Class Assignment Strategies

- **Team-Attack**: Team attack
- **Individualistic**: Search for possible
- **Political:** look at others and make decision based on who is winning, who is loosing, and conversation
- **Emotion and reasoning:** use aggression, honesty, and ruthelessness to vary agents behavior

Can you do this with just search?

Short-comings of just using search

- Inflexibility: You need to hard code every state
- Performance: exponential in the number of states
- Observability: No inference based on information seen
- No inference and reasoning

Knowledge & Reasoning

To address these issues we will introduce

- A knowledge base (KB): a list of facts that are known to the agent.
- A mechanism (Inference) to infer new facts from old facts
- Logic provides the natural language for this

Logic

Chapter 7 *(some of the materials are from Welling at UCI)*

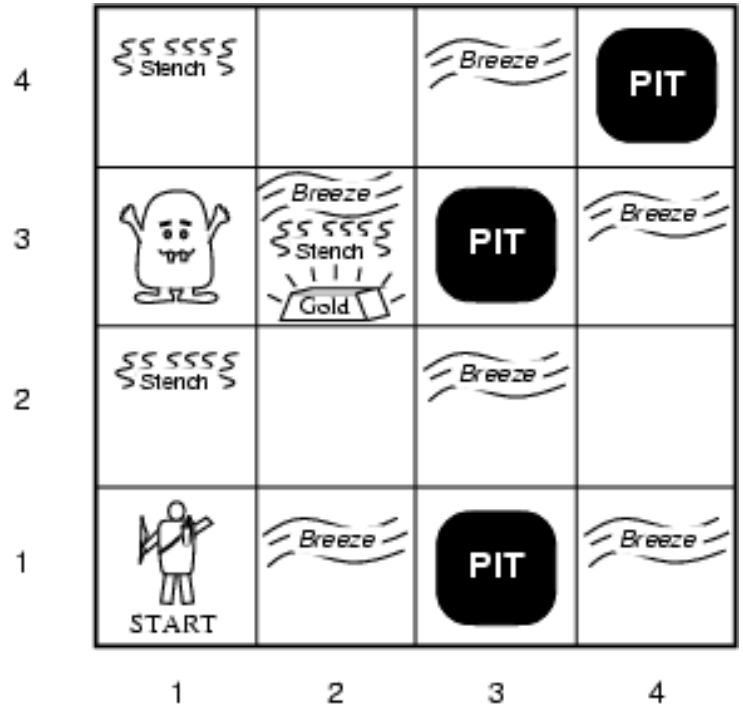
Knowledge Bases

- Knowledge base:
  - set of sentences in a formal language.

- Declarative approach to building an agent:
  - `Tell the KB` it what it needs to know.
  - `Ask the KB` it what to do

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
  - Grabbing picks up gold if in same square
  - Releasing drops the gold in same square

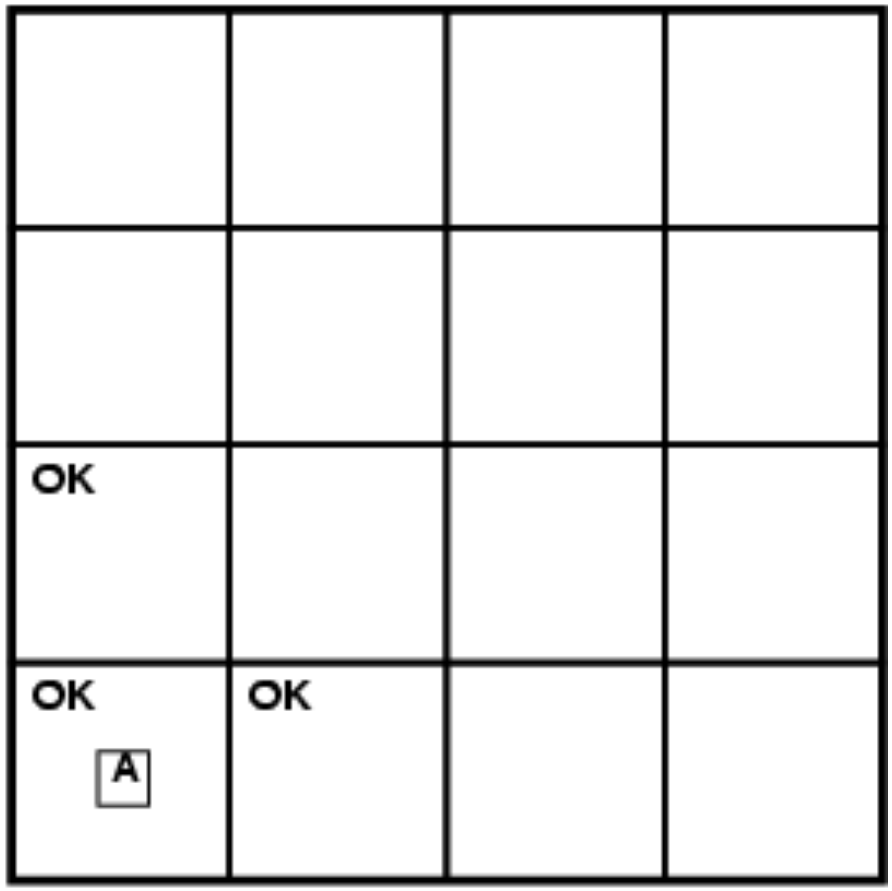


- Sensors: Stench, Breeze, Glitter, Bump, Scream
- Actuators: Left turn, Right turn, Forward, Grab, Release, Shoot

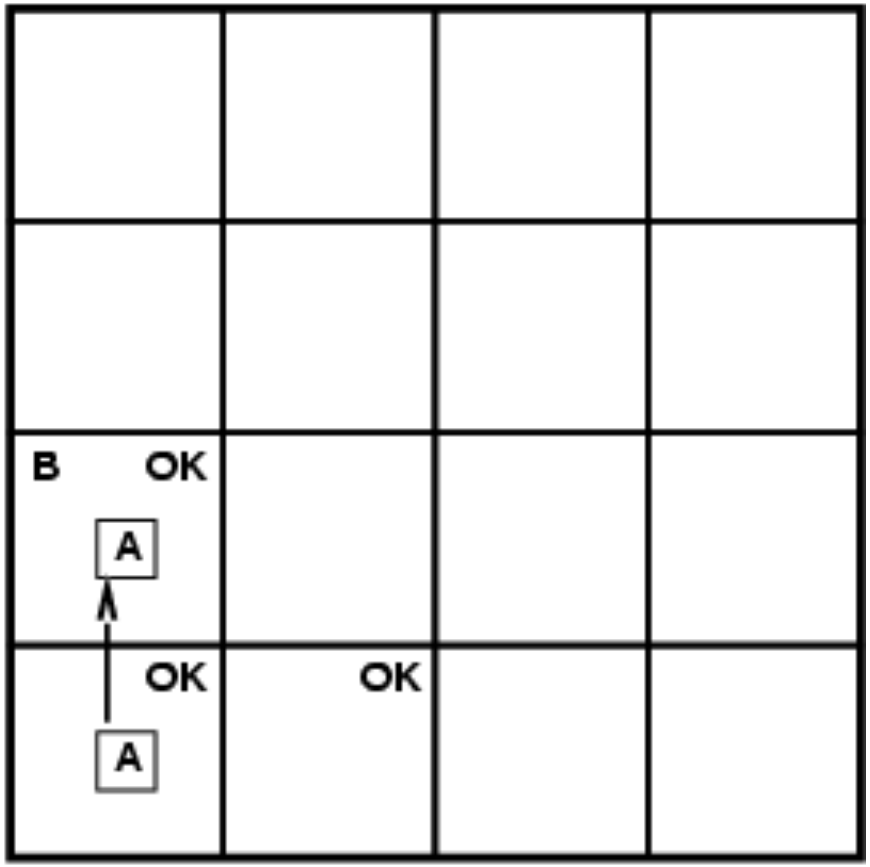
Wumpus world characterization

- Fully Observable No – only local perception
- Deterministic Yes – outcomes exactly specified
- Episodic No – things we do have an impact.
- Static Yes – Wumpus and Pits do not move
- Discrete Yes
- Single-agent? Yes – Wumpus is essentially a natural feature

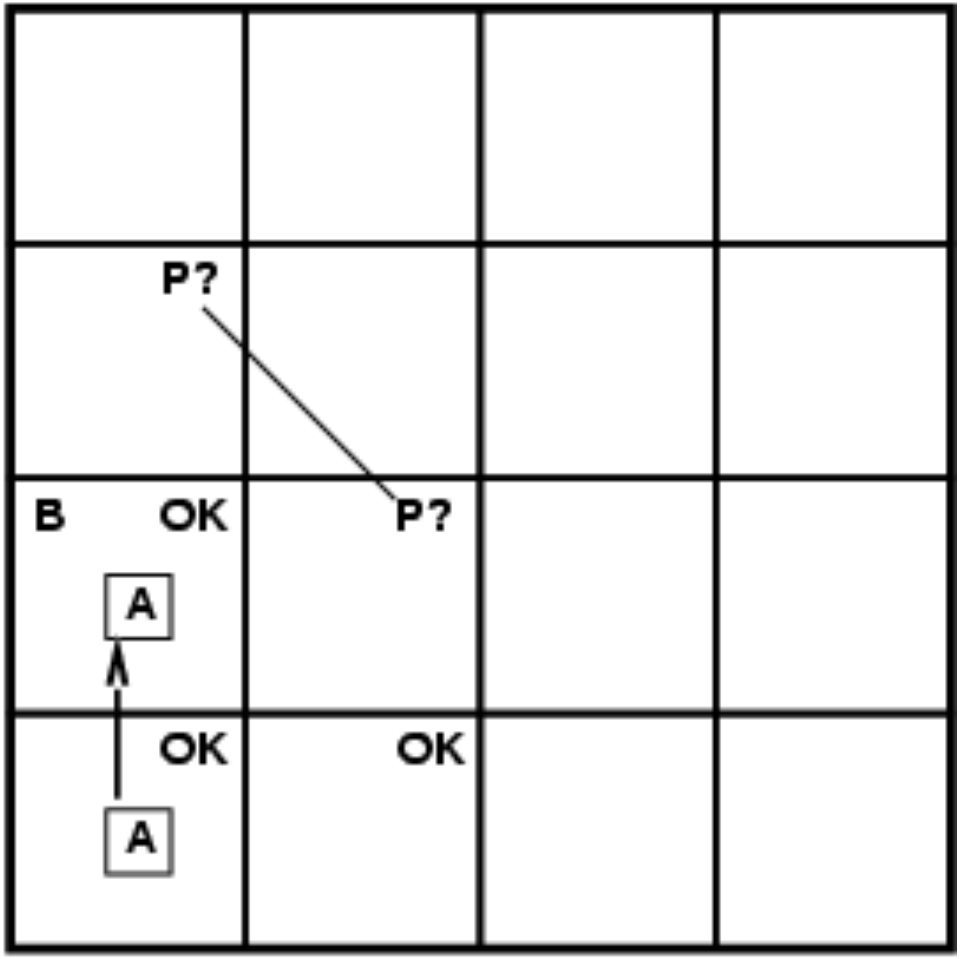
Exploring a wumpus world



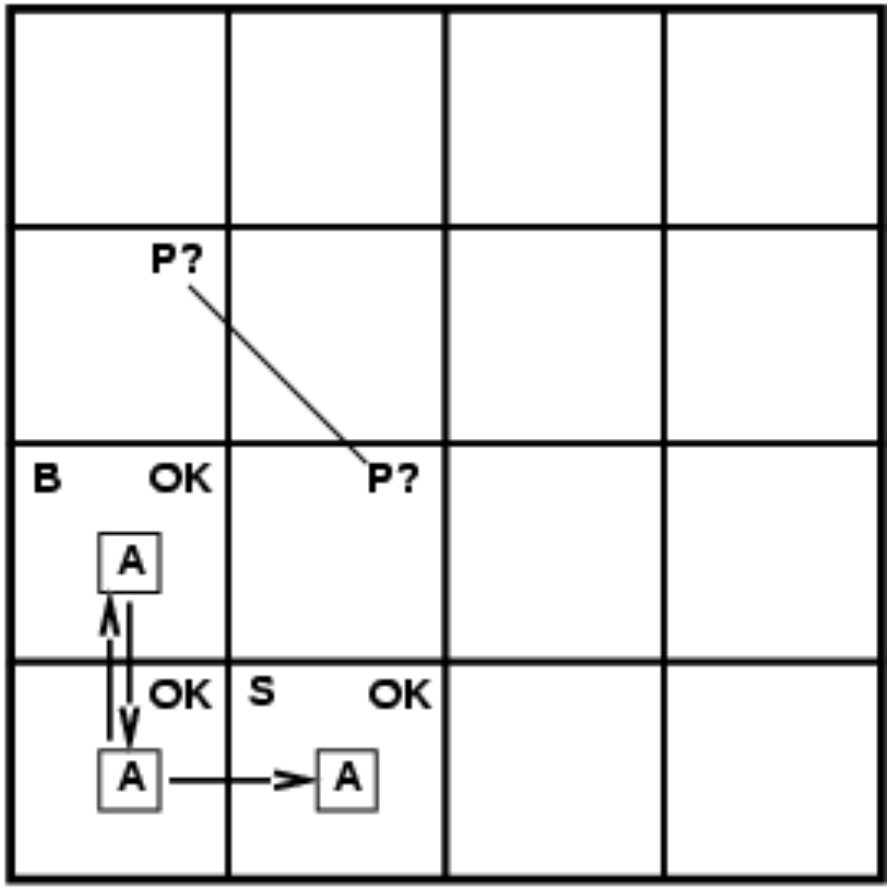
Exploring a wumpus world



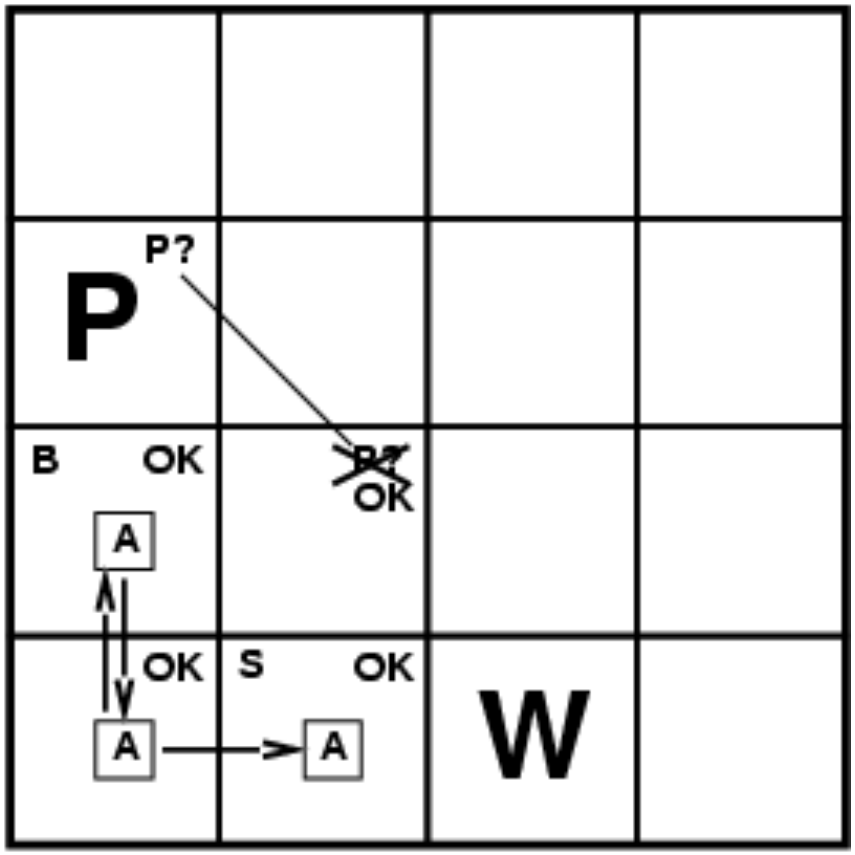
Exploring a wumpus world



Exploring a wumpus world



Exploring a Wumpus world



Why is W in 3,1 and P in 1, 3?

Logic

- We used logical reasoning to find the gold.
- 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

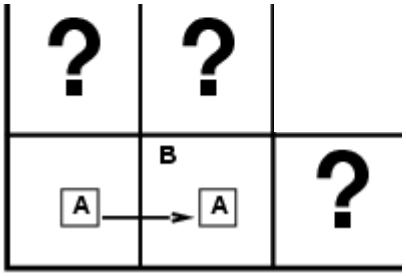
Entailment

- 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 "The Giants won".

Entailment in the wumpus world

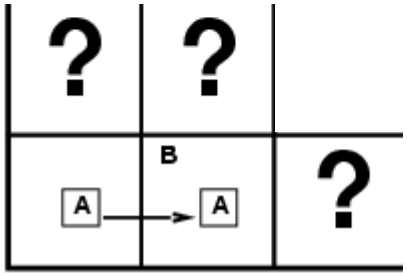


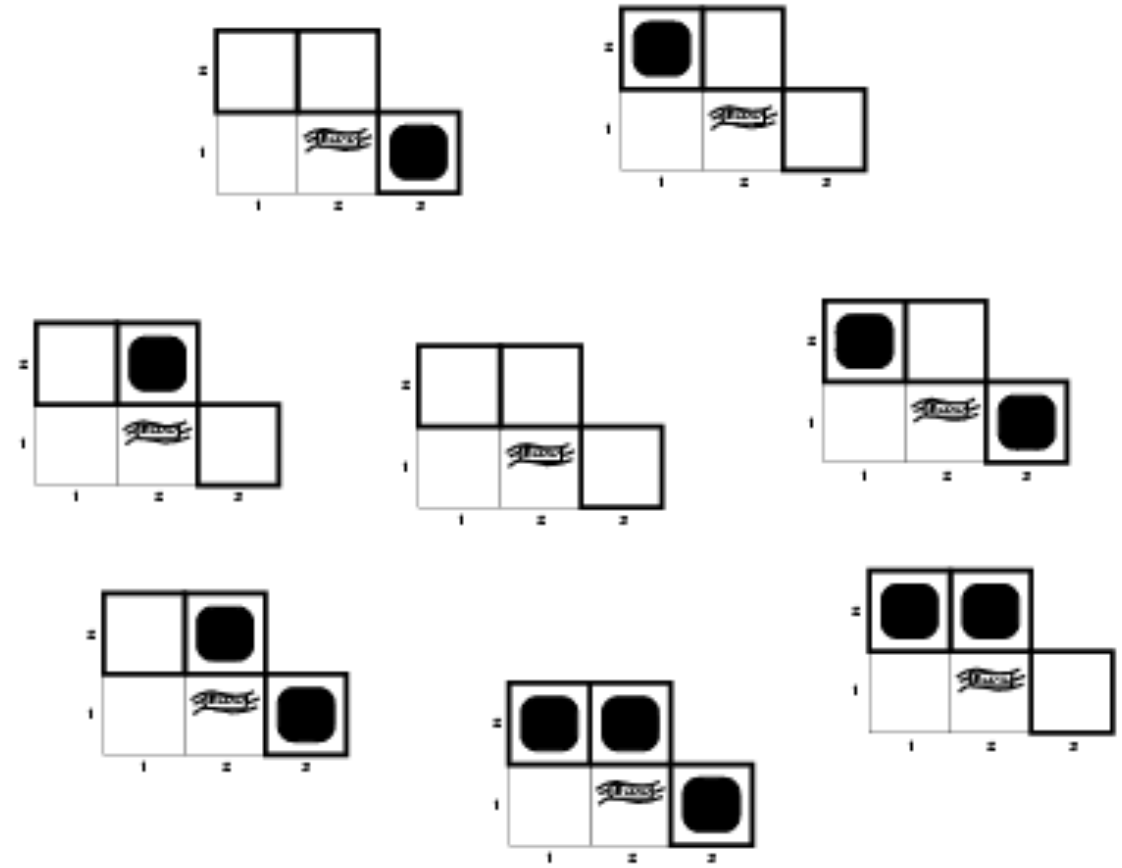
- Consider possible models for *KB* assuming only pits and a reduced Wumpus world

- Situation after detecting

nothing in [1,1], moving right, breeze in [2,1]

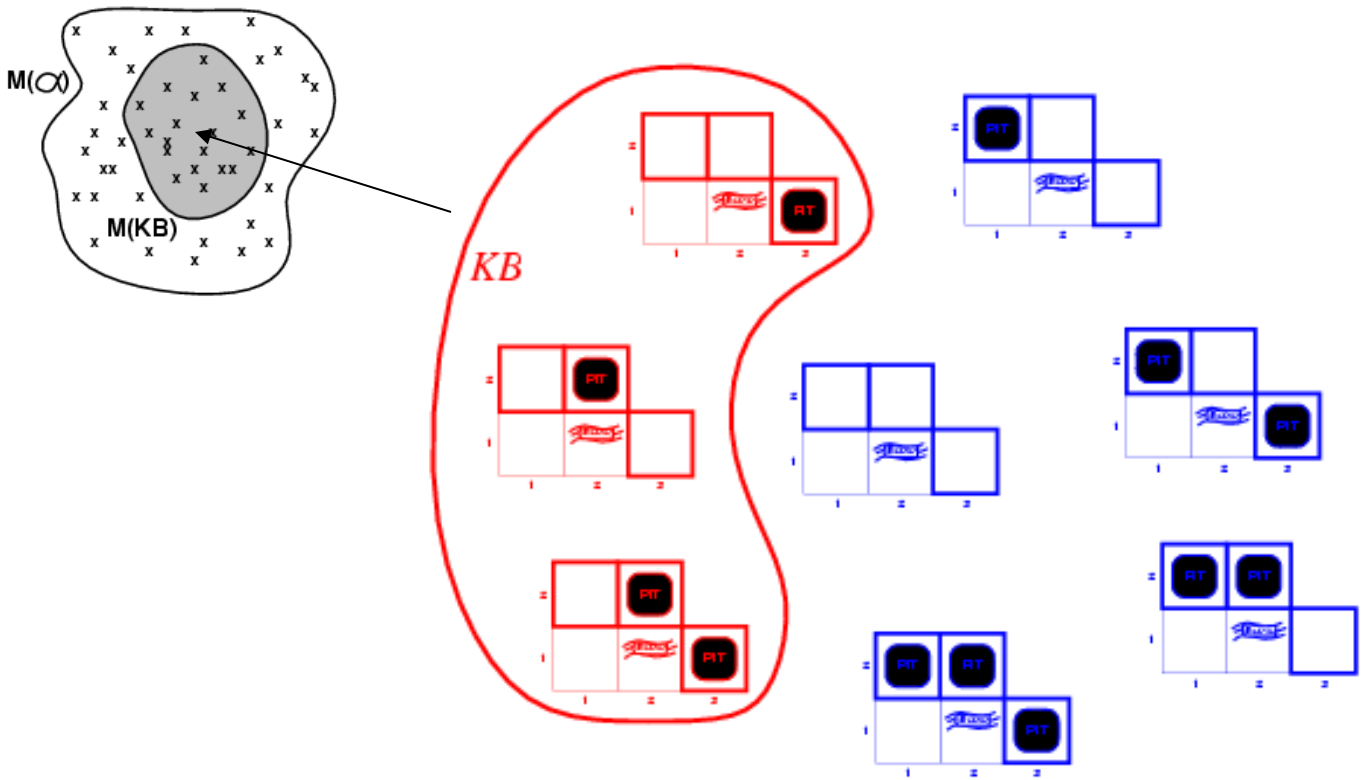
Wumpus models





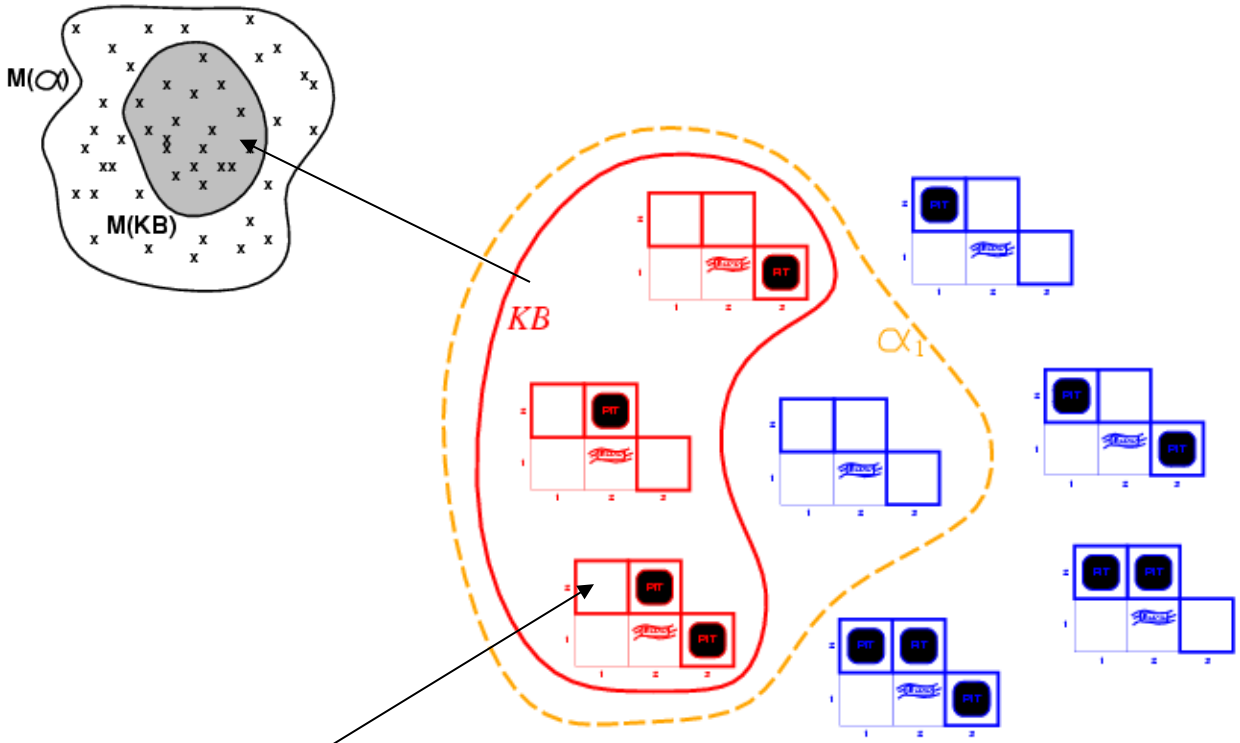
All possible ways to fill in the ?'s.

Wumpus models



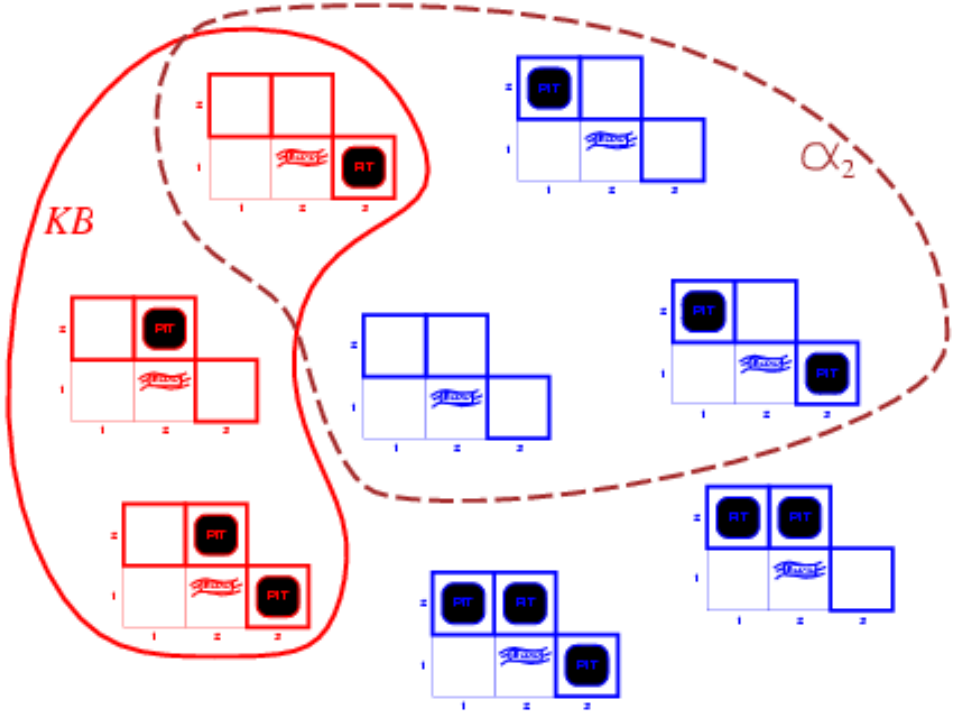
- *KB* = all possible wumpus-worlds consistent with the observations and the "physics" of the Wumpus world.

Wumpus models



α_1 = "[1,2] is safe", $KB \models \alpha_1$, proved by model checking

Wumpus models



α_2 = "[2,2] is safe", $KB \models \alpha_2$

Inference Procedures

- *KB* $\vdash_i$ α = sentence α can be derived from *KB* by procedure *i*
- Soundness: *i* is sound if whenever *KB* $\vdash_i$ α, it is also true that *KB* $\models$ α
  (i.e., no wrong inferences, but maybe not all inferences)
- Completeness: *i* is complete if whenever *KB* $\models$ α, it is also true that *KB* $\vdash_i$ α
  (i.e., all inferences can be made, but maybe some wrong extra ones as well)

Propositional logic

- Propositional logic is the simplest logic

- The proposition symbols are sentences, S
  - $\neg S$ (negation)
  - $S_1 \wedge S_2$ (conjunction)
  - $S_1 \vee S_2$ (disjunction)
  - $S_1 \Rightarrow S_2$ (implication)
  - $S_1 \Leftrightarrow S_2$ (biconditional)

Propositional logic

Each model/world 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
$S_1 \Rightarrow S_2$	is true iff	S_1 is false or	S_2 is true
$S_1 \Leftrightarrow S_2$	is true iff	$S_1 \Rightarrow S_2$ is true and $S_2 \Rightarrow S_1$ is true	

Propositional logic

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

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

Truth tables for connectives

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

OR: P or Q is true or both are true.
XOR: P or Q is true but not both.

Implication is always true when the premises are False!

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].

start: $\neg P_{1,1}$

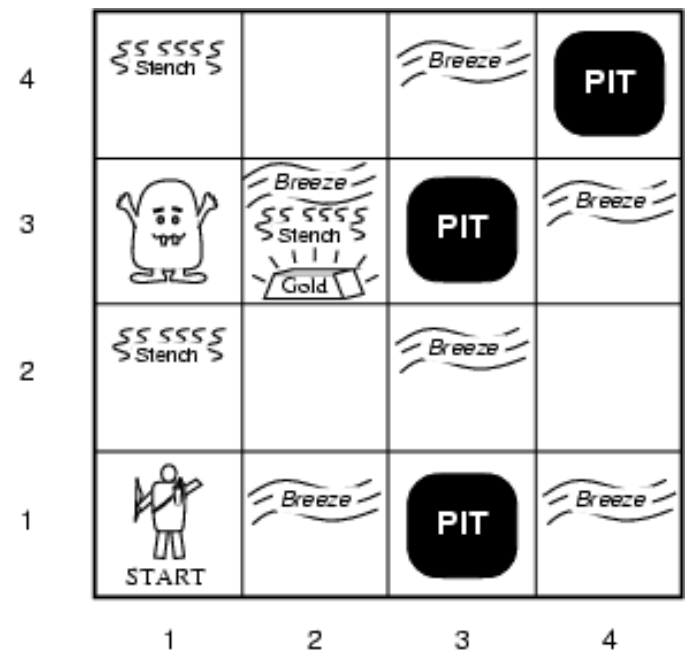
$\neg B_{1,1}$

$B_{2,1}$

- "Pits cause breezes in adjacent squares"

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

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



Inference by enumeration

- Enumeration of all models is sound and complete.
- For *n* symbols, time complexity is $O(2^n)$…
- We need a smarter way to do inference!
- In particular, we are going to infer new logical sentences from the data-base and see if they match a query.

Logical equivalence

- Two sentences are logically equivalent iff they are 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 false in all 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

(there is no model for which KB=true and α is false)

Proof methods

- Proof methods divide into (roughly) two kinds:

Application of inference rules:

Legitimate (sound) generation of new sentences from old.

- Resolution
- Forward & Backward chaining

Model checking

Searching through truth assignments.

- Improved backtracking: Davis--Putnam-Logemann-Loveland (DPLL)
- Heuristic search in model space: Walksat.

Normal Form

We like to prove:

$$\boxed{\begin{array}{l} KB \models \alpha \\ \textit{equivalent to}: KB \wedge \neg\alpha \textit{ unsatifiable} \end{array}}$$

We first rewrite $KB \wedge \neg\alpha$ **into conjunctive normal form (CNF).**

A "conjunction of disjunctions"

literals

$$\underbrace{(A \vee \neg B)}_{\textbf{Clause}} \wedge \underbrace{(B \vee \neg C \vee \neg D)}_{\textbf{Clause}}$$

- **Any KB can be converted into CNF.**
- **In fact, any KB can be converted into CNF-3 using clauses with at most 3 literals.**

Example: Conversion to CNF

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

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
   $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

2. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$

3. Move $\neg$ inwards using de Morgan's rules and double-negation:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$

4. Apply distributive law ($\wedge$ over $\vee$) and flatten:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

Resolution Algorithm

- **The resolution algorithm tries to prove:** $KB \models \alpha$ *equivalent to* $KB \wedge \neg\alpha$ *unsatisfiable*

- **Generate all new sentences from KB and the query.**
- **One of two things can happen:**

1. **We find** $P \wedge \neg P$ **which is unsatisfiable. I.e. we can entail the query.**

2. **We find no contradiction: there is a model that satisfies the sentence** $KB \wedge \neg\alpha$ **(non-trivial) and hence we cannot entail the query.**

Proof by Resolution

1. $P \vee Q$
2. $\neg P \vee R$
3. $\neg Q \vee R$

Prove R.

Proof by Resolution

1. P v Q
2. ¬P v R
3. ¬Q v R

Knowledge Base

1. P v Q
2. ¬P v R
3. ¬Q v R

1, 2 Resolution

4. Q v R
3. ¬Q v R

Prove R.

(P v Q) and (¬P v R) = true
means either Q or R have to be true

P has to be True to False, then
If P is false then
Q has to be true

If P is true then
R has to be true

Therefore, if P has to be T or F, then Q is T or R is T, thus you can cancel literals

Proof by Resolution

1. P v Q
2. ¬P v R
3. ¬Q v R

Prove R.

Knowledge Base

1. P v Q
2. ¬P v R
3. ¬Q v R

1, 2 Resolution

4. Q v R
3. ¬Q v R

4, 3 Resolution

4. Q v R
3. ¬Q v R

(Q v R) and (¬Q v R) = true
means either R or R has to be true

Q has to be True to False, then
If Q is false then
R has to be true

If Q is true then
R has to be true

Therefore, if Q has to be T or F, then R is T, thus you can cancel literals

Horn Clauses: A Special Case

- Resolution can be exponential in space and time.
- If we can reduce all clauses to "Horn clauses" resolution is linear in space and time
- Horn clause: a clause with at most one positive literal
  - $\neg A \vee \neg B \vee \neg C \vee \neg D$
- Definite clause: a Horn clause with exactly one positive literal
  - $\neg A \vee \neg B \vee C \vee \neg D$

Chaining

- Horn clauses are **closed under resolution**

$$(\neg A \vee B \vee \neg C) \quad (\neg B \vee \neg D \vee \neg E \vee F)$$

$$\neg A \vee \neg C \vee \neg D \vee \neg E \vee F$$

Forward Chaining

- Start with known facts and derive new knowledge to add to the knowledge base
- Agent can derive conclusions from incoming percepts
- Data-driven approach

Forward Chaining

- Horn clauses:
  - C1: $\neg P_1 \vee \neg P_2 \vee P_4$ $(P_1 \wedge P_2 \rightarrow P_4)$
  - C2: $\neg P_4 \vee P_5$ $(P_4 \rightarrow P_5)$

- Facts:
  - P_1, P_2

Forward Chaining

- Horn clauses:
  - C1: $\neg P_1 \vee \neg P_2 \vee P_4$ $(P_1 \wedge P_2 \rightarrow P_4)$
  - C2: $\neg P_4 \vee P_5$ $(P_4 \rightarrow P_5)$

- Facts:
  - P_1, P_2

- Percepts P_1 and P_2 resolve with C1 to get P_4

Forward Chaining

- Horn clauses:
  - C1: $\neg P_1 \vee \neg P_2 \vee P_4$ $(P_1 \wedge P_2 \rightarrow P_4)$
  - C2: $\neg P_4 \vee P_5$ $(P_4 \rightarrow P_5)$

- Facts:
  - P_1, P_2

- Percepts P_1 and P_2 resolve with C1 to get P_4
- Resolve P_4 with C2 to get P_5

Forward Chaining

- Horn clauses:
  - C1: $\neg P_1 \vee \neg P_2 \vee P_4$
  - C2: $\neg P_4 \vee P_5$

<u>Knowledge Base</u>

1. $P_1 \wedge P_2 \rightarrow P_4$
2. $P_4 \rightarrow P_5$

Forward Chaining

- Horn clauses:
  - C1: $\neg P_1 \vee \neg P_2 \vee P_4$
  - C2: $\neg P_4 \vee P_5$

Fact: P_1 is seen

<u>Knowledge Base</u>

1. $P_1 \wedge P_2 \rightarrow P_4$
2. $P_4 \rightarrow P_5$

<u>Knowledge Base</u>

1. $P_1 \wedge P_2 \rightarrow P_4$
2. $P_4 \rightarrow P_5$
3. P_1

Forward Chaining

- Horn clauses:
  - C1: $\neg P_1 \vee \neg P_2 \vee P_4$
  - C2: $\neg P_4 \vee P_5$

Knowledge Base

1. $P_1 \wedge P_2 \rightarrow P_4$
2. $P_4 \rightarrow P_5$

Fact: P_1 is seen

Knowledge Base

1. $P_1 \wedge P_2 \rightarrow P_4$
2. $P_4 \rightarrow P_5$
3. P_1

Fact: P_2 is seen

Knowledge Base

1. $P_1 \wedge P_2 \rightarrow P_4$
2. $P_4 \rightarrow P_5$
3. P_1
4. P_2
5. P_4
6. P_5

Using 1, 3, 4 deduce P_4
Using 2, 5 deduce P_5

Backward Chaining

- Goal-driven reasoning
- Work backwards to see if query is true
- If inconclusive, query is false
- Efficient: only touches relevant facts or rules

Backward Chaining

- Horn clauses:
  - C1: $\neg P_1 \vee \neg P_2 \vee P_4$ $(P_1 \wedge P_2 \rightarrow P_4)$
  - C2: $\neg P_4 \vee P_5$ $(P_4 \rightarrow P_5)$

- Facts:
  - P_1, P_2

- Goal: P_5

- Subgoal: prove P_4

Backward Chaining

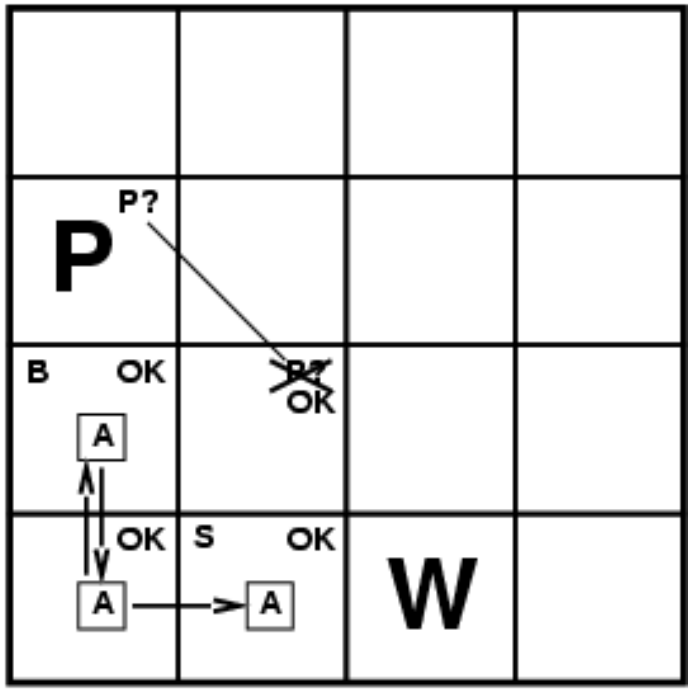
- Horn clauses:
  - C1: $\neg P_1 \vee \neg P_2 \vee P_4$ $(P_1 \wedge P_2 \rightarrow P_4)$
  - C2: $\neg P_4 \vee P_5$ $(P_4 \rightarrow P_5)$

- Facts:
  - P_1, P_2

- Goal: P_5

- Subgoal: prove P_4
  - Sub-sub goal: prove P_2
  - Sub-sub goal: prove P_1

Class Exercise



- Write this problem in terms of propositional logic
- Use resolution to show inference that
  - W is in 3, 1
  - P is in 1, 3

(give the facts shown at different time steps)

- Show your work