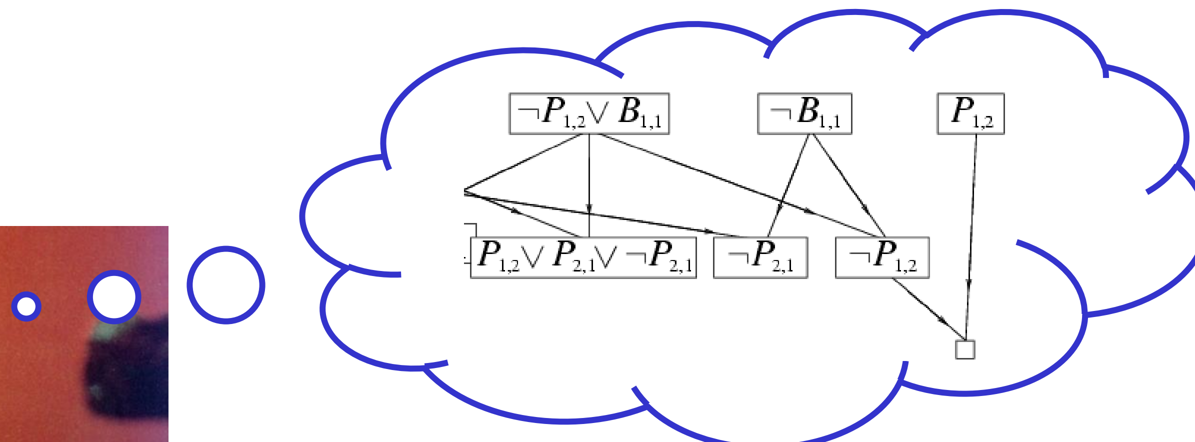
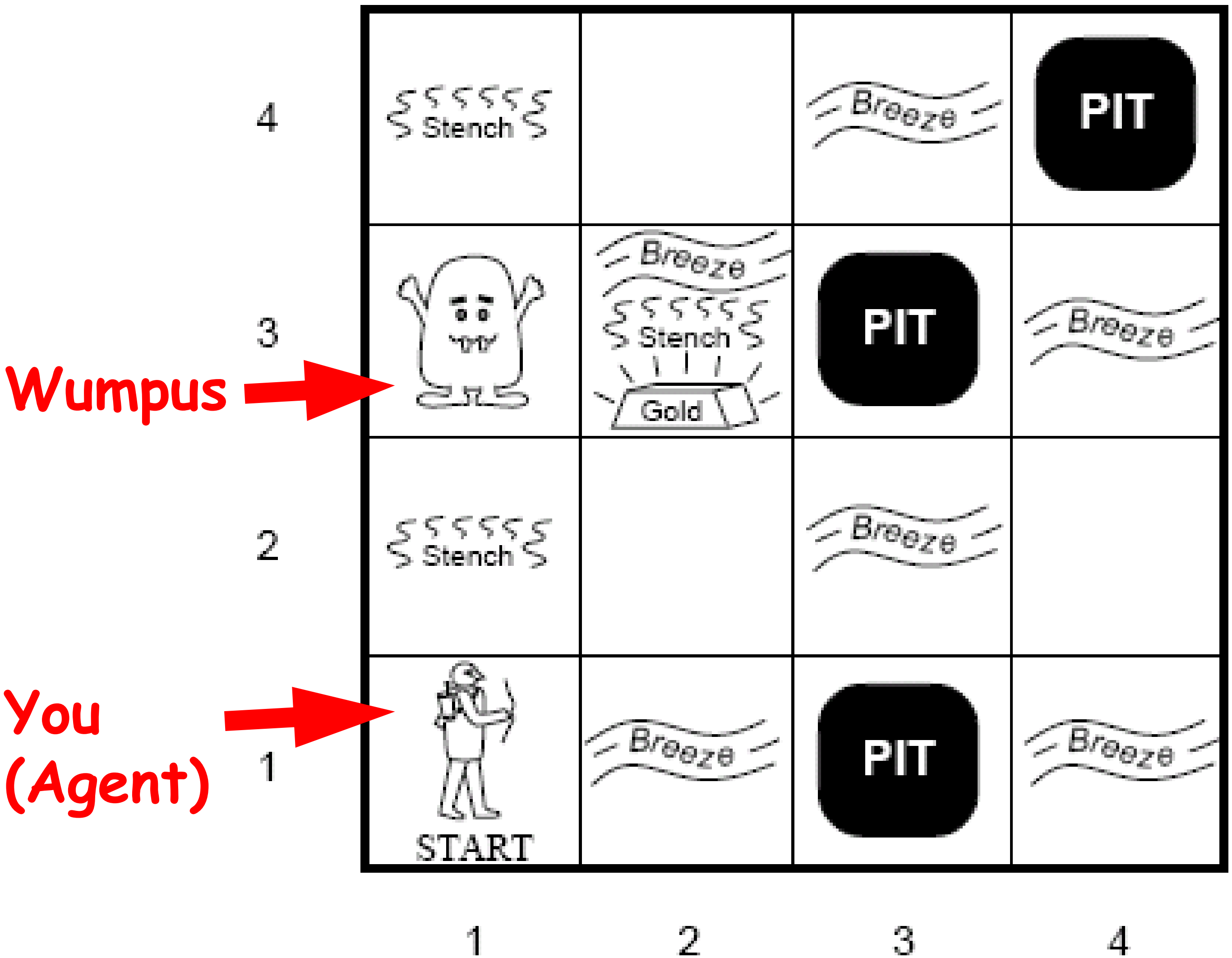
CSE 473

Chapter 7

Inference Techniques for Logical Reasoning



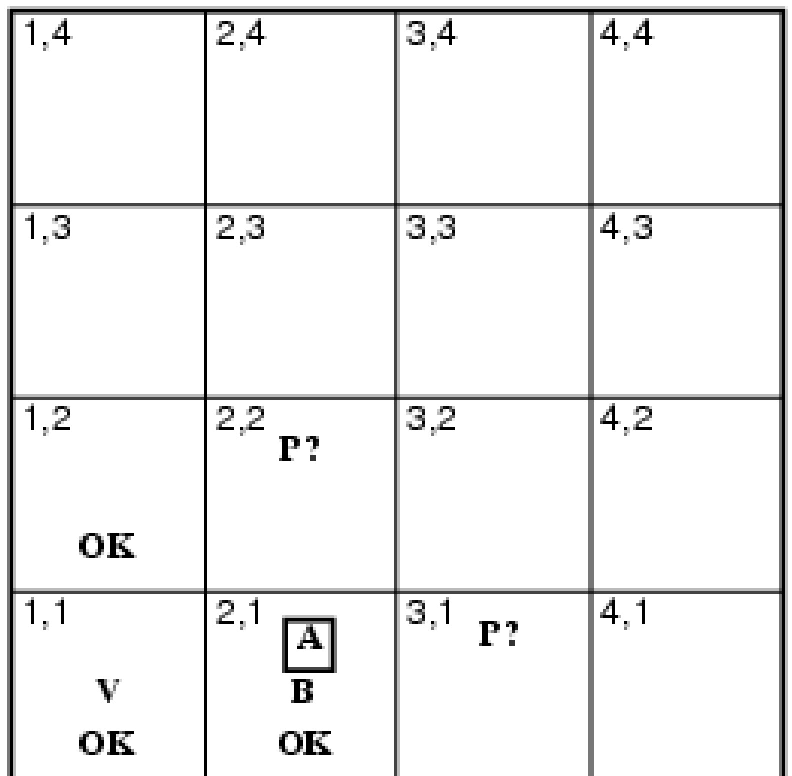
Recall: Wumpus World



Wumpusitional Logic

Proposition Symbols and Semantics:

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].



Wumpus KB

Knowledge Base (KB) includes the following sentences:

- Statements currently known to be true:

  $\neg P_{1,1}$

  $\neg B_{1,1}$

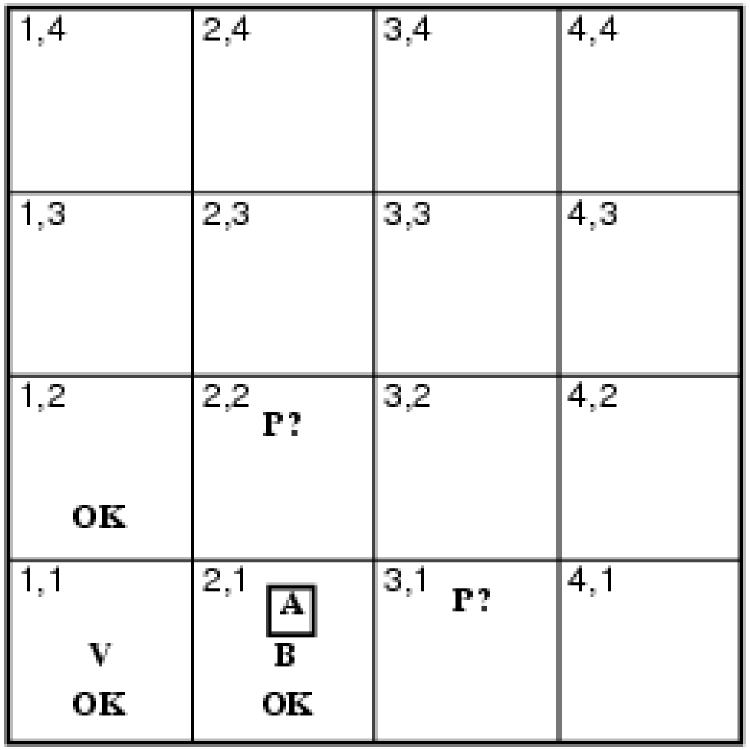
  $B_{2,1}$

- Properties of the world: E.g., "Pits cause breezes in adjacent squares"

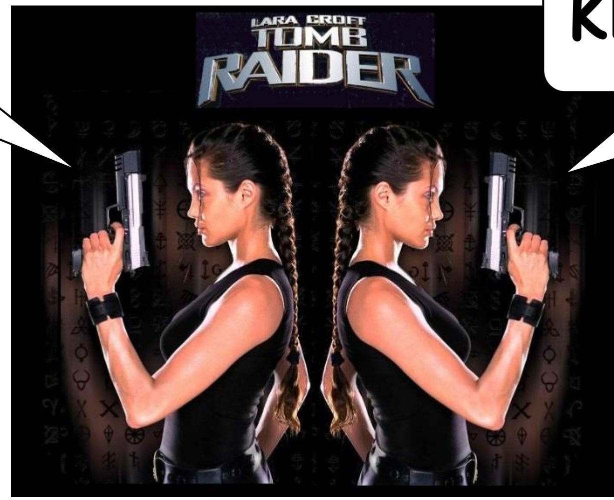
  $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

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

  (and so on for all squares)



Is there no pit in [1,2]?



KB $\models \neg P_{1,2}$?

Recall from last time:

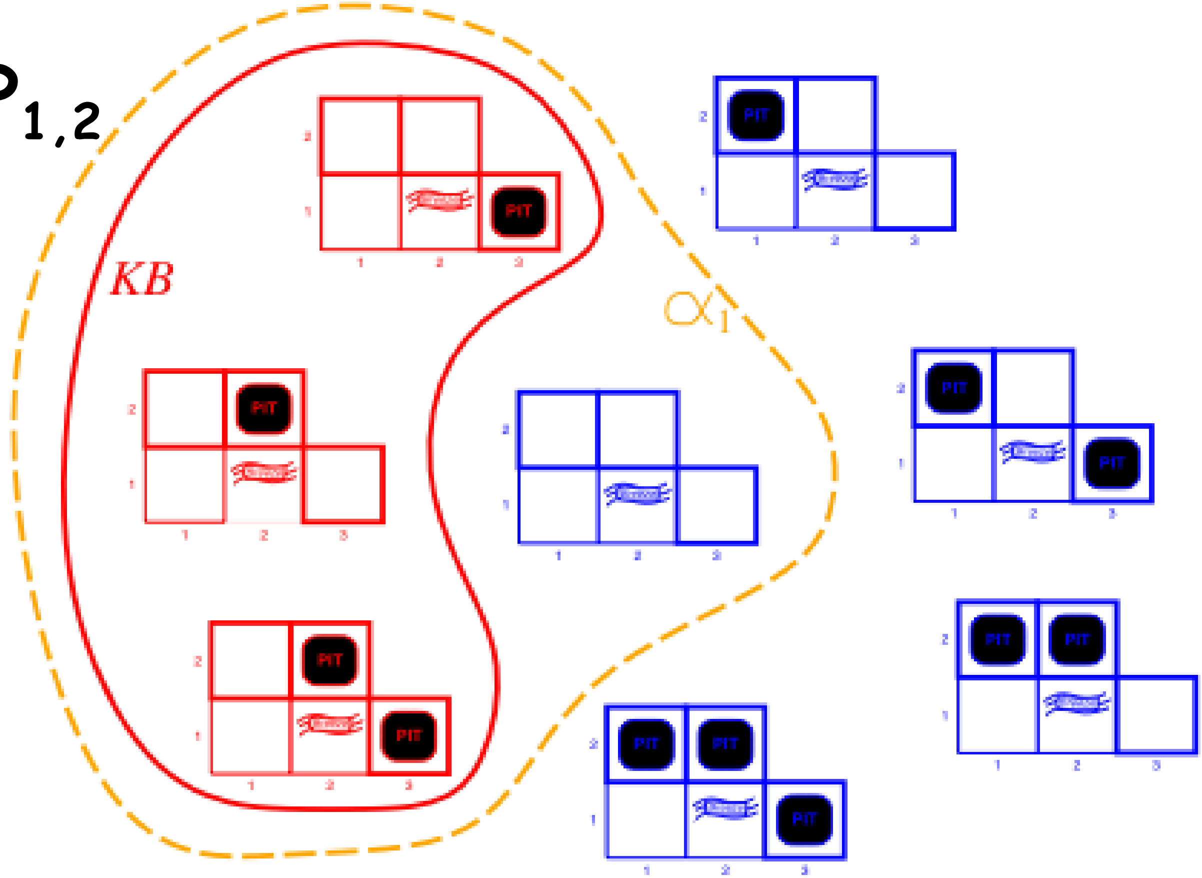
m is a model of a sentence α if α is true in *m*

M(α*)* is the set of all models of α

KB $\models \alpha$ (KB "entails" α) iff $M(KB) \subseteq M(\alpha)$

$\alpha_1 = \neg P_{1,2}$

$M(KB) \subseteq M(\alpha_1)$



Therefore,
$KB \models \neg P_{1,2}$

KB = wumpus-world rules + observations

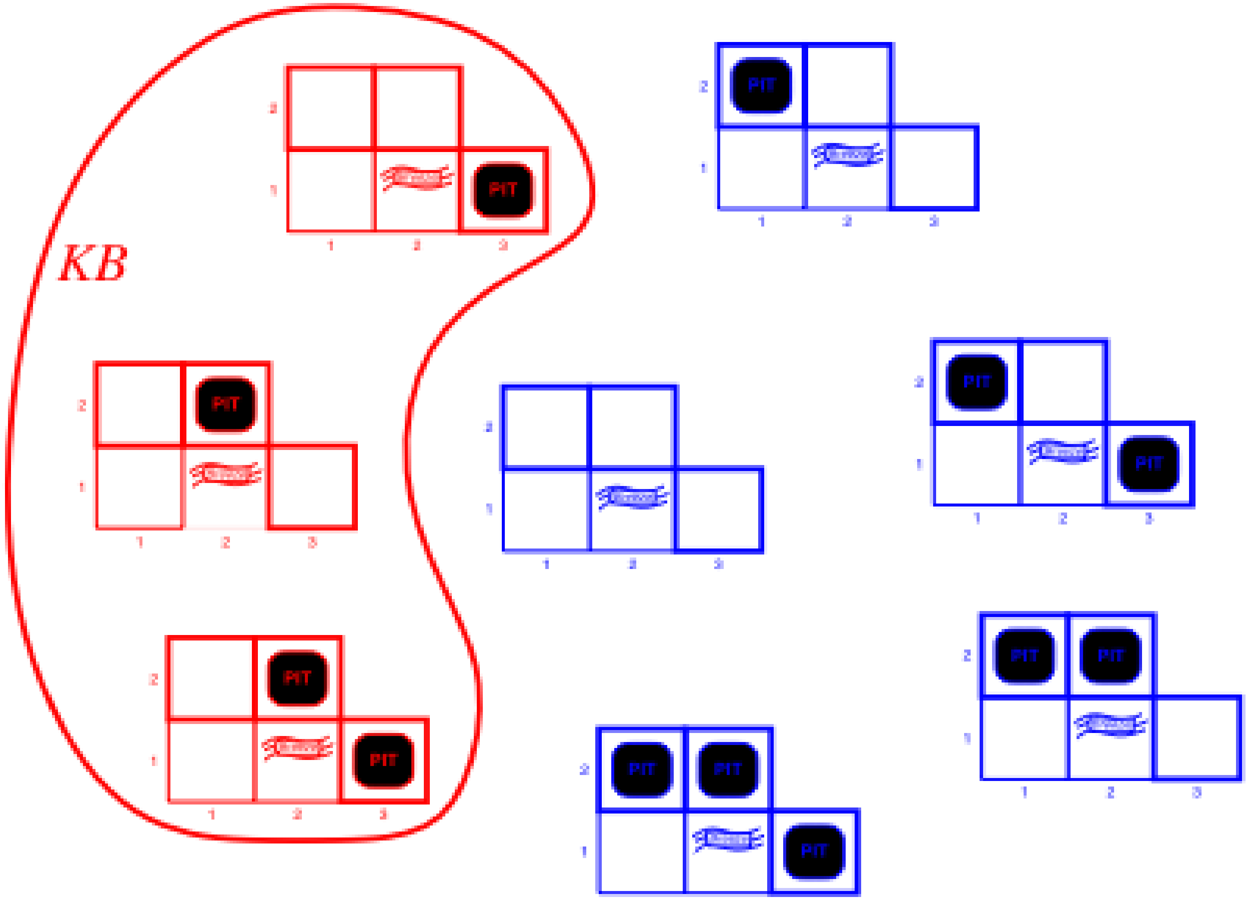
Inference by Truth Table Enumeration

	$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	KB	$\neg P_{1,2}$
Model 1	*false*	*false*	*false*	*false*	*false*	*false*	*false*	*false*	*true*
Model 2	*false*	*false*	*false*	*false*	*false*	*false*	*true*	*false*	*true*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	*false*	*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*

In all models in which KB is true, $\neg P_{1,2}$ is also true
Therefore, KB $\models \neg P_{1,2}$

Another Example

Is there a pit in [2,2]?



Inference by Truth Table Enumeration

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	KB
false	*false*	*false*	*false*	*false*	*false*	*false*	*false*
false	*false*	*false*	*false*	*false*	*false*	*true*	*false*
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
false	*true*	*false*	*false*	*false*	*false*	*false*	*false*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*false*
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*

$P_{2,2}$ is false in a model in which KB is true
Therefore, KB $\not\models P_{2,2}$

Inference by TT Enumeration

- Algorithm: Depth-first enumeration of all models (see Fig. 7.10 in text for pseudocode)
- Algorithm sound?

  Yes
- Algorithm complete?

  Yes
- For *n* symbols, time and space?
- time complexity = $O(2^n)$, space = $O(n)$

Other Inference Techniques Rely on Logical Equivalence Laws

Two sentences are logically equivalent iff they are true in the same models: $\alpha \equiv \beta$ iff $\alpha \models \beta$ and $\beta \models \alpha$

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

Inference Techniques also rely on Validity and Satisfiability

- A sentence is *valid* if it is true in *all* models (a tautology)

  e.g., *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 (proof by contradiction)

Inference/Proof Techniques

- Two kinds (roughly):

  1. Model checking
     - Truth table enumeration (always exponential in *n*)
     - Efficient backtracking algorithms,
       e.g., Davis-Putnam-Logemann-Loveland (DPLL)
     - Local search algorithms (sound but incomplete)
       e.g., randomized hill-climbing (WalkSAT)

  2. Successive application of inference rules
     - Generate new sentences from old in a sound way
     - Proof = a sequence of inference rule applications
     - Use inference rules as *successor function* in a
       standard search algorithm

Let us look at a #2 type technique: Resolution…

Inference Technique I: Resolution

Motivation

There is a pit in [1,3] or
There is a pit in [2,2]

There is no pit in [2,2]

There is a pit in [1,3]

More generally,

$$\frac{\ell_1 \vee \ldots \vee \ell_k, \qquad \neg \ell_i}{\ell_1 \vee \ldots \vee \ell_{i-1} \vee \ell_{i+1} \vee \ldots \vee \ell_k}$$

Resolution

Terminology:

Literal = proposition symbol or its negation

E.g., A, $\neg A$, B, $\neg B$, etc.

Clause = disjunction of literals

E.g., $(B \vee \neg C \vee \neg D)$

Resolution assumes sentences are in *Conjunctive Normal Form (CNF):*

sentence = conjunction of clauses

E.g., $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

Conversion to CNF

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

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 rule:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$

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

   This is in CNF – Done!

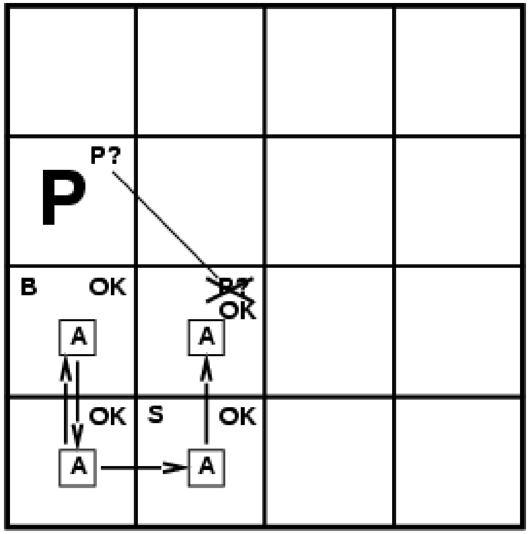
Inference Technique: Resolution

- General Resolution inference rule (for CNF):

$$\frac{l_1 \vee \ldots \vee l_i \ldots \vee l_k, \quad m_1 \vee \ldots \vee m_j \ldots \vee m_n}{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n}$$

where l_i and m_j are complementary literals i.e. $l_i = \neg m_j$.

E.g.,
$$\frac{P_{1,3} \vee P_{2,2}, \qquad \neg P_{2,2}}{P_{1,3}}$$



Soundness of Resolution Inference Rule

(Recall logical equivalence $A \Rightarrow B \equiv \neg A \vee B$)
Express each clause as:

$$\neg(l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k) \Rightarrow l_i$$

$$\neg m_j \Rightarrow (m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n)$$

$$\neg(l_i \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k) \Rightarrow (m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n)$$

(since $l_i = \neg m_j$)

Resolution algorithm

- To show **KB** $\models \alpha$, use proof by contradiction, i.e., show $KB \wedge \neg\alpha$ unsatisfiable

```
function PL-RESOLUTION(KB, α) returns true or false
    clauses ← the set of clauses in the CNF representation of KB ∧ ¬α
    new ← { }
    loop do
        for each Ci, Cj in clauses do
            resolvents ← PL-RESOLVE(Ci, Cj)
            if resolvents contains the empty clause then return true
            new ← new ∪ resolvents
        if new ⊆ clauses then return false
        clauses ← clauses ∪ new
```

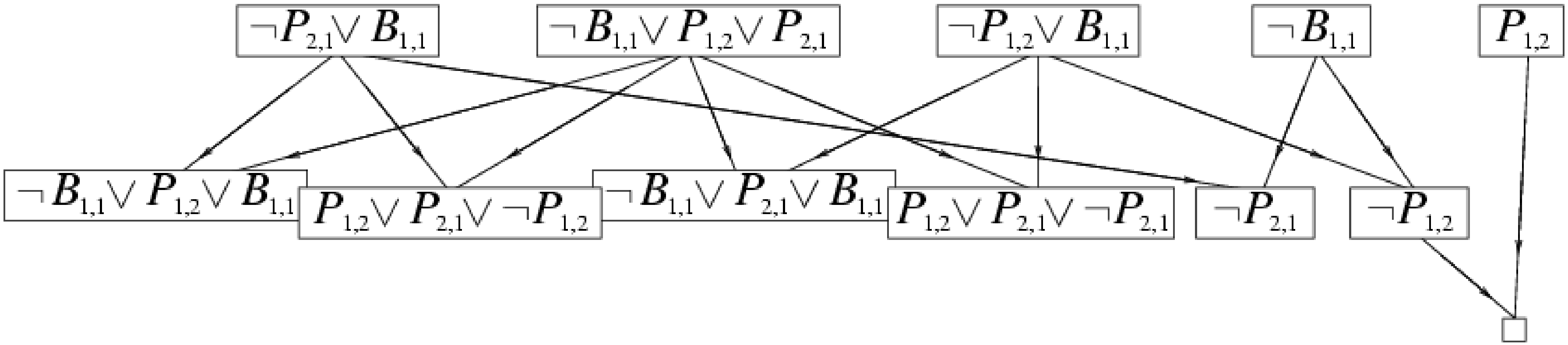
Resolution example

Given no breeze in [1,1], prove there's no pit in [1,2]

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

Resolution: Convert to CNF and show $KB \wedge \neg\alpha$ is unsatisfiable

Resolution example



Empty clause

(i.e., KB $\wedge \neg \alpha$ unsatisfiable)

Next Time

- WalkSAT
- Logical Agents: Wumpus
- First-Order Logic
- To Do:

  Project #2
  Finish Chapter 7
  Start Chapter 8