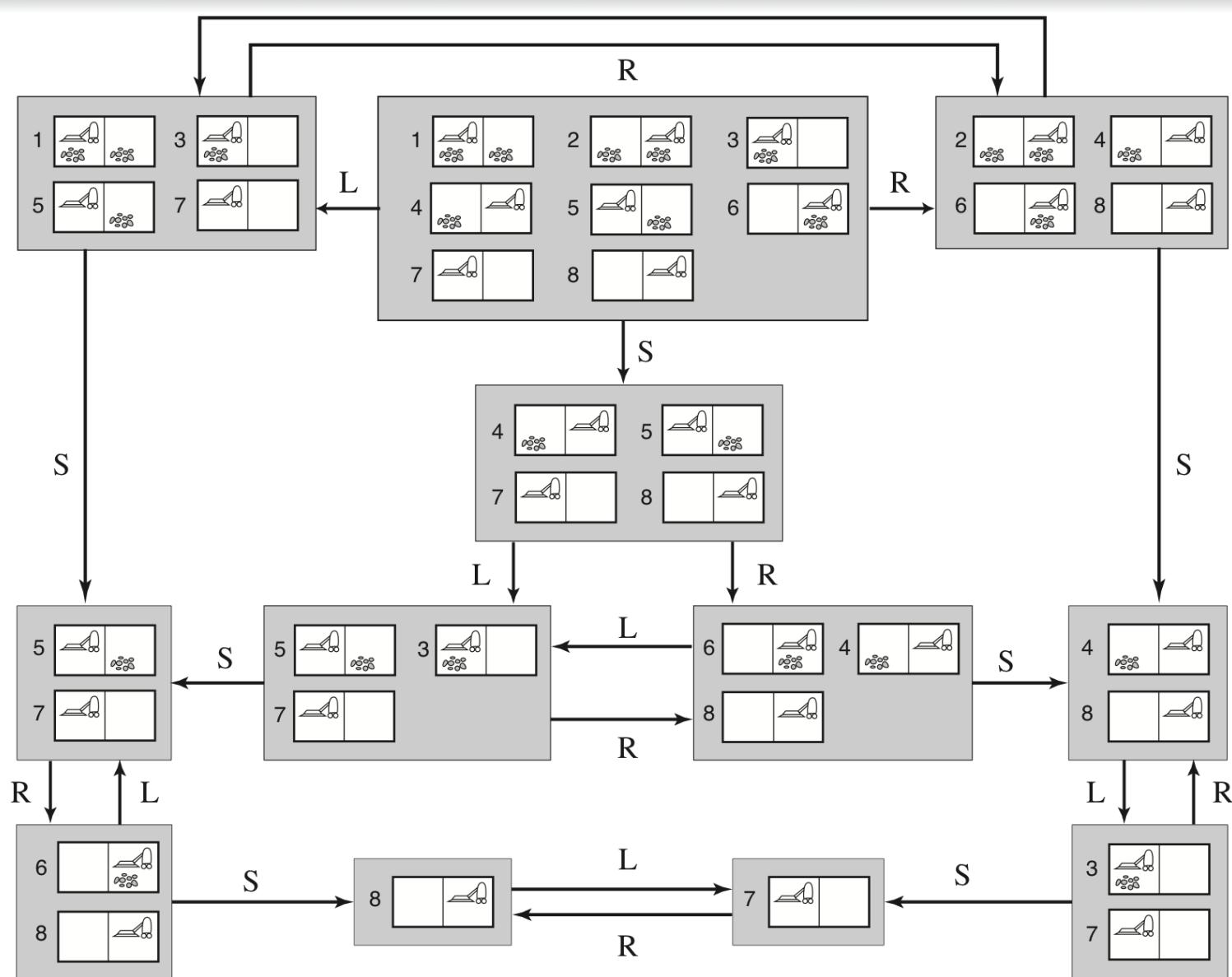
- *Sensors:* Breeze; Glitter; Smell



Wumpus world properties

- Observable? No (partially observable)
- Deterministic? Yes
- Episodic? No (sequential)
- Static? Yes (wumpus doesn't move)
- Discrete? Yes
- Single agent? Yes (wampus is a natural feature)

Reminder: Planning in belief space



Wumpus world as search problem

- Partially observable → Need to plan in belief state
- Number of states:

$$\binom{16}{3} \cdot 13 \cdot 12 \cdot 11 \cdot 4 \cdot 3 \approx 10 \cdot 10^6$$

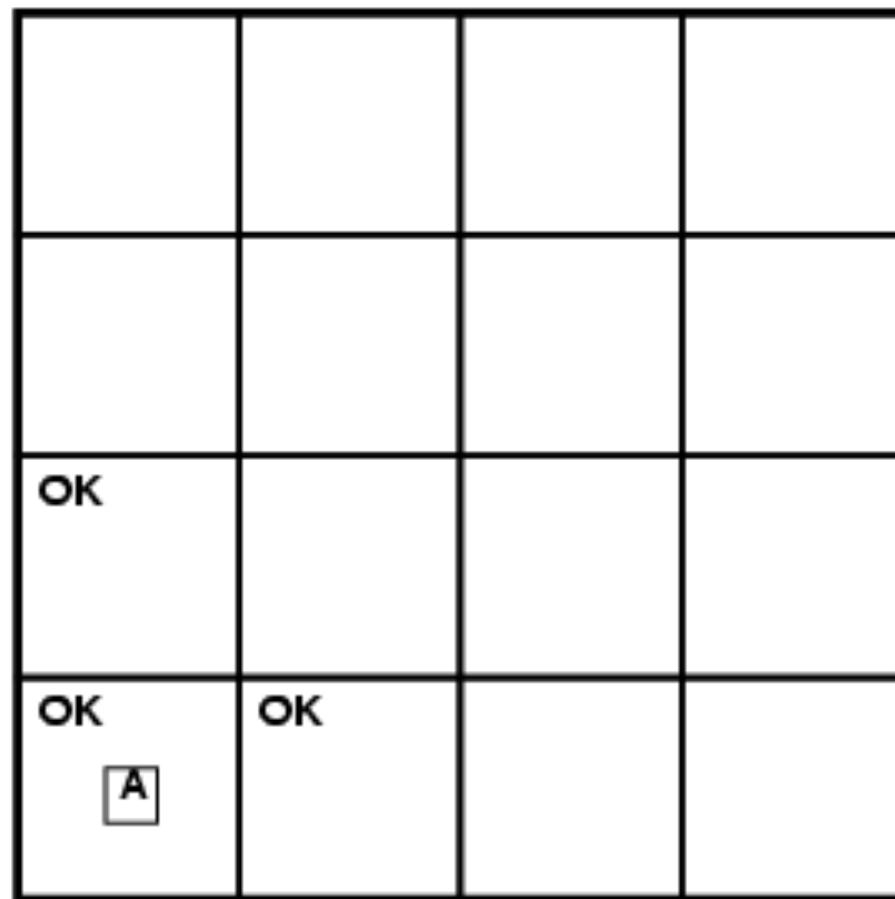
$\uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow$
Pits Gold Wumpus Agent Dir. Arrow/Wumpus health

belief states : $\approx 2^{10^7} \approx 10^{3 \cdot 10^6}$

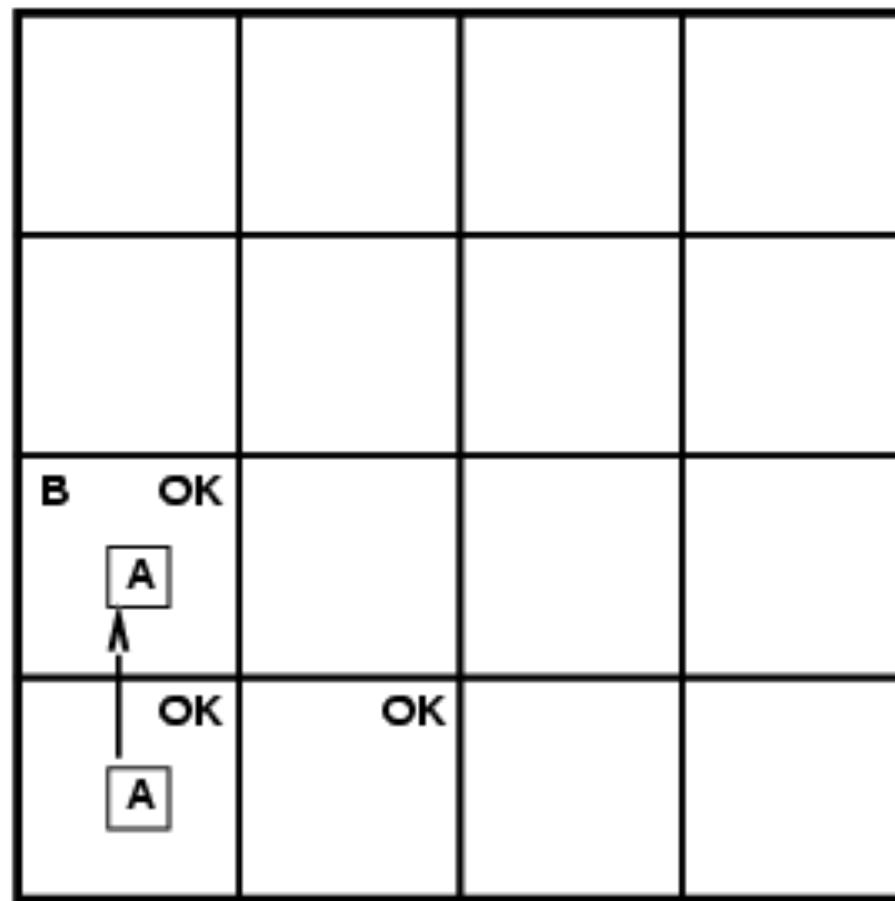
Go : $3^{10^2} \approx 10^{100}$

- Completely intractable!
- Want to implicitly represent state space

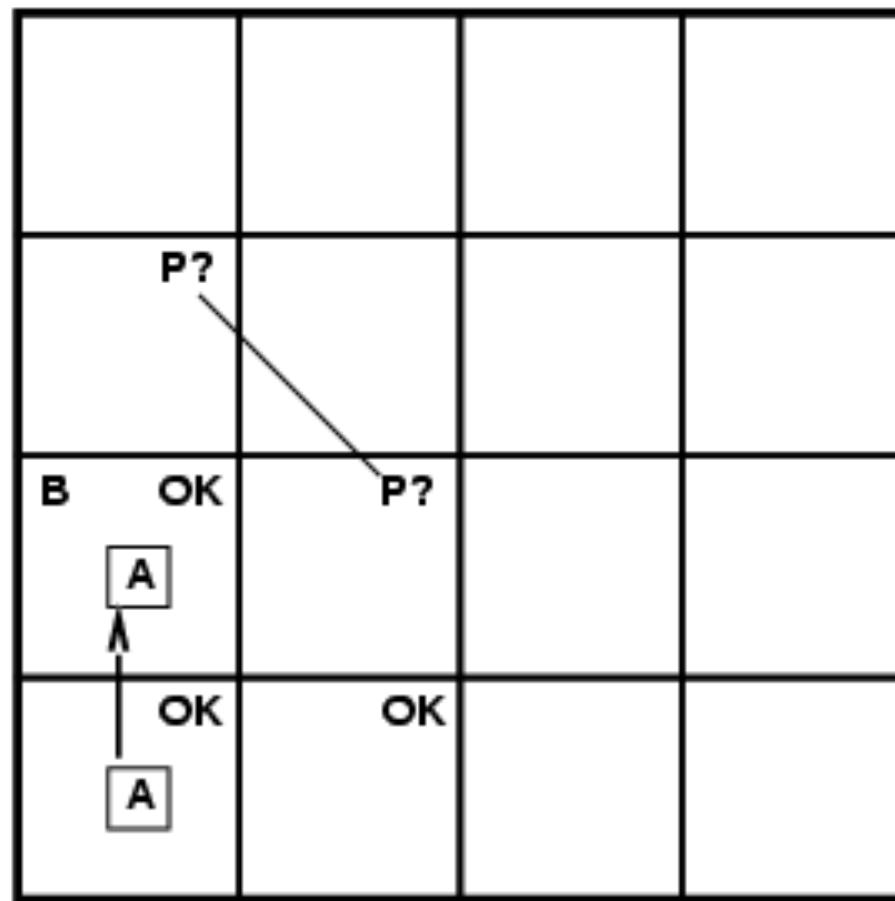
Exploring a Wumpus world



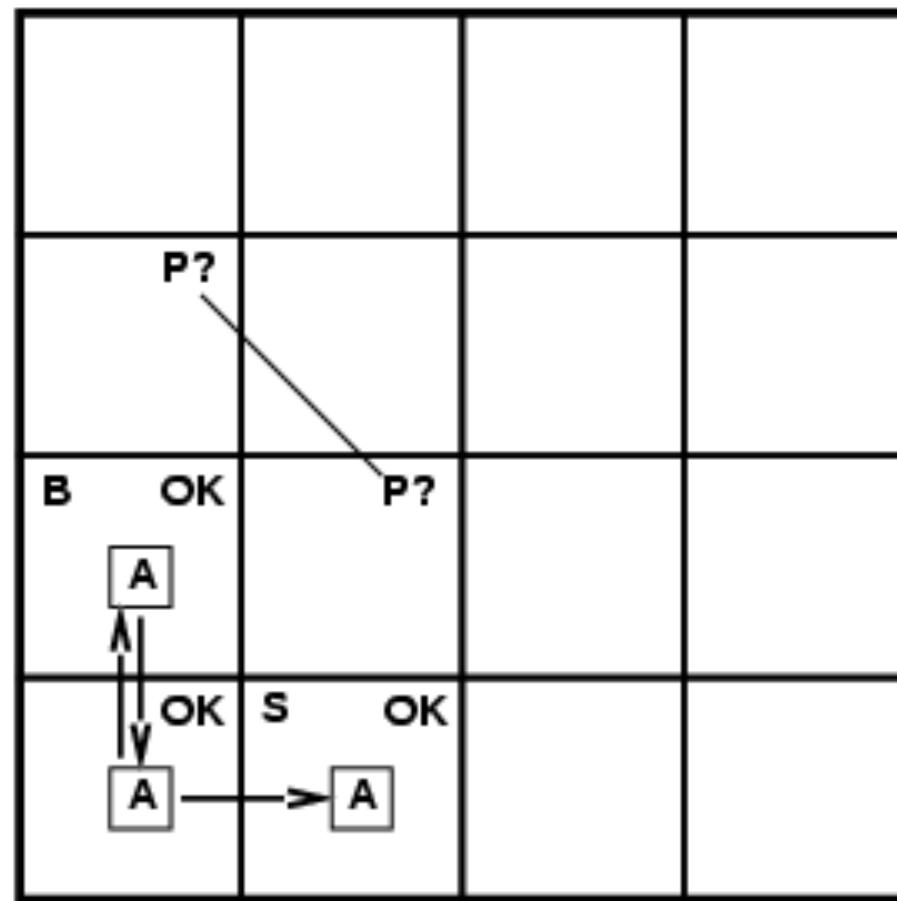
Exploring a Wumpus world



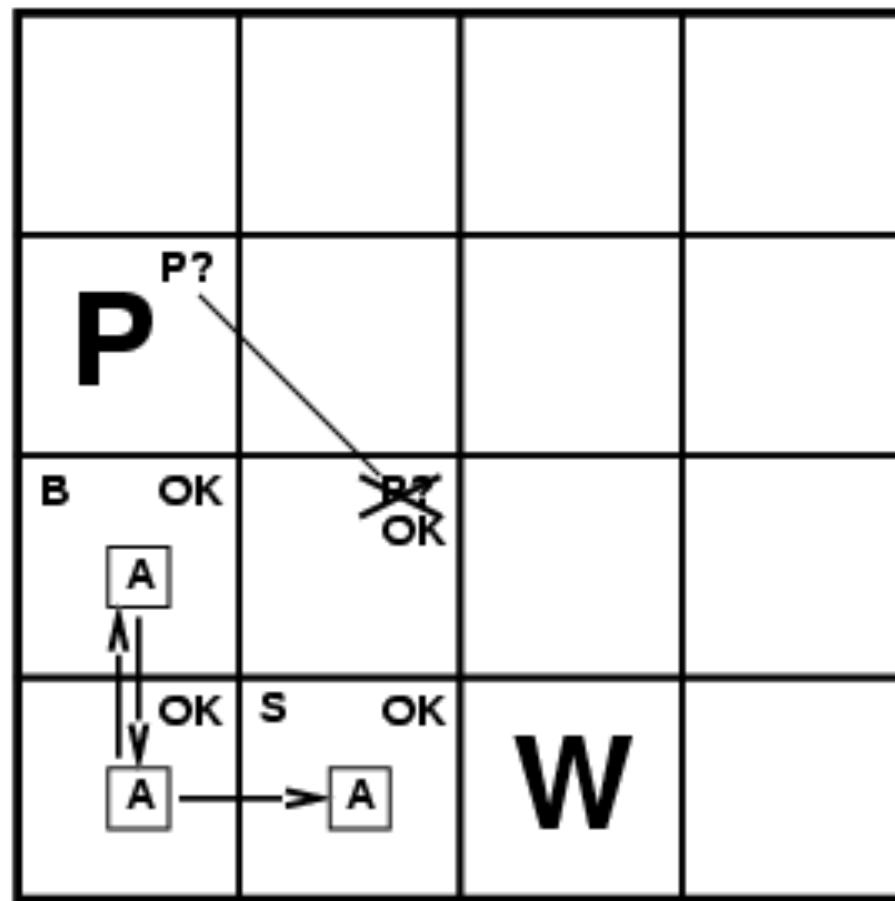
Exploring a Wumpus world



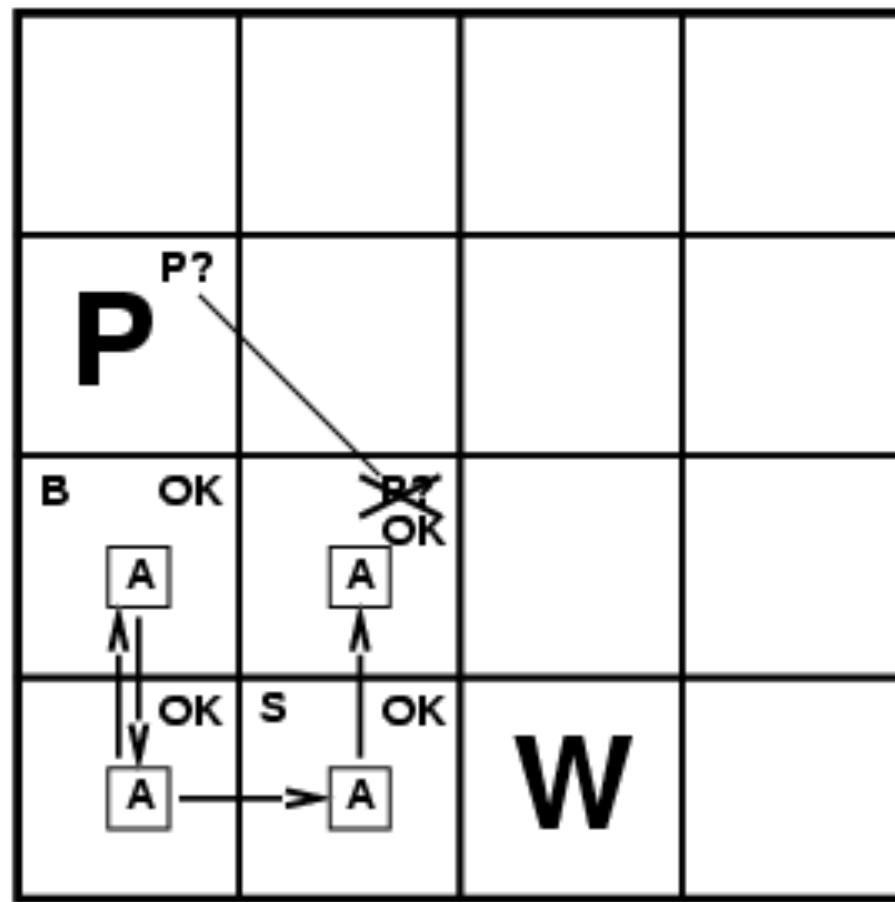
Exploring a Wumpus world



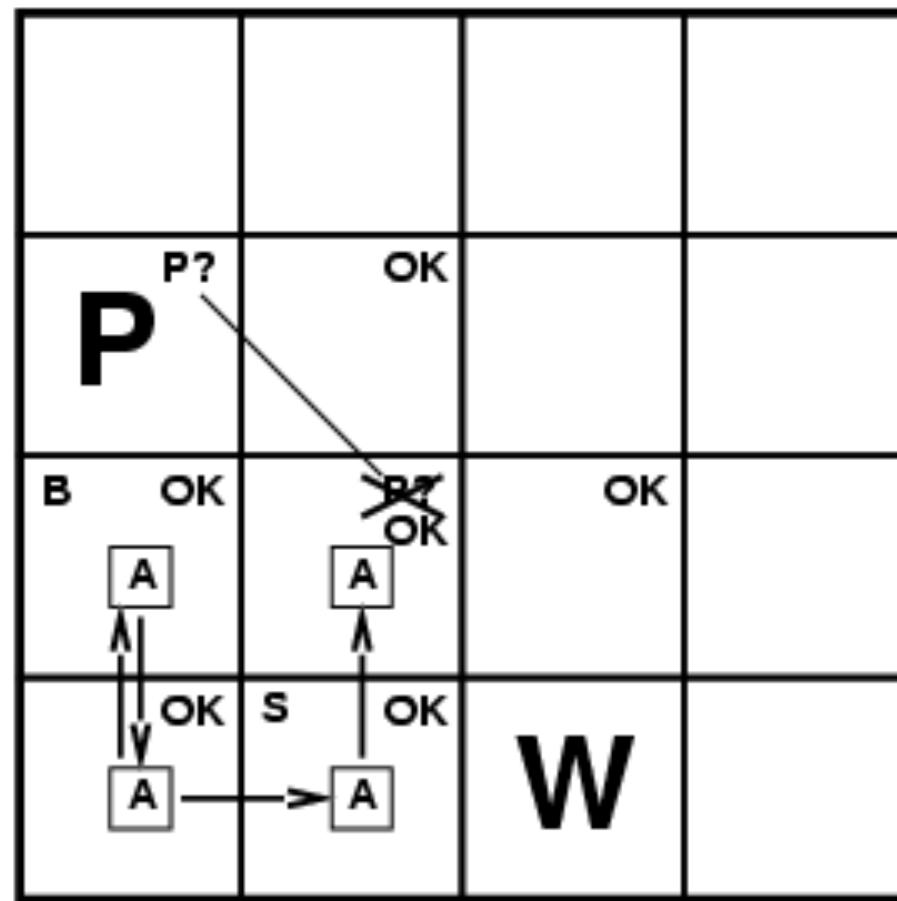
Exploring a Wumpus world



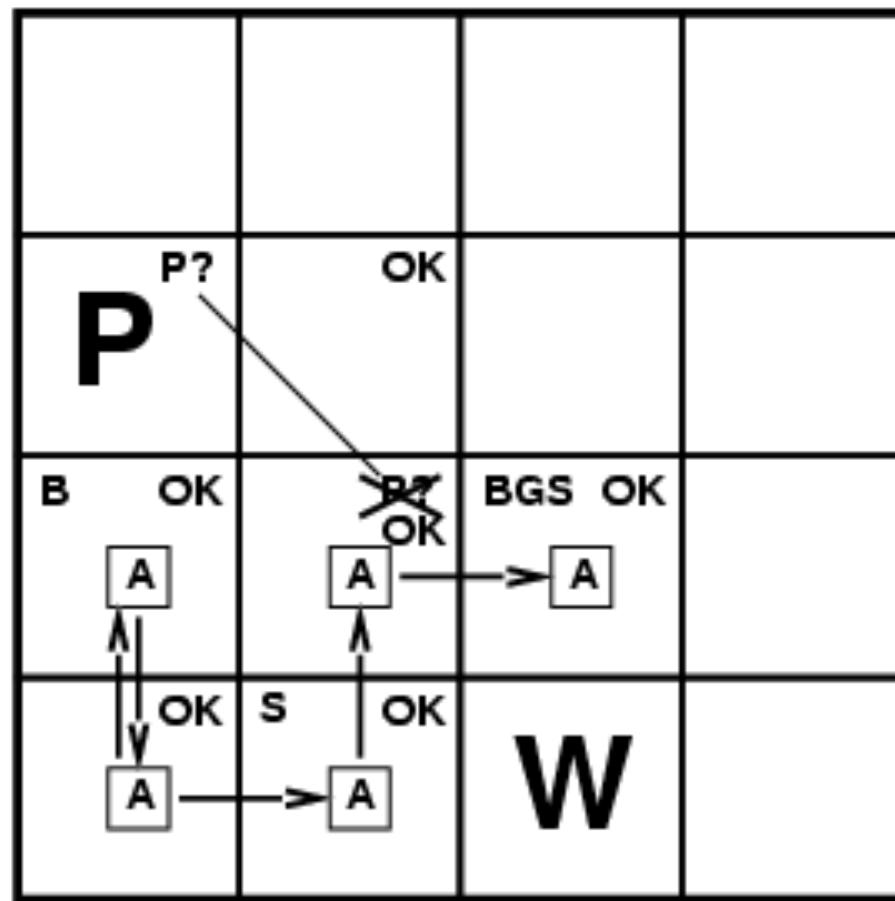
Exploring a Wumpus world



Exploring a Wumpus world



Exploring a Wumpus world



Logics 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, i.e., the truth of a sentence in a world (environment state)
- Example: Language of arithmetic

$3 + * 4 = =$	not well-formed
$3 + 4 = 6$	well formed, but false
$3 + x = 6$	true in world $\{(x, 3)\}$ false in world $\{(x, 2)\}$
$3 = 3$	true in world $\{(x, 3), (y, 2)\}$ true in all worlds

Entailment

- Entailment means that one thing follows from another

$$KB \models \alpha$$

- Knowledge base KB entails sentence α

if and only

α is true in all worlds where KB is true

- Example: $KB = \{ x = 3 \}$

$$KB \models (x + 2) = 5$$

- Entailment is a semantic relationship between sentences

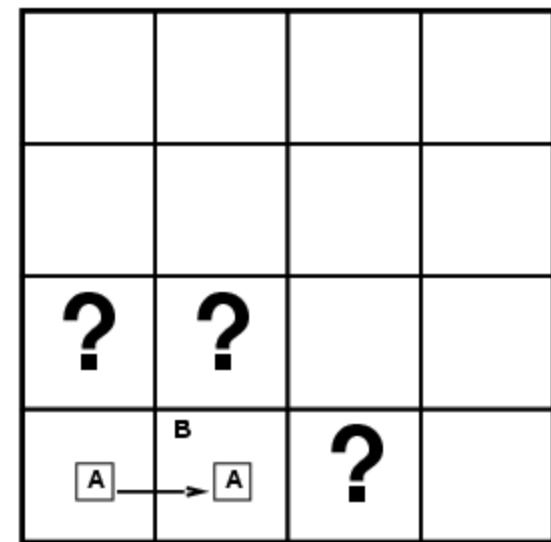
Models

- Logicians think in terms of **models**
 - Formally structured worlds w.r.t. which truth can be evaluated
- We say m is a **model of** a sentence α if α is true in m
 $\alpha = (x + 2 = 5)$ is true in model $m = \{(x, 3)\}$
- $M(\alpha)$ is the set of all models of α
 $\alpha = (x + 2 = y) \quad M(\alpha) = \{ \{(x, 0), (y, 2)\}, \{(x, 3), (y, 5)\}, \dots \}$
- Then $KB \models \alpha$ if and only if
 $M(KB) \subseteq M(\alpha)$

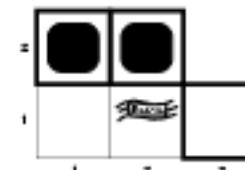
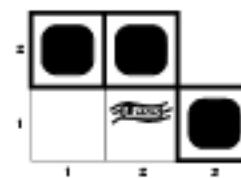
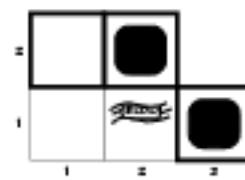
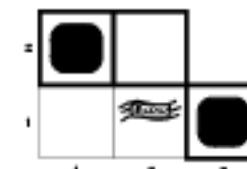
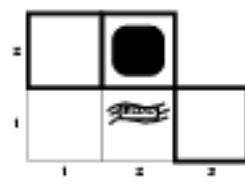
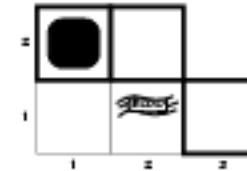
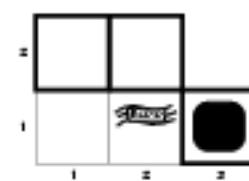
Entailment in the wumpus world

Suppose we observe
nothing in [1,1], moving right,
breeze in [2,1]

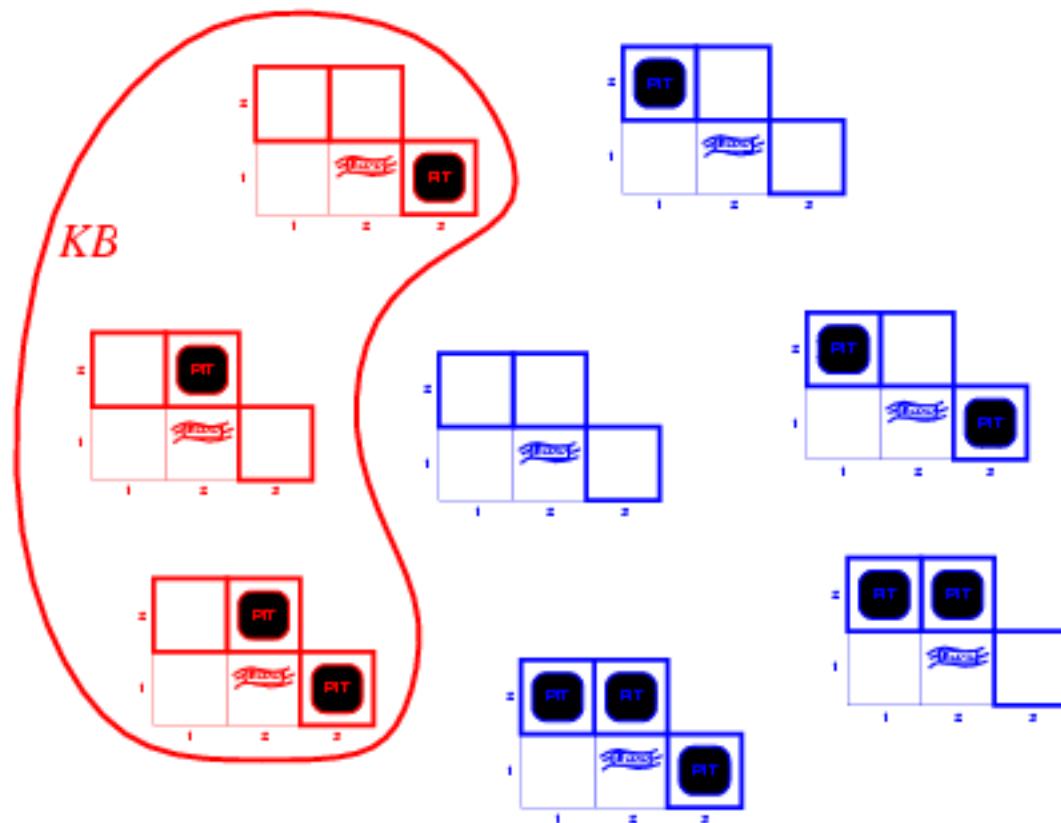
Possible models for KB
(assuming only pits)?



Wumpus models

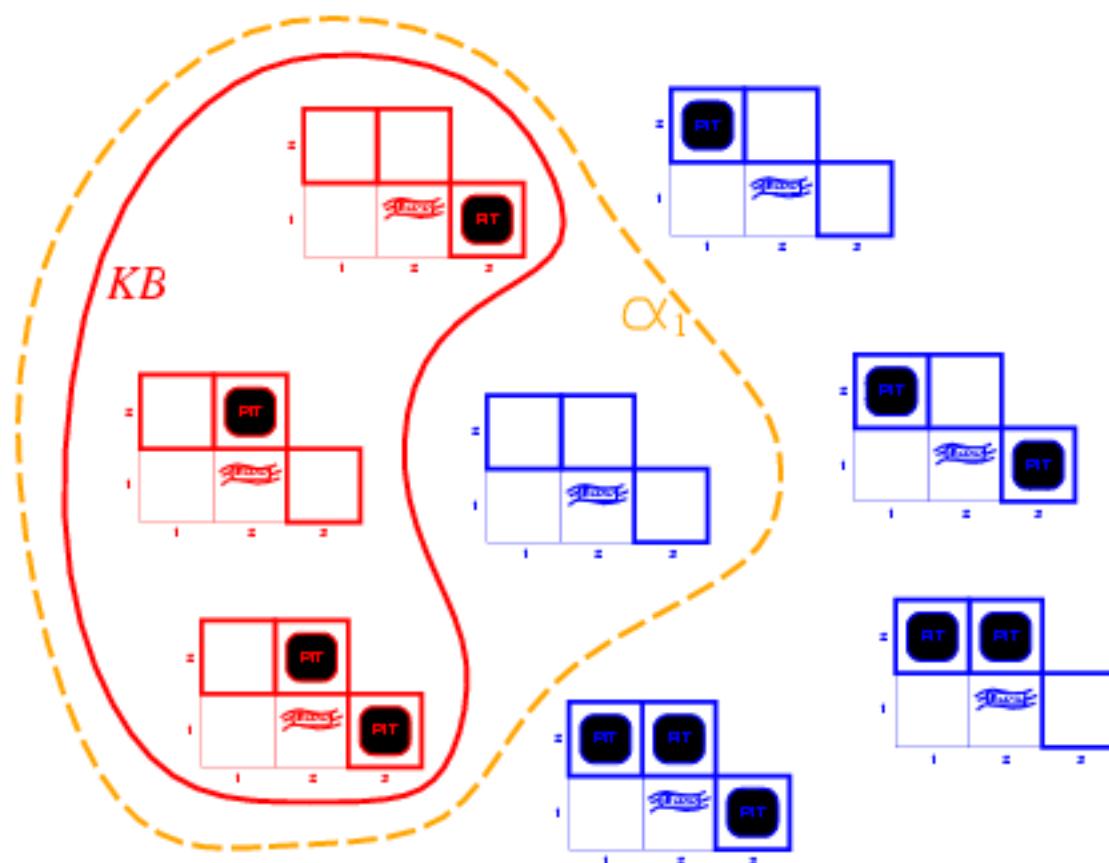


Wumpus models



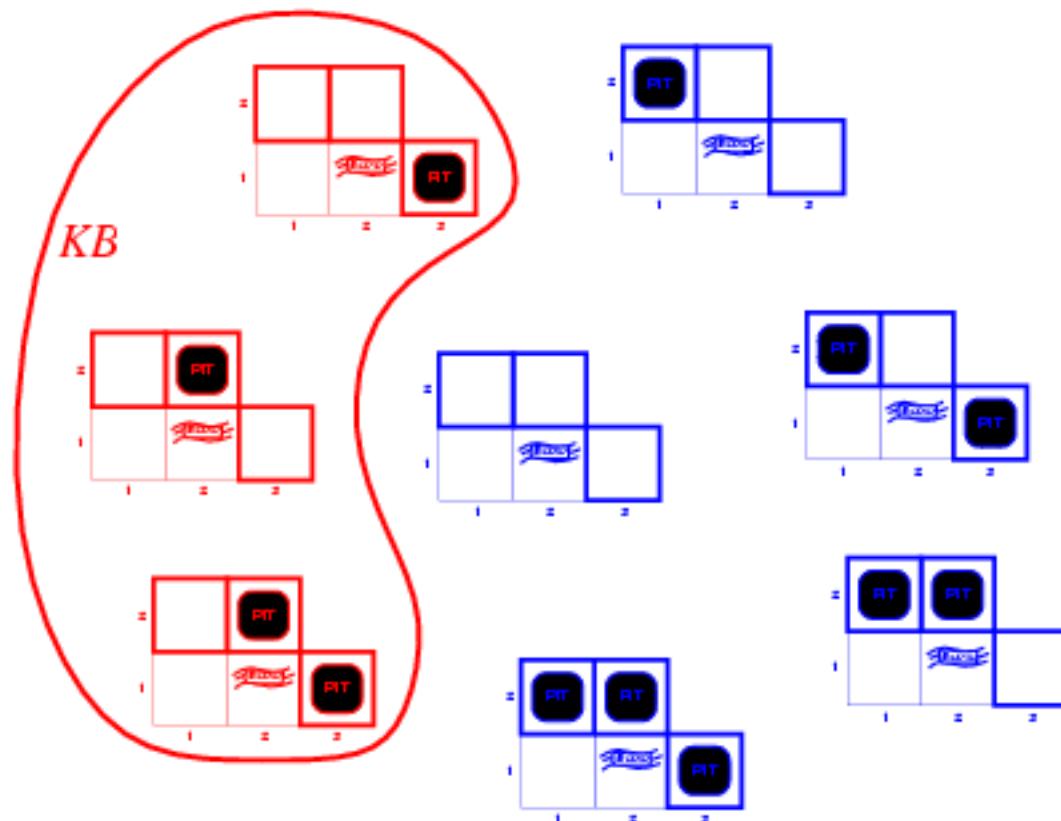
- $KB = \text{wumpus-world rules} + \text{observations}$

Wumpus models



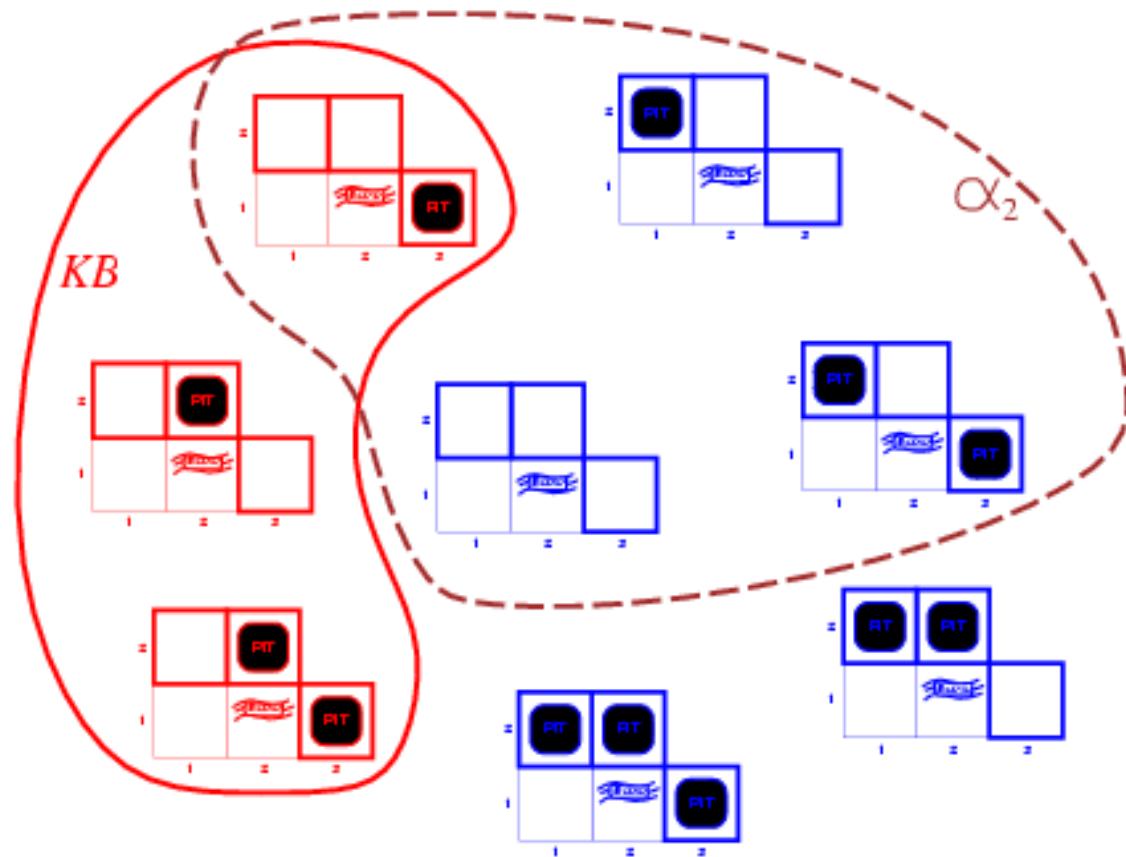
- $KB = \text{wumpus-world rules} + \text{observations}$
- $\alpha_1 = "[1,2] \text{ is safe}", \quad KB \models \alpha_1$

Wumpus models



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

Wumpus models



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

Propositional logic: Syntax

- Simplest example of a logic; illustrates basic ideas
- Propositional symbols 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)
- Notation shorthand:
 - $S_1 \Rightarrow S_2$ for $\neg S_1 \vee S_2$ (implication)
 - $S_1 \Leftrightarrow S_2$ for $(S_1 \Rightarrow S_2) \wedge (S_2 \Rightarrow S_1)$ (biconditional)

Propositional logic: Semantics

Each **model** specifies *true* or *false* for each proposition symbol

E.g.

$P_{1,2}$	$P_{2,2}$	$P_{3,1}$
false	true	false

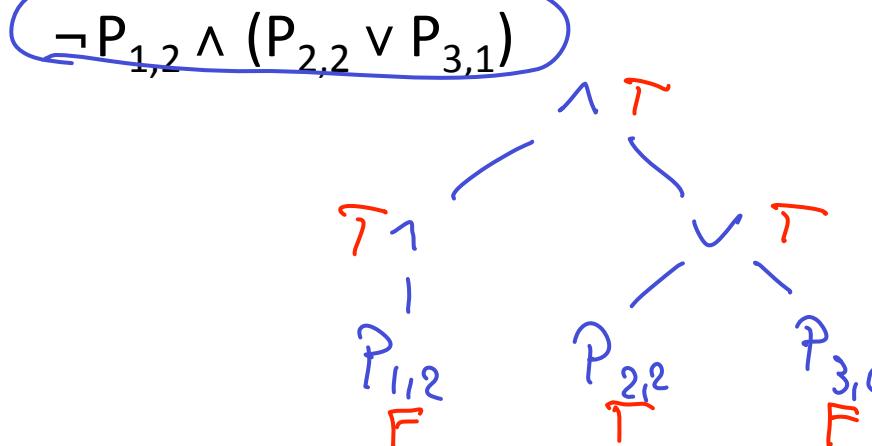
Rules for evaluating truth with respect to a model m :

$\neg S$ 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

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



Truth tables for connectives

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
F	F	T	F	F	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
T	T	F	T	T	T	T

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
$(\alpha \Rightarrow \beta) \equiv (\neg \beta \Rightarrow \neg \alpha)$	contraposition
$(\alpha \Rightarrow \beta) \equiv (\neg \alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	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

$\alpha \equiv \beta$ if and only if $M(\alpha) = M(\beta)$

Wumpus world in prop. logic

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 $[i, j]$.

$$KB = \left\{ \begin{array}{l} \neg P_{1,1} \\ \neg B_{1,1} \\ B_{2,1} \end{array} \right\} \quad KB = \neg P_{1,1} \wedge \neg B_{1,1} \wedge B_{2,1}$$

"Pits cause breezes in adjacent squares"

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

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

3 coloring in prop. logic

Vars WA_R, WA_B, WA_G
 NT_R, NT_B, NT_G
:



Rules $\neg(WA_R \wedge NT_R) \wedge \neg(WA_B \wedge NT_B) \wedge \dots$

$(WA_R \wedge \neg WA_B \wedge \neg WA_G) \vee (\neg WA_R \wedge WA_B \wedge \neg WA_G) \dots$

How do we find a sol'n?

Logical inference

- **Inference:** procedure i for deducing (proving) sentences from knowledge base
- We say $KB \vdash_i \alpha$ if α can be inferred from KB using inference procedure i
- Inference i is called
 - **Sound** if whenever $KB \vdash_i \alpha$ then also $KB \models \alpha$
 - **Complete** if whenever $KB \models \alpha$ then also $KB \vdash_i \alpha$
- Thus, a sound and complete inference procedure *correctly answers any question whose answer can be inferred from KB*

Checking entailment using truth tables

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>false</i>						
<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>
\vdots												
<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>
<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>						
<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>						
<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>
\vdots												
<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>						

- Instantiate all variables
- Check that α is true whenever KB is true
- Sound and complete! ☺
- Need to check 2^n possible assignments! ☹

Checking entailment using CSP

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:

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

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

e.g.: $A \vee B$ is satisfiable ($m = \{(a, T), (b, F)\}$)

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

e.g.: $A \wedge \neg A$

Satisfiability is connected to inference:

$KB \models \alpha$ if and only if $(KB \wedge \neg \alpha)$ unsatisfiable

Satisfiability is connected to

Proof methods

- Two main classes of methods for proving $KB \models \alpha$
- Model checking
 - Truth table enumeration (always exponential in n)
 - Better: CSP (e.g, improved backtracking such as DPLL)
Check whether $(KB \wedge \neg \alpha)$ is unsatisfiable
- Proof using inference
 - Apply sequence of inference rules (syntactic manipulations)
 - Can use inference rules in a standard search algorithm

Inference rules

- Infer new valid sentences from knowledge base
- Example: **Modus ponens**

$$\frac{\alpha_1, \dots, \alpha_k, \quad \alpha_1 \wedge \dots \wedge \alpha_k \Rightarrow \beta}{\beta}$$

- Example: $\frac{\text{Rainy} , \quad \text{Rainy} \Rightarrow \text{WetLawn}}{\text{WetLawn}}$
- Modus ponens alone is sound, but **incomplete**

Resolution

- Assumes sentences in Conjunctive Normal Form (CNF)
 - This is no restriction (Tseitin transformation)
 - Example $(P_{11} \vee \neg B_{12}) \wedge (B_{12} \vee P_{11} \vee P_{22}) \wedge \dots$

- Resolution inference rule

$$\frac{\ell_1 \vee \dots \vee \ell_k, \quad m_1 \vee \dots \vee m_n}{\ell_1 \vee \dots \vee \ell_{i-1} \vee \ell_{i+1} \vee \ell_k \vee m_1 \vee m_{j-1} \vee m_{j+1} \dots \vee m_n}$$

- Sound and complete for propositional logic!

- Example:

$$\frac{\begin{array}{c} \text{Rainy} \Rightarrow \text{WetLawn}, \text{ WetLawn} \Rightarrow \text{Slippery} \\ \neg R \vee W \end{array}}{\neg R \vee S} = \text{Rainy} \Rightarrow \text{Slippery}$$

Example: Conversion to CNF

$$\begin{aligned} B_{1,1} &\Leftrightarrow (P_{1,2} \vee P_{2,1}) \\ &\equiv (B_{11} \Rightarrow (P_{12} \vee P_{21})) \wedge ((P_{12} \vee P_{21}) \Rightarrow B_{11}) \\ &\equiv (\neg B_{11} \vee P_{12} \vee P_{21}) \wedge (\neg(P_{12} \vee P_{21}) \vee B_{11}) \\ &\equiv \overbrace{\quad\quad\quad}^{\neg\neg} \wedge ((\neg P_{12} \wedge \neg P_{21}) \vee B_{11}) \\ &\equiv \overbrace{\quad\quad\quad}^{\neg\neg} \wedge (\neg P_{12} \vee B_{11}) \wedge (\neg P_{21} \vee B_{11}) \end{aligned}$$

Resolution example

- $KB = (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$ $\alpha = \neg P_{1,2}$

