Lecture 5a: Propositional Logic Inference

CSCI 360

Introduction to Artificial Intelligence

USC

Assumption:

- It is not sunny this afternoon and it is colder than yesterday.
- We will go swimming only if it is sunny.
- If we do not go swimming, then we will take a canoe trip.
- If we take a canoe trip, then we will be home by sunset.

Conclusion:

- We will be home by the sunset.

Here is where we are…

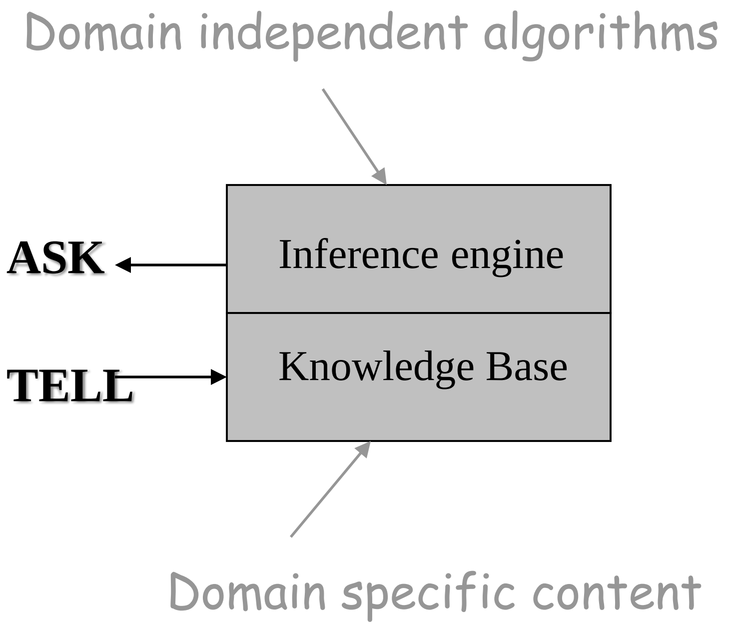
Week	30000D	30282R	Topics	Chapters
1	1/7	1/8	Intelligent Agents	[Ch 1.1-1.4 and 2.1-2.4]
	1/9	1/10	Problem Solving and Search	[Ch 3.1-3.3]
2	1/14	1/15	Uninformed Search	[Ch 3.3-3.4]
	1/16	1/17	Heuristic Search (A*)	[Ch 3.5]
3	1/21	1/22	Heuristic Functions	[Ch 3.6]
	1/23	1/24	Local Search	[Ch 4.1-4.2]
	1/25		Project 1 Out	
4	1/28	1/29	Adversarial Search	[Ch 5.1-5.3]
	1/30	1/31	Knowledge Based Agents	[Ch 7.1-7.3]
5	2/4	2/5	Propositional Logic Inference	[Ch 7.4-7.5]
	2/6	2/7	First-Order Logic	[Ch 8.1-8.4]
	2/8		Project 1 Due	
	2/8		Homework 1 Out	
6	2/11	2/12	Rule-Based Systems	[Ch 9.3-9.4]
	2/13	2/14	Search-Based Planning	[Ch 10.1-10.3]
	2/15		Homework 1 Due	
7	2/18	2/19	SAT-Based Planning	[Ch 10.4]
	2/20	2/21	Knowledge Representation	[Ch 12.1-12.5]
8	2/25	2/26	Midterm Review	
	2/27	2/28	**Midterm Exam**	

Outline

- What is AI?
- Problem-solving agent
  - Uninformed (DFS), informed (A*), and local search
  - Adversarial search (minimax, alpha-beta pruning)
- **Knowledge-based agent**
  - The Wumpus World
  - Propositional Logic
  - **Propositional Logic Inference**

Recap: *Logic for knowledge representation*

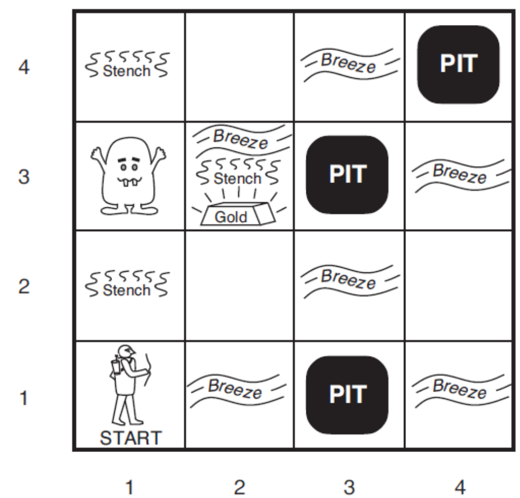
- **Logic** as a language for knowledge representation
  - Propositional logic (Boolean)
  - First-order logic (FOL)



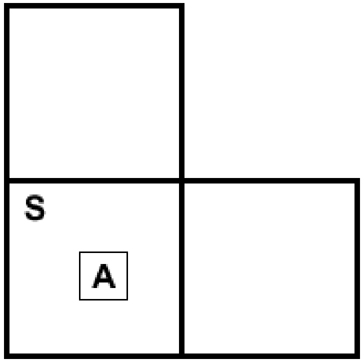
- Advantage
  - Can combine and recombine information to suit many purposes

Recap: *The Wumpus World*

- Illustrating **unique strength** of “knowledge-based” agents
  - A cave consisting of dark rooms, on a 4x4 grid
  - Beast (named Wumpus) hidden in one room
  - Pits hidden in some rooms
  - Gold hidden in one room



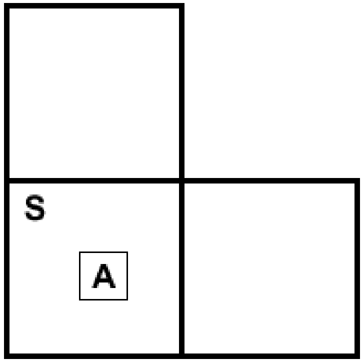
Recap: *Tight spots*



Smell in (1,1)

$\Rightarrow$ cannot move

Recap: *Tight spots*



Smell in (1,1)

$\Rightarrow$ cannot move

Can use a strategy of <u>coercion</u>:

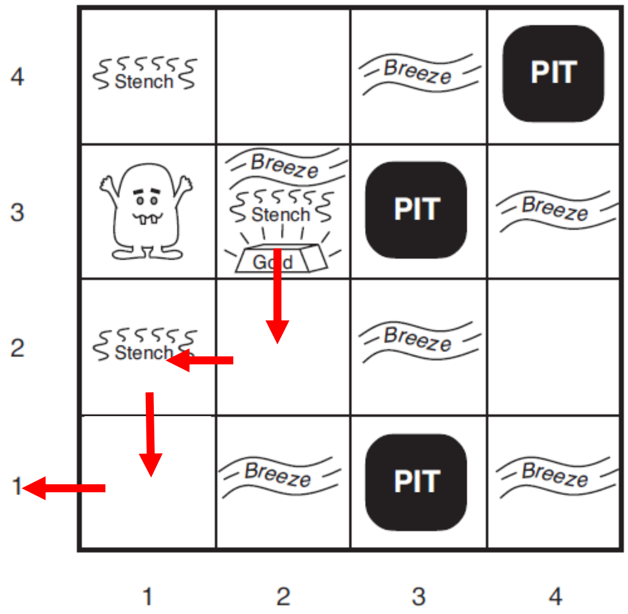
shoot straight ahead
wumpus was there $\Rightarrow$ dead $\Rightarrow$ safe
wumpus wasn't there $\Rightarrow$ safe

Recap: *Can you solve it **using search** alone?*

- **No,** unless you risk "being eaten" or "dying in pit" **multiple times**, before learning the entire **transition model** of the environment

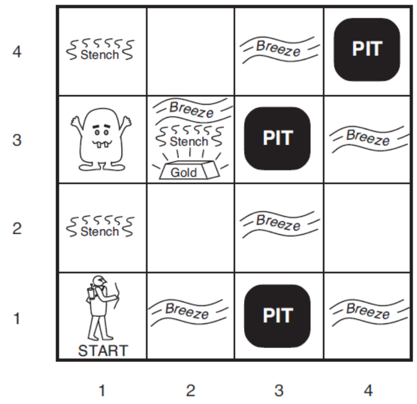
- With a "**Knowledge Base (KB)**", you can infer facts such as
  - [2,2] cannot have Pit
  - [2,2] cannot have Wumpus
  - [1,3] must have Wumpus
  - [3,1] must have Pit

- ***Correctness is guaranteed***
  - As long as KB is correct



Recap: *Knowledge Base (KB)*

- A set of **sentences** describing aspects of the environment



- **Symbols:**

$P_{x,y}$ is true if there is a pit in $[x, y]$.
$W_{x,y}$ is true if there is a wumpus in $[x, y]$, dead or alive.
$B_{x,y}$ is true if the agent perceives a breeze in $[x, y]$.
$S_{x,y}$ is true if the agent perceives a stench in $[x, y]$.

- **Sentence#1:** There is no pit in [1,1] $\boxed{\neg P_{1,1}\,.}$
- **Sentence#2:** There is breeze in [2,1] $\boxed{B_{2,1}\,.}$
- **Sentence#3:** *A square is breezy IFF pit is in a neighboring square*

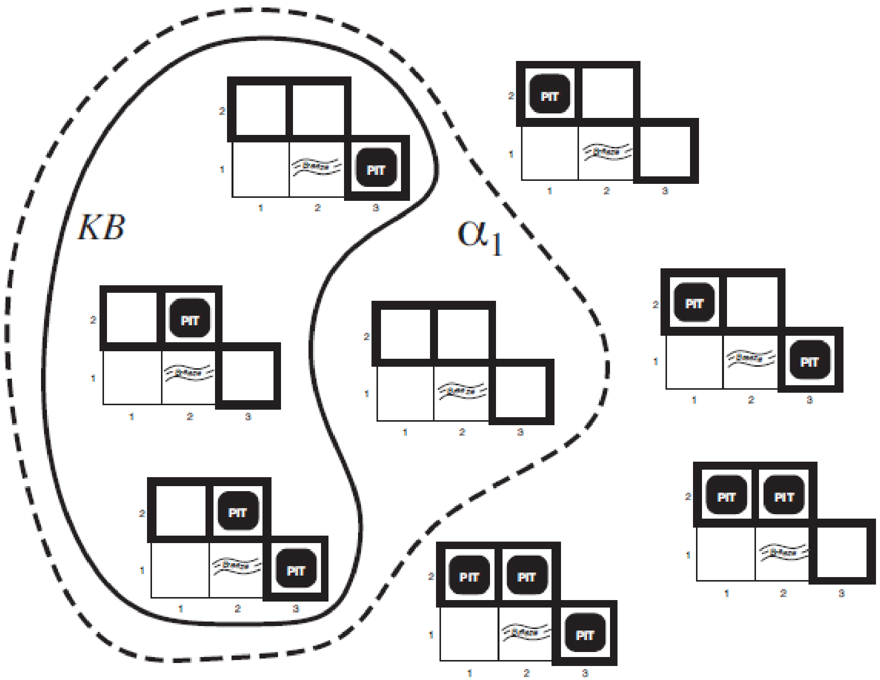
$$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})\,.$$
$$B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})\,.$$
$$B_{3,1} \ldots$$
$$B_{4,1} \ldots$$

Recap: *Entailment*

$\boxed{\alpha_1 = \text{"There is no pit in [1,2]."}}$

in every model in which KB is true, α_1 is also true.

$$\boxed{KB \models \alpha_1}$$



First, find all models in KB

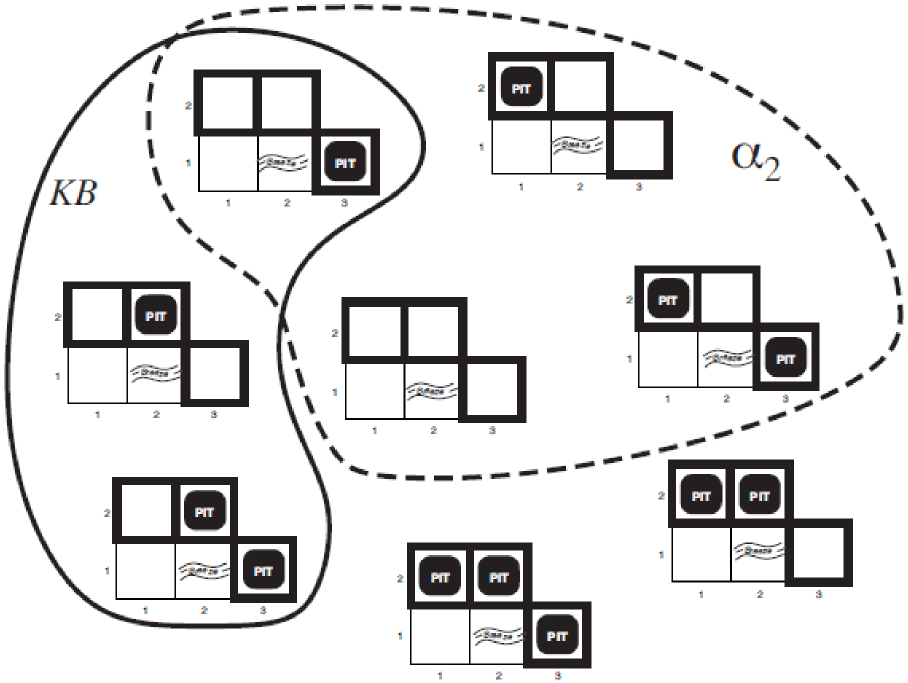
Then, for each model, check if the sentence is true

Recap: *No entailment*

α_2 = "There is no pit in [2,2]."

in some models in which KB is true, α_2 is false.

$$\boxed{KB \not\models \alpha_2}$$



First, find all models in KB

Then, for each model, check if the sentence is true

Recap: *Checking entailment*

- Two methods
  - Method#1: Based on enumeration (model checking)
  - Method#2: Based on inference rules (theorem proving)

- Enumerate all models and check if "α is true in all models in which KB is true"

$$\boxed{M(KB) \subseteq M(\alpha).}$$

Recap: *Checking entailment (example)*

$\boxed{KB \models \alpha_1}$

$\boxed{KB \not\models \alpha_2}$

α_1 = "There is no pit in [1,2]."

α_2 = "There is no pit in [2,2]."

$R_1: \quad \neg P_{1,1}$.
$R_2: \quad B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$.
$R_3: \quad B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$.
$R_4: \quad \neg B_{1,1}$.
$R_5: \quad B_{2,1}$.

KB

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	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*	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	<u>*true*</u>
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*

Recap: *Checking entailment*

- Two methods
  - Method#1: Based on enumeration (model checking)
  - Method#2: Based on inference rules (theorem proving)

- Logical equivalences

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

Recap: *Checking entailment (theory)*

- Can be done by checking "**validity**" or "**unsatisfiability**"

For any sentences α *and* β, $\alpha \models \beta$

$\boxed{\textit{if and only if the sentence } (\alpha \Rightarrow \beta) \textit{ is valid.}}$

$\boxed{\textit{if and only if the sentence } (\alpha \wedge \neg\beta) \textit{ is unsatisfiable.}}$

Checking validity

For any sentences α *and* β, $\alpha \models \beta$

$\boxed{\textit{if and only if the sentence } (\alpha \Rightarrow \beta) \textit{ is valid.}}$

- Example: $\boxed{KB \models \alpha_1}$

$$\left.\begin{array}{ll} R_1: & \neg P_{1,1}\,. \\ R_2: & B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})\,. \\ R_3: & B_{2,1} \Leftrightarrow (P_{1,1} \lor P_{2,2} \lor P_{3,1})\,. \\ R_4: & \neg B_{1,1}\,. \\ R_5: & B_{2,1}\,. \end{array}\right\} \textbf{\textit{KB}}$$

$\boldsymbol{\alpha_1}$ = "There is no pit in [1,2]."

- $(R_1 \land \mathbf{R_2} \land R_3 \land \mathbf{R_4} \land R_5) \Rightarrow (\boldsymbol{\neg P_{1,2}})$

Inference rules

- **Modus Ponens** (Latin for mode that affirms)

$$\frac{\alpha \Rightarrow \beta, \qquad \alpha}{\beta}$$

- **And-Elimination**

$$\frac{\alpha \wedge \beta}{\alpha}$$

Inference rules (more)

- $\rightarrow$ $$\frac{\alpha \Leftrightarrow \beta}{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}$$

- $\leftarrow$ $$\frac{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}{\alpha \Leftrightarrow \beta}$$

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

Applying inference rules

- Example: $\boxed{KB \models \alpha_1}$

$$\left.\begin{array}{lll} R_1: & \neg P_{1,1}\,. & \\ R_2: & B_{1,1} \;\Leftrightarrow\; (P_{1,2} \lor P_{2,1})\,. & \\ R_3: & B_{2,1} \;\Leftrightarrow\; (P_{1,1} \lor P_{2,2} \lor P_{3,1})\,. & \\ R_4: & \neg B_{1,1}\,. & \\ R_5: & B_{2,1}\,. & \end{array}\right\} \textbf{KB}$$

$\boldsymbol{\alpha_1}$ = "There is no pit in [1,2]."

- $(R_1 \land \mathbf{R_2} \land R_3 \land \mathbf{R_4} \land R_5) \Rightarrow (\neg \mathbf{P_{1,2}})$

- *From R_2:*
- *And-Elimination:*
- *Contra-positive:*
- *From R_4:*
- *Modus Ponens:*
- *De Morgan:*

Applying inference rules

- Example: $\boxed{KB \models \alpha_1}$

$$\left.\begin{array}{ll} R_1: & \neg P_{1,1}. \\ R_2: & B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1}). \\ R_3: & B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1}). \\ R_4: & \neg B_{1,1}. \\ R_5: & B_{2,1}. \end{array}\right\} \textbf{\textit{KB}}$$

α_1 = "There is no pit in [1,2]."

- $(R_1 \wedge \mathbf{R_2} \wedge R_3 \wedge \mathbf{R_4} \wedge R_5) \Rightarrow (\neg \mathbf{P_{1,2}})$

- *From R_2:* $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$.
- *And-Elimination:* $((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$.
- *Contra-positive:* $(\neg B_{1,1} \Rightarrow \neg(P_{1,2} \vee P_{2,1}))$.
- *From R_4:* $\neg B_{1,1}$
- *Modus Ponens:* $\neg(P_{1,2} \vee P_{2,1})$.
- *De Morgan:* $\neg P_{1,2} \wedge \neg P_{2,1}$

Applying inference rules

- Example: $\boxed{KB \models \alpha_1}$

$$\left.\begin{array}{ll} R_1: & \neg P_{1,1}\,. \\ R_2: & B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})\,. \\ R_3: & B_{2,1} \Leftrightarrow (P_{1,1} \lor P_{2,2} \lor P_{3,1})\,. \\ R_4: & \neg B_{1,1}\,. \\ R_5: & B_{2,1}\,. \end{array}\right\} KB$$

α_1 = "There is no pit in [1,2]."

A sequence of actions -- Find it by hand, or by computer?

- $(R_1 \land \mathbf{R_2} \land R_3 \land \mathbf{R_4} \land R_5) \Rightarrow (\neg \mathbf{P_{1,2}})$

- From R_2: $(B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1})\,.$
- And-Elimination: $((P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1})\,.$
- Contra-positive: $(\neg B_{1,1} \Rightarrow \neg(P_{1,2} \lor P_{2,1}))\,.$
- From R_4: $\neg B_{1,1}$
- Modus Ponens: $\neg(P_{1,2} \lor P_{2,1})\,.$
- De Morgan: $\neg P_{1,2} \land \neg P_{2,1}$

Automation of the proof

- Problem-solving agent that searches a proof:
  - **Initial State:**
  - **Actions:**
  - **Transition model**:
  - **Goal:**

Automation of the proof

- Problem-solving agent that searches a proof:
  - **Initial State:** KB
  - **Actions:** Inference rules applied to sentences in KB
  - **Transition model**: KB’ $\leftarrow$ RESULT(KB, rule), where KB’ is KB plus all the inferred sentences
  - **Goal:** State (KB) containing the sentence to be proved

Automation of the proof

- Problem-solving agent:

$$\left.\begin{array}{llll} R_1: & \neg P_{1,1} . & & \\ R_2: & B_{1,1} & \Leftrightarrow & (P_{1,2} \vee P_{2,1}) . \\ R_3: & B_{2,1} & \Leftrightarrow & (P_{1,1} \vee P_{2,2} \vee P_{3,1}) . \\ R_4: & \neg B_{1,1} . & & \\ R_5: & B_{2,1} . & & \end{array}\right\} \textbf{\textit{KB}}$$

  - **Initial State:** KB
  - **Actions:** Inference rules applied to sentences in KB
  - **Transition model**: KB' ← RESULT(KB, rule), where KB' is KB plus all the inferred sentences
  - **Goal:** State (KB) containing the sentence to be proved

$$\boxed{KB \models \alpha_1}$$

Monotonicity of logical systems

- As more information is added to KB, the set of entailed sentences **can only increase** (and **never decrease**)

$$\boxed{\text{if} \quad KB \models \alpha \quad \text{then} \quad KB \wedge \beta \models \alpha\,.}$$

? But what if
$\beta = \neg\alpha$

Deriving expressions from functions

- Given a function *F(P,Q)* in truth table form, find a logic expression for it that uses only V, ∧ and ¬.
- **Idea:** We sum up the "T" rows of the truth table.

Example: XOR function

P	Q	RESULT
T	T	F
T	F	T
F	T	T
F	F	F

F(P,Q) = *???*

Deriving expressions from functions

- Given a function *F(P,Q)* in truth table form, find a logic expression for it that uses only V, ∧ and ¬.
- **Idea:** We sum up the "T" rows of the truth table.

Example: XOR function

P	Q	RESULT
T	T	F
T	F	T
F	T	T
F	F	F

$P \wedge (\neg Q)$
$(\neg P) \wedge Q$

$F(P,Q) = (P \wedge (\neg Q)) \vee ((\neg P) \wedge Q)$

Problem of the inference rules

- Soundness:
  - obviously sound

- Completeness:
  - if the inference rules are inadequate, proof may not be reachable

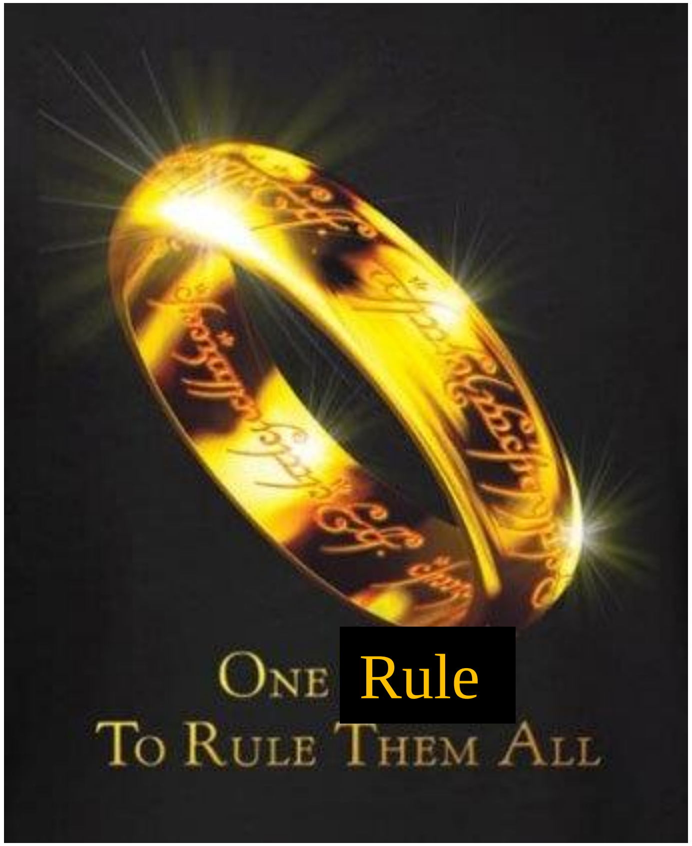
$$
\begin{aligned}
(\alpha \wedge \beta) &\equiv (\beta \wedge \alpha) && \text{commutativity of } \wedge \\
(\alpha \vee \beta) &\equiv (\beta \vee \alpha) && \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) &\equiv (\alpha \wedge (\beta \wedge \gamma)) && \text{associativity of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

Outline

- What is AI?
- Problem-solving agent
  - Uninformed (DFS), informed (A*), and local search
  - Adversarial search (minimax, alpha-beta pruning)
- **Knowledge-based agent**
  - The Wumpus World
  - Propositional Logic
  - **Propositional Logic Inference**
    - Inference and proof
    - Resolution

What is “resolution”?

- Single rule that yields a **complete** inference algorithm



Resolution

- Resolvent
- Pivot variable (*x*)

$$\boxed{(C \lor x) \land (D \lor \neg x) \longrightarrow (C \lor D)}$$

- Resolution can be understood as "either… or…":

Resolution

- Resolvent
- Pivot variable (x)

$$\boxed{(C \vee x) \wedge (D \vee \neg x) \rightarrow (C \vee D)}$$

- Resolution can be understood as "either… or…":
  - When (x = false), the formula equals (C)
  - When (x = true), the formula equals (D)
  - So, regardless, the formula is either (C) or (D)

Resolution

- Resolvent
- Pivot variable (*x*)

$$\boxed{(C \vee x) \wedge (D \vee \neg x) \longrightarrow (C \vee D)}$$

- Resolution can be viewed as "transitivity of implication"

Resolution

- Resolvent
- Pivot variable (x)

$$\boxed{(C \vee x) \wedge (D \vee \neg x) \rightarrow (C \vee D)}$$

- Resolution can be viewed as "transitivity of implication"

$$(\neg C \rightarrow x) \wedge (x \rightarrow D) \qquad = (\neg C \rightarrow D)$$
$$= (C \vee D)$$

Resolution

- Resolvent
- Pivot variable (*x*)

$$\boxed{(C \vee x) \wedge (D \vee \neg x) \rightarrow (C \vee D)}$$

- Example: simple case (unit resolution)

$\boxed{P_{1,1} \vee P_{2,2} \vee P_{3,1}}$ $\boxed{\neg P_{2,2}}$

$$\boxed{P_{1,1} \vee P_{3,1}}$$

Resolution

- Resolvent
- Pivot variable (x)

$$\boxed{(C \vee x) \wedge (D \vee \neg x) \rightarrow (C \vee D)}$$

- Example: simple case (unit resolution = *modus ponens*)

$$\boxed{P_{1,1} \vee P_{2,2} \vee P_{3,1}} \qquad \boxed{\neg P_{2,2}}$$

$$\boxed{P_{1,1} \vee P_{3,1}}$$

$$\boxed{\frac{\alpha \Rightarrow \beta, \qquad \alpha}{\beta}}$$

Resolution

- Resolvent
- Pivot variable (*x*)

$$\boxed{(C \vee x) \wedge (D \vee \neg x) \rightarrow (C \vee D)}$$

- Example: general case

$$\boxed{\frac{P_{1,1} \vee P_{3,1}, \qquad \neg P_{1,1} \vee \neg P_{2,2}}{P_{3,1} \vee \neg P_{2,2}}}$$

Resolution

- Resolvent
- Pivot variable (x)

$$\boxed{(C \vee x) \wedge (D \vee \neg x) \rightarrow (C \vee D)}$$

- Example: special cases
  - When $(D = \neg C)$, the resolvent $(C \vee \neg C)$ equals *(true)*
  - When $(x) \wedge (\neg x)$, the resolvent $(\ )$ equals *(false)*

Resolution-based inference

$$\boxed{(C \lor x) \land (D \lor \neg x) \longrightarrow (C \lor D)}$$

- Applying resolution repeatedly can decide SAT/UNSAT
  - If resolution leads to an “empty clause” (false) ➔ UNSAT
  - Else ➔ SAT

- Both **sound** and **complete**

Conjunctive Normal Form (CNF)

- Variable (a symbol whose value is either true or false)
- Literal (either a variable or its negation)
- Clause (a disjunction of literals)
- CNF Formula (a conjunction of clauses)

Examples

$$(\mathbf{a} \vee \neg\mathbf{c}) \wedge (\mathbf{b} \vee \neg\mathbf{c}) \wedge (\neg\mathbf{a} \vee \neg\mathbf{b} \vee \mathbf{c}) \wedge \ldots$$

literal *clause*

CNF formula

- Syntax

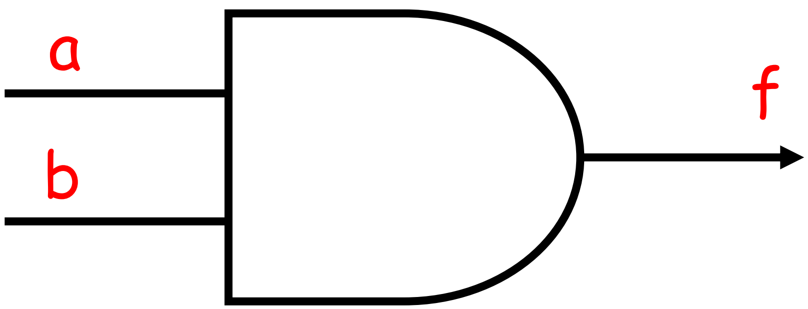
$$
\begin{array}{rcl}
CNFSentence & \rightarrow & Clause_1 \land \cdots \land Clause_n \\
Clause & \rightarrow & Literal_1 \lor \cdots \lor Literal_m \\
Literal & \rightarrow & Symbol \mid \neg Symbol \\
Symbol & \rightarrow & P \mid Q \mid R \mid \ldots
\end{array}
$$

Converting to CNF

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

Equivalence rules…

Converting to CNF (example)



Question: Is (f) satisfiable?

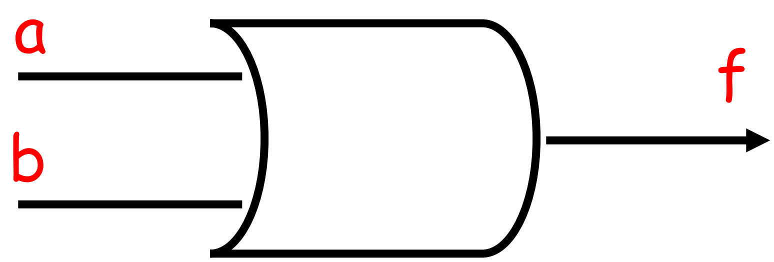
Are there values of (a) and (b) such that (f) is TRUE

Equivalent formula:

$$(\neg a \rightarrow \neg f) \wedge$$
$$(\neg b \rightarrow \neg f) \wedge$$
$$(a \wedge b \rightarrow f)$$

$$\boxed{(a \vee \neg f) \wedge (b \vee \neg f) \wedge (\neg a \vee \neg b \vee f)}$$

Converting to CNF (example)



Question: Is (f) satisfiable?

Are there values of (a) and (b) such that (f) is TRUE

Equivalent CNF formula:

$(a \rightarrow f) \wedge$

$(b \rightarrow f) \wedge$

$(\neg a \wedge \neg b \rightarrow \neg f)$

$\boxed{(\neg a \vee f) \wedge (\neg b \vee f) \wedge (a \vee b \vee \neg f)}$

Conjunctive Normal Form (re-cap)

- Boolean Variable
- Literal
- Clause
- CNF

Examples

$$(\mathbf{a} \vee \neg\mathbf{c}) \wedge (\mathbf{b} \vee \neg\mathbf{c}) \wedge (\neg\mathbf{a} \vee \neg\mathbf{b} \vee \mathbf{c}) \wedge \ldots$$

literal *clause*

Checking entailment (theory)

- Can be done by checking "**validity**" or "**unsatisfiability**"

For any sentences α *and* β, $\alpha \models \beta$

if and only if the sentence $(\alpha \Rightarrow \beta)$ *is valid.*

if and only if the sentence $(\alpha \wedge \neg\beta)$ *is unsatisfiable.*

Checking unsatisfiability

For any sentences α *and* β, $\alpha \models \beta$

$\boxed{\textit{if and only if the sentence } (\alpha \land \neg\beta) \textit{ is unsatisfiable.}}$

- Example: $\boxed{KB \models \alpha_1}$

$$\left.\begin{aligned}
R_1 &: \quad \neg P_{1,1} . \\
R_2 &: \quad B_{1,1} \quad \Leftrightarrow \quad (P_{1,2} \lor P_{2,1}) . \\
R_3 &: \quad B_{2,1} \quad \Leftrightarrow \quad (P_{1,1} \lor P_{2,2} \lor P_{3,1}) . \\
R_4 &: \quad \neg B_{1,1} . \\
R_5 &: \quad B_{2,1} .
\end{aligned}\right\} \boxed{KB}$$

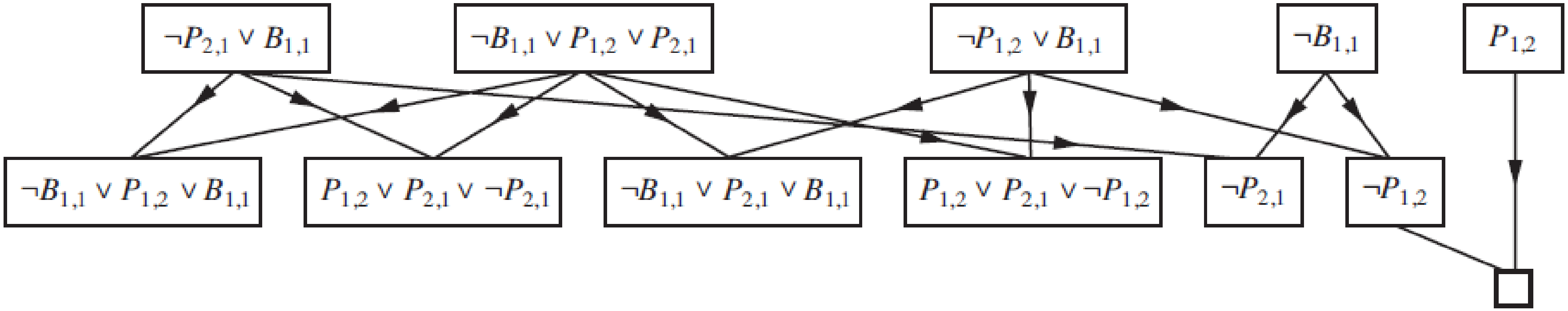
$\boxed{\alpha_1}$ = "There is no pit in [1,2]."

- $\boxed{(R_1 \land R_2 \land R_3 \land R_4 \land R_5)} \land \neg \boxed{(\neg P_{1,2})}$

Applying resolution

$\boxed{\neg P_{2,1} \vee B_{1,1}}$ $\boxed{\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}}$ $\boxed{\neg P_{1,2} \vee B_{1,1}}$ $\boxed{\neg B_{1,1}}$ $\boxed{P_{1,2}}$

Applying resolution



unsatisfiable

For any sentences α *and* β, $\alpha \models \beta$

if and only if the sentence $(\alpha \wedge \neg\beta)$ *is unsatisfiable.*

$\boxed{KB \models \alpha_1}$

Resolution algorithm

```
function PL-RESOLUTION(KB, α) returns true or false
   inputs: KB, the knowledge base, a sentence in propositional logic
           α, the query, a sentence in propositional logic

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

Resolution Proof (example)

$$(\neg a \vee \neg b \vee c) \wedge (a \vee b \vee c) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \wedge$$
$$(\neg a \vee c \vee d) \wedge (\neg a \vee b \vee \neg d) \wedge (b \vee c \vee \neg d) \wedge (a \vee b \vee d)$$

If we eliminate the variables in alphabetic order,

$$\begin{aligned} S'_0 =& (\neg a \vee \neg b \vee c) \wedge (a \vee b \vee c) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \wedge \\ & (\neg a \vee c \vee d) \wedge (\neg a \vee b \vee \neg d) \wedge (b \vee c \vee \neg d) \wedge (a \vee b \vee d) \\ S'_1 =& (b \vee c \vee d) \wedge (b \vee c \vee \neg d) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \end{aligned}$$

Resolution Proof (example)

$$\boxed{\begin{aligned}&(\neg a \vee \neg b \vee c) \wedge (a \vee b \vee c) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \wedge \\ &\qquad (\neg a \vee c \vee d) \wedge (\neg a \vee b \vee \neg d) \wedge (b \vee c \vee \neg d) \wedge (a \vee b \vee d)\end{aligned}}$$

If we eliminate the variables in alphabetic order,

$$\begin{aligned}S'_0 =& (\neg a \vee \neg b \vee c) \wedge (a \vee b \vee c) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \wedge \\ & (\neg a \vee c \vee d) \wedge (\neg a \vee b \vee \neg d) \wedge (b \vee c \vee \neg d) \wedge (a \vee b \vee d) \\ S'_1 =& (b \vee c \vee d) \wedge (b \vee c \vee \neg d) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \\ S'_2 =& (\neg c \vee d) \wedge (\neg c \vee \neg d) \\ S'_3 =& \text{true} \\ S'_4 =& \text{true} \quad .\end{aligned}$$

Resolution Proof (compute assignment)

$$(\neg a \vee \neg b \vee c) \wedge (a \vee b \vee c) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \wedge \\ (\neg a \vee c \vee d) \wedge (\neg a \vee b \vee \neg d) \wedge (b \vee c \vee \neg d) \wedge (a \vee b \vee d)$$

If we eliminate the variables in alphabetic order,

$$\begin{aligned}
S'_0 =& (\neg a \vee \neg b \vee c) \wedge (a \vee b \vee c) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \wedge \\
& (\neg a \vee c \vee d) \wedge (\neg a \vee b \vee \neg d) \wedge (b \vee c \vee \neg d) \wedge (a \vee b \vee d) \\
S'_1 =& (b \vee c \vee d) \wedge (b \vee c \vee \neg d) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \\
S'_2 =& (\neg c \vee d) \wedge (\neg c \vee \neg d) \\
S'_3 =& \text{true} \\
S'_4 =& \text{true} \quad .
\end{aligned}$$

$$\begin{aligned}
\eta_0 &= \{\neg a, b, \neg c, d\} \\
\eta_1 &= \{b, \neg c, d\} \\
\eta_2 &= \{\neg c, d\} \\
\eta_3 &= \{d\} \\
\eta_4 &= \emptyset
\end{aligned}$$

Resolution Proof (an alternative assignment)

$$(\neg a \vee \neg b \vee c) \wedge (a \vee b \vee c) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \wedge$$
$$(\neg a \vee c \vee d) \wedge (\neg a \vee b \vee \neg d) \wedge (b \vee c \vee \neg d) \wedge (a \vee b \vee d)$$

If we eliminate the variables in alphabetic order,

$$\begin{aligned}
S'_0 =& (\neg a \vee \neg b \vee c) \wedge (a \vee b \vee c) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \wedge \\
& (\neg a \vee c \vee d) \wedge (\neg a \vee b \vee \neg d) \wedge (b \vee c \vee \neg d) \wedge (a \vee b \vee d) \\
S'_1 =& (b \vee c \vee d) \wedge (b \vee c \vee \neg d) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \\
S'_2 =& (\neg c \vee d) \wedge (\neg c \vee \neg d) \\
S'_3 =& \text{true} \\
S'_4 =& \text{true} \quad .
\end{aligned}$$

$$\begin{aligned}
\eta_0 &= \{\neg a, b, \neg c, d\} \\
\eta_1 &= \{b, \neg c, d\} \\
\eta_2 &= \{\neg c, d\} \\
\eta_3 &= \{d\} \\
\eta_4 &= \emptyset
\end{aligned}$$

Resolution Proof (another example)

$$\boxed{\begin{aligned}&(\neg a \vee \neg b \vee c) \wedge (a \vee \neg b \vee c) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \wedge \\ &\qquad (\neg a \vee c \vee d) \wedge (\neg a \vee b \vee \neg d) \wedge (b \vee c \vee \neg d) \wedge (a \vee b \vee d)\end{aligned}}$$

If we eliminate the variables in alphabetic order,

$$\begin{aligned}
S'_0 =& (\neg a \vee \neg b \vee c) \wedge (a \vee \neg b \vee c) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \wedge \\
& (\neg a \vee c \vee d) \wedge (\neg a \vee b \vee \neg d) \wedge (b \vee c \vee \neg d) \wedge (a \vee b \vee d) \\
S'_1 =& (\neg b \vee c) \wedge (\neg b \vee c \vee d) \wedge (b \vee c \vee d) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \wedge (b \vee c \vee \neg d) \\
S'_2 =& (c \vee d) \wedge (c \vee \neg d) \wedge (\neg c \vee d) \wedge (\neg c \vee \neg d) \\
S'_3 =& d \wedge \neg d \\
S'_4 =& \text{false} \ .
\end{aligned}$$

The CNF formula is therefore unsatisfiable.

Outline

- What is AI?
- Problem-solving agent
  - Uninformed (DFS), informed (A*), and local search
  - Adversarial search (minimax, alpha-beta pruning)
- **Knowledge-based agent**
  - The Wumpus World
  - Propositional Logic
  - **Propositional Logic Inference**
    - Inference and proof
    - Resolution
    - Horn clause

Horn clause

- A disjunction of literals of which "**at most one**" is positive
  - Exactly one literal is positive (definite clause)

$$\boxed{(\neg L_{1,1} \vee \neg Breeze \vee B_{1,1})} \qquad (L_{1,1} \wedge Breeze) \Rightarrow B_{1,1}$$

$$\boxed{L_{1,1}} \qquad True \Rightarrow L_{1,1}$$

  - None is positive (goal clause)

Horn clause resolution

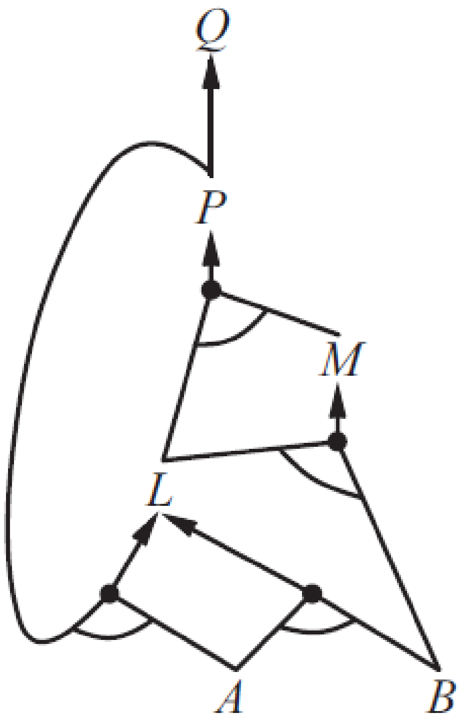
- *Resolution algorithm* *(KB and Q)*

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$

Horn clause resolution

- Forward chaining

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Forward chaining algorithm

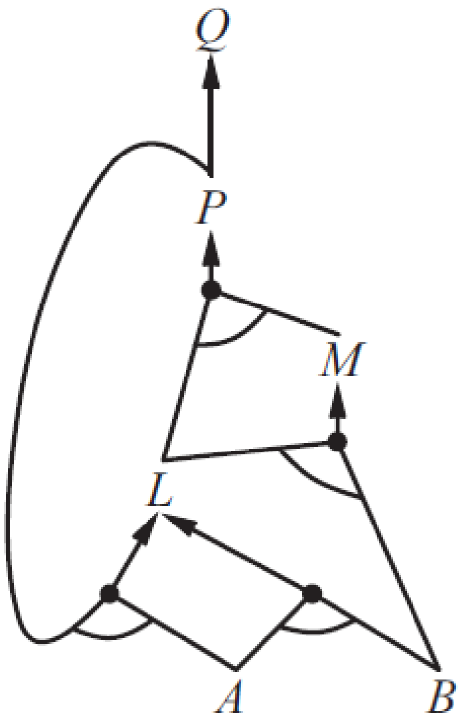
```
function PL-FC-ENTAILS?(KB, q) returns true or false
   inputs: KB, the knowledge base, a set of propositional definite clauses
           q, the query, a proposition symbol
   count ← a table, where count[c] is the number of symbols in c's premise
   inferred ← a table, where inferred[s] is initially false for all symbols
   agenda ← a queue of symbols, initially symbols known to be true in KB

   while agenda is not empty do
      p ← POP(agenda)
      if p = q then return true
      if inferred[p] = false then
         inferred[p] ← true
         for each clause c in KB where p is in c.PREMISE do
            decrement count[c]
            if count[c] = 0 then add c.CONCLUSION to agenda
   return false
```

Horn clause resolution

- Backward chaining

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Outline

- What is AI?
- Problem-solving agent
  - Uninformed (DFS), informed (A*), and local search
  - Adversarial search (minimax, alpha-beta pruning)
- **Knowledge-based agent**
  - The Wumpus World
  - Propositional Logic
  - **Propositional Logic Inference**
    - Inference and proof
    - Resolution
    - Horn clause
    - Examples

Inference example

Assumption:

- It is not sunny this afternoon and it is colder than yesterday.
- We will go swimming only if it is sunny.
- If we do not go swimming, then we will take a canoe trip.
- If we take a canoe trip, then we will be home by sunset.

Conclusion:

- We will be home by the sunset.

Inference example

Assumption:

- It is not sunny this afternoon and it is colder than yesterday. $\neg s \wedge c$
- We will go swimming only if it is sunny. $w \rightarrow s$
- If we do not go swimming, then we will take a canoe trip. $\neg w \rightarrow t$
- If we take a canoe trip, then we will be home by sunset. $t \rightarrow h$

Conclusion:

- We will be home by the sunset. h

Where:
s: "it is sunny this afternoon"
c: "it is colder than yesterday"
w: "we will go swimming"
t: "we will take a canoe trip.
h: "we will be home by the sunset."

Inference example

Assumption:

- It is not sunny this afternoon and it is colder than yesterday. $\neg s \wedge c$
- We will go swimming only if it is sunny. $w \rightarrow s$
- If we do not go swimming, then we will take a canoe trip. $\neg w \rightarrow t$
- If we take a canoe trip, then we will be home by sunset. $t \rightarrow h$

Conclusion:

- We will be home by the sunset. h

Step	Reason
1. $\neg s \wedge c$	hypothesis

Where:
s: "it is sunny this afternoon"
c: "it is colder than yesterday"
w: "we will go swimming"
t: "we will take a canoe trip.
h: "we will be home by the sunset."

Inference example

Assumption:

- It is not sunny this afternoon and it is colder than yesterday. $\neg s \wedge c$
- We will go swimming only if it is sunny. $w \rightarrow s$
- If we do not go swimming, then we will take a canoe trip. $\neg w \rightarrow t$
- If we take a canoe trip, then we will be home by sunset. $t \rightarrow h$

Conclusion:

- We will be home by the sunset. h

Step	Reason
1. $\neg s \wedge c$	hypothesis
2. $\neg s$	simplification

Where:
s: "it is sunny this afternoon"
c: "it is colder than yesterday"
w: "we will go swimming"
t: "we will take a canoe trip.
h: "we will be home by the sunset."

Inference example

Assumption:

- It is not sunny this afternoon and it is colder than yesterday. $\neg s \wedge c$
- We will go swimming only if it is sunny. $w \rightarrow s$
- If we do not go swimming, then we will take a canoe trip. $\neg w \rightarrow t$
- If we take a canoe trip, then we will be home by sunset. $t \rightarrow h$

Conclusion:

- We will be home by the sunset. h

Step	Reason
1. $\neg s \wedge c$	hypothesis
2. $\neg s$	simplification
3. $w \rightarrow s$	hypothesis

Where:
s: "it is sunny this afternoon"
c: "it is colder than yesterday"
w: "we will go swimming"
t: "we will take a canoe trip.
h: "we will be home by the sunset."

Inference example

Assumption:

- It is not sunny this afternoon and it is colder than yesterday. $\neg s \wedge c$
- We will go swimming only if it is sunny. $w \rightarrow s$
- If we do not go swimming, then we will take a canoe trip. $\neg w \rightarrow t$
- If we take a canoe trip, then we will be home by sunset. $t \rightarrow h$

Conclusion:

- We will be home by the sunset. h

Step	Reason
1. $\neg s \wedge c$	hypothesis
2. $\neg s$	simplification
3. $w \rightarrow s$	hypothesis
4. $\neg w$	modus tollens of 2 and 3

Where:
s: "it is sunny this afternoon"
c: "it is colder than yesterday"
w: "we will go swimming"
t: "we will take a canoe trip.
h: "we will be home by the sunset."

Inference example

Assumption:

- It is not sunny this afternoon and it is colder than yesterday. $\neg s \wedge c$
- We will go swimming only if it is sunny. $w \rightarrow s$
- If we do not go swimming, then we will take a canoe trip. $\neg w \rightarrow t$
- If we take a canoe trip, then we will be home by sunset. $t \rightarrow h$

Conclusion:

- We will be home by the sunset. h

Step	Reason
1. $\neg s \wedge c$	hypothesis
2. $\neg s$	simplification
3. $w \rightarrow s$	hypothesis
4. $\neg w$	modus tollens of 2 and 3
5. $\neg w \rightarrow t$	hypothesis

Where:
s: "it is sunny this afternoon"
c: "it is colder than yesterday"
w: "we will go swimming"
t: "we will take a canoe trip.
h: "we will be home by the sunset."

Inference example

Assumption:

- It is not sunny this afternoon and it is colder than yesterday. $\neg s \wedge c$
- We will go swimming only if it is sunny. $w \rightarrow s$
- If we do not go swimming, then we will take a canoe trip. $\neg w \rightarrow t$
- If we take a canoe trip, then we will be home by sunset. $t \rightarrow h$

Conclusion:

- We will be home by the sunset. h

Step	Reason
1. $\neg s \wedge c$	hypothesis
2. $\neg s$	simplification
3. $w \rightarrow s$	hypothesis
4. $\neg w$	modus tollens of 2 and 3
5. $\neg w \rightarrow t$	hypothesis
6. t	modus ponens of 4 and 5

Where:
s: "it is sunny this afternoon"
c: "it is colder than yesterday"
w: "we will go swimming"
t: "we will take a canoe trip.
h: "we will be home by the sunset."

Inference example

Assumption:

- It is not sunny this afternoon and it is colder than yesterday. $\neg s \wedge c$
- We will go swimming only if it is sunny. $w \rightarrow s$
- If we do not go swimming, then we will take a canoe trip. $\neg w \rightarrow t$
- If we take a canoe trip, then we will be home by sunset. $t \rightarrow h$

Conclusion:

- We will be home by the sunset. h

Step	Reason
1. $\neg s \wedge c$	hypothesis
2. $\neg s$	simplification
3. $w \rightarrow s$	hypothesis
4. $\neg w$	modus tollens of 2 and 3
5. $\neg w \rightarrow t$	hypothesis
6. t	modus ponens of 4 and 5
7. $t \rightarrow h$	hypothesis

Where:
s: "it is sunny this afternoon"
c: "it is colder than yesterday"
w: "we will go swimming"
t: "we will take a canoe trip.
h: "we will be home by the sunset."

Inference example

Assumption:

- It is not sunny this afternoon and it is colder than yesterday. $\neg s \wedge c$
- We will go swimming only if it is sunny. $w \rightarrow s$
- If we do not go swimming, then we will take a canoe trip. $\neg w \rightarrow t$
- If we take a canoe trip, then we will be home by sunset. $t \rightarrow h$

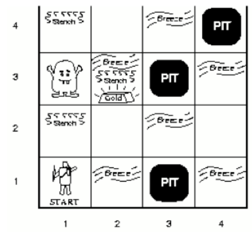
Conclusion:

- We will be home by the sunset. h

Step	Reason
1. $\neg s \wedge c$	hypothesis
2. $\neg s$	simplification
3. $w \rightarrow s$	hypothesis
4. $\neg w$	modus tollens of 2 and 3
5. $\neg w \rightarrow t$	hypothesis
6. t	modus ponens of 4 and 5
7. $t \rightarrow h$	hypothesis
8. h	modus ponens of 6 and 7

Where:
s: "it is sunny this afternoon"
c: "it is colder than yesterday"
w: "we will go swimming"
t: "we will take a canoe trip.
h: "we will be home by the sunset."

Wumpus world: example



- **Facts:** Percepts inject (TELL) facts into the KB
  - [no stench at 1,1 and 2,1] → $\neg S_{1,1}$; $\neg S_{2,1}$
- **Rules:** if square has no stench then neither the square or adjacent squares contain the Wumpus
  - R1: $\neg S_{1,1} \Rightarrow \neg W1,1 \wedge \neg W1,2 \wedge \neg W_{2,1}$
  - R2: $\neg S_{2,1} \Rightarrow \neg W1,1 \wedge \neg W2,1 \wedge \neg W2,2 \wedge \neg W3,1$
  - …
- **Inference:**
  - KB contains $\neg S_{1,1}$ then using ***Modus Ponens*** we infer $\neg W1,1 \wedge \neg W1,2 \wedge \neg W2,1$
  - Using ***And-Elimination*** we get: $\neg W1,1$ $\neg W1,2$ $\neg W2,1$,
  - KB contains $\neg S_{2,1}$ then …

Limitations of Propositional Logic

1. It can be too weak, i.e., has limited expressiveness:

- Each rule has to be represented for each situation:
  e.g., "don't go forward if the wumpus is in front of you" takes 64 rules

2. It cannot keep track of changes:

- If one needs to track changes, e.g., where the agent has been before then we need a timed-version of each rule. To track 100 steps we'll then need 6400 rules for the previous example.

Its **hard to write and maintain** such a huge rule-base
Inference becomes intractable

Outline

- What is AI?
- Problem-solving agent
  - Uninformed (DFS), informed (A*), and local search
  - Adversarial search (minimax, alpha-beta pruning)
- **Knowledge-based agent**
  - The Wumpus World
  - Propositional Logic
  - **Propositional Logic Inference**
    - Inference and proof
    - Resolution
    - Horn clause
    - Examples