Lecture 8
Logical Inference

CS 440: Intro AI
Jingjin Yu | Rutgers
Review: Simultaneous Two-Player Games

- Simultaneous 2-player games
  - A game in normal form has a payoff matrix
  - Zero-sum if each one player’s gain equals the opponent’s loss
  - A dominant strategy for a player is one with which the player cannot do better by switching to other strategies regardless of opponent’s choice
  - A strategy yields a Nash equilibrium if a player cannot do better by switching his/her own strategy alone
  - Nash equilibrium always exists for games in normal form with finite payoff matrix
    - The strategy may be a mixed strategy

<table>
<thead>
<tr>
<th></th>
<th>Player 1</th>
<th></th>
<th>Player 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rock</td>
<td>Scissors</td>
<td>Paper</td>
</tr>
<tr>
<td>Rock</td>
<td>0,0</td>
<td>-1,1</td>
<td>1,-1</td>
</tr>
<tr>
<td>Scissors</td>
<td>1,-1</td>
<td>0,0</td>
<td>-1,1</td>
</tr>
<tr>
<td>Paper</td>
<td>-1,1</td>
<td>1,-1</td>
<td>0,0</td>
</tr>
</tbody>
</table>

Source: wikipedia

Player 2 actions

- Scissors beats paper
- Rock beats scissors
- Paper beats rock

Payoff matrix

Source: wikipedia
Why Logic?

What have we studied in the course so far?
- Uninformed (tree/graph) search, including DFS, BFS, UC, iterative deepening, ...
- Informed search: greedy best-first and A*
- Local search: hill-climbing, beam search, simulated-annealing, ...
- Constraint satisfaction problems
- Adversarial search: MinMax + alpha-beta pruning

Common characteristics among these methods?
- All of these problems have “nice” state space
  - States usually have physical meanings
    - Searching for a route – a state is a location
    - CSP – a state is an assignment, possibly invalid
    - Adversarial search – a state is a possible step in a game search tree
  - Such states are domain-specific
    - The states from one problem is drastically different from states from another
    - Not much “transferrable knowledge”
Why Logic?

- Logical inference aims for a more general approach
- Seeks to mimic how humans solve problems
  - We learn and obtain knowledge
  - We then use the knowledge we learned to reason
  - Knowledge learned in one domain appears applicable to other domains
    - I.e., people working in one area can also work in other areas
    - This is still being argued
  - How can computer algorithms be more domain independent?
  - Perhaps through some form of logical inference

- Logic: the study of valid reasoning and its applications
- The hope: equip computers with higher level of cognition capabilities
The General Approach

- **Knowledge base (KB):** “sentences” – statements assumed to be true
- A KB-agent can perform **inference** on the KB to obtain additional true statements
- A KB-agent may be **queried** on whether some statement is true
- In this lecture
  - The wumpus world example
  - Propositional logic
  - Briefly, First order logic and Gödel's incompleteness theorem
The Wumpus World

Who does not enjoy shower

START

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

Who does not enjoy shower

START

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!

PIT

Breeze

Stench

WUMPUS

Gold !!!
The Wumpus World

⇒ **Environment:** $4 \times 4$ grid of rooms, with locations of gold and wumpus chosen randomly, and pits have 20% chance to appear

⇒ **Sensors**
  - Stench
  - Breeze
  - Gold glitter
  - Wall bump
  - Wumpus screams when killed

⇒ **Possible actions:**
  - Forward, left turn, right turn
  - Shoot arrow

⇒ **Performance measure:**
  - 1000 for gold
  - -1000 for getting killed (by wumpus or fall in pit)
  - -1 for each action
  - -10 for using the arrow

[stench, none, none, none, none] or [s,n,n,n,n]
Inference over The Wumpus World

<table>
<thead>
<tr>
<th>B = breeze</th>
<th>G = glitter</th>
<th>OK = safe square</th>
<th>P = pit</th>
<th>S = stench</th>
<th>V = visited</th>
<th>W = wumpus</th>
</tr>
</thead>
</table>

Legend

<table>
<thead>
<tr>
<th>1,1</th>
<th>2,1</th>
<th>3,1</th>
<th>4,1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2</td>
<td>2,2</td>
<td>3,2</td>
<td>4,2</td>
</tr>
<tr>
<td>1,3</td>
<td>2,3</td>
<td>3,3</td>
<td>4,3</td>
</tr>
<tr>
<td>1,4</td>
<td>2,4</td>
<td>3,4</td>
<td>4,4</td>
</tr>
</tbody>
</table>
Inference over The Wumpus World

Legend
B = breeze
G = glitter
OK = safe square
P = pit
S = stench
V = visited
W = wumpus

⇒ Agent starts at [1,1], assumed to be OK
Inference over The Wumpus World

Legend

B = breeze
G = glitter
OK = safe square
P = pit
S = stench
V = visited
W = wumpus

⇒ Agent starts at [1,1], assumed to be OK
⇒ Senses: [n,n,n,n,n,n]
⇒ Infers: [1,2], [2,1] are both OK
Inference over The Wumpus World

Legend

B = breeze
G = glitter
OK = safe square
P = pit
S = stench
V = visited
W = wumpus

⇒ Agent decides to move to the right
Inference over The Wumpus World

Legend
B = breeze
G = glitter
OK = safe square
P = pit
S = stench
V = visited
W = wumpus

⇒ [1,1] marked as visited
⇒ Senses \([n,b,n,n,n]\)
⇒ Infers that [2,2] and [3,1] may be pits
Inference over The Wumpus World

Legend

B = breeze
G = glitter
OK = safe square
P = pit
S = stench
V = visited
W = wumpus

⇒ Agent decides to move to [1, 2]
Inference over The Wumpus World

⇒ Senses [s,n,n,n,n,n]
⇒ Infers [2,2] is not pit, [3,1] must be pit
⇒ Infers [1,3] has wumpus
## Inference over The Wumpus World

### Legend

- **B** = breeze
- **G** = glitter
- **OK** = safe square
- **P** = pit
- **S** = stench
- **V** = visited
- **W** = wumpus

---

### Grid

<table>
<thead>
<tr>
<th>1,1</th>
<th>2,1</th>
<th>3,1</th>
<th>4,1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>V, OK</strong></td>
<td><strong>B</strong></td>
<td><strong>V, OK</strong></td>
<td><strong>P</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1,2</th>
<th>2,2</th>
<th>3,2</th>
<th>4,2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1,3</th>
<th>2,3</th>
<th>3,3</th>
<th>4,3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>W</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1,4</th>
<th>2,4</th>
<th>3,4</th>
<th>4,4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

⇒ Agent can move to [2,2]
Inference over The Wumpus World

### Legend

- **B** = breeze
- **G** = glitter
- **OK** = safe square
- **P** = pit
- **S** = stench
- **V** = visited
- **W** = wumpus

⇒ Senses\([n,n,n,n,n]\)
⇒ Infers \([2,3]\) and \([3,2]\) are OK
⇒ Agent moves to \([2,3]\) and finds gold
Inference over The Wumpus World

→ We have used reasoning to build up our knowledge base

→ The conclusion is guaranteed to be correct if we reason using correct information

→ Can we do this more formally with an algorithmic method?
Components of a Logic System

- **Syntax** defines what are valid sentences
  - E.g., $x + y$ is not a valid equation

- **Semantics** gives valid sentences meanings
  - E.g., “that guy is rather orgulous”
  - If you do not know what “orgulous” means, the sentence is useless
  - In logic, this is done by assigning sentences to be true or false

- **Model** and possible worlds – the truth value of a sentence is not absolute. A particular set of assignments of truth values to sentences form a possible world or a model

- **Entailment** ($\models$): $\alpha \models \beta$ says that $\beta$ follows logically from $\alpha$

- **Soundness**: an inference algorithm is sound if it only derives entailed sentences

- **Completeness**: an inference algorithm is complete if all entailed sentences can be derived using the algorithm
Propositional Logic (Boolean Logic) Syntax

⇒ Operates over binary propositions
  ⇒ Each proposition is either true (True) or false (False)

⇒ Operands
  ¬ : not, negation, \( x = true \) then \( \neg x = false \)
  ∧ : and, conjunction, \( x \land y = true \) if and only if \( x = true, y = true \)
  ∨ : or, disjunction, \( x \lor y = true \) if one or both of \( x \) and \( y \) is/are true
  ⇒ : implies
  ⇔ : if and only if

⇒ Order of operands: \( \neg, \land, \lor, \Rightarrow, \Leftrightarrow \)

⇒ Parentheses should be used to avoid ambiguity
  ⇒ E.g., \((A \lor B) \land (C \lor D)\) evaluates \((A \lor B)\) and \((C \lor D)\) first
Propositional Logic Syntax

Syntax of propositional logic

<table>
<thead>
<tr>
<th>Sentence</th>
<th>AtomicSentence</th>
<th>ComplexSentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>AtomicSentence</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>ComplexSentence</td>
<td>Sentence</td>
<td>Sentence</td>
</tr>
<tr>
<td></td>
<td>¬</td>
<td>Sentence</td>
</tr>
<tr>
<td></td>
<td>Sentence ∧ Sentence</td>
<td>Sentence</td>
</tr>
<tr>
<td></td>
<td>Sentence</td>
<td>Sentence</td>
</tr>
</tbody>
</table>

“A sentence can be an atomic sentence or a complex sentence

E.g. \((P \lor Q) \Rightarrow (R \lor S)\)

⇒ Propositions \(P, Q, R, S\) are atomic sentences
⇒ \((P \lor Q)\) is a complex sentence
⇒ So is \((R \lor S)\) and \((P \lor Q) \Rightarrow (R \lor S)\)
Propositional Logic Semantics

Semantics tells us how to interpret a sentence

For propositional logic (a world or model is always assumed)

- *true* is always true and *false* is always false
- Other propositions, e.g., *P*, *Q*, *R*, must be given true or false values
- For complex sentences
  - \( \neg P \) is true iff (reads if and only if) \( P \) is false
  - \( P \land Q \) is true iff both \( P \) and \( Q \) are true
  - \( P \lor Q \) is true iff either \( P \) or \( Q \) is true
  - \( P \Rightarrow Q \) is true unless \( P \) is true and \( Q \) is false
  - \( P \Leftrightarrow Q \) is true iff \( P \) and \( Q \) are both true or both false

In truth table form

<table>
<thead>
<tr>
<th>( P )</th>
<th>( Q )</th>
<th>( \neg P )</th>
<th>( P \land Q )</th>
<th>( P \lor Q )</th>
<th>( P \Rightarrow Q )</th>
<th>( P \Leftrightarrow Q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>
A Wumpus World Knowledge Base

\[ P_{i,j} : \text{true if square } [i, j] \text{ has a pit} \]

\[ W_{i,j} : \text{true if square } [i, j] \text{ has the wumpus} \]

\[ B_{i,j} : \text{true if square } [i, j] \text{ has breeze} \]

\[ S_{i,j} : \text{true if square } [i, j] \text{ has stench} \]

\( \Rightarrow \) No Pit in square \([1,1]\)

\[ R_1 : \neg P_{1,1} \]

\( \Rightarrow \) A square is breezy if having a neighboring pit

\[ R_2 : B_{1,1} \iff (P_{1,2} \lor P_{2,1}) \]

\[ R_3 : B_{2,1} \iff (P_{1,1} \lor P_{2,2} \lor P_{3,1}) \]

\( \ldots \)

\( \Rightarrow \) Observations

\[ R_4 : \neg B_{1,1} \]

\[ R_5 : B_{2,1} \]

\( \ldots \)
Inference in Propositional Logic

⇒ To add more to the KB, use the **deduction theorem**:

\[ \alpha \equiv \beta \text{ if and only if } \alpha \Rightarrow \beta \]

⇒ **Inference rules**

⇒ **Modus Ponens**: if \( \alpha \) is true and \( \alpha \Rightarrow \beta \), then \( \beta \) is true

\[ \frac{\alpha \Rightarrow \beta, \ \alpha}{\beta} \]

⇒ **And-Elimination**: if \( \alpha \land \beta \) is true, then \( \alpha \) (as well as \( \beta \)) must be true

\[ \frac{\alpha \land \beta}{\alpha} \]

⇒ More complex rules can also be composed, e.g., **bidirectional elimination**:

\[ \frac{\alpha \equiv \beta}{(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)} \]

\[ \frac{(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)}{\alpha \equiv \beta} \]
Inference over Wumpus KB

⇒ Apply bidirectional elimination to $R_2$

$R_6: (B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1})$

⇒ Followed by And-Elimination

$R_7: (P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1}$

⇒ This is the same as (contra positive)

$R_8: \neg B_{1,1} \Rightarrow \neg(P_{1,2} \lor P_{2,1})$

⇒ Modus Ponens of $R_8$ and $R_4$

$R_9: \neg(P_{1,2} \lor P_{2,1})$

⇒ De Morgan over $R_9$

$R_{10}: \neg P_{1,2} \land \neg P_{2,1}$

⇒ $R_{10}$ says that [1,2] and [2,1] cannot be pits

⇒ Our first logical inference!
Unit Resolution

⇒ Now add

\[ R_{11}: \neg B_{1,2}, \ R_{12}: B_{1,2} \iff (P_{1,1} \lor P_{2,2} \lor P_{1,3}) \]

⇒ Using these, we can infer (as we did on last slide)

\[ R_{13}: \neg P_{2,2}, \ R_{14}: \neg P_{1,3} \]

⇒ Bidirectional elimination to \( R_3 \), Modus Ponens to \( R_5 \), we get

\[ R_{15}: P_{1,1} \lor P_{2,2} \lor P_{3,1} \]

⇒ Now apply **unit resolution** \((m < n)\)

\[
\begin{array}{c}
\ell_1 \lor \cdots \lor \ell_n \\
\ell_1 \lor \cdots \ell_{m-1} \lor \ell_{m+1} \lor \cdots \lor \ell_n \\
\ell_m
\end{array}
\]

⇒ Now apply unit resolution \((m < n)\) to \( R_{15} \) and \( R_{13} \)

\[ R_{16}: P_{1,1} \lor P_{3,1} \]

⇒ Resolution again with \( R_1 \) yields \([3, 1]\) is a pit!

\[ R_{16}: P_{3,1} \]

⇒ Extends readily to more general resolution rule

---

\(\text{KB}\)

\[ R_1: \neg P_{1,1} \]
\[ R_2: B_{1,1} \iff (P_{1,2} \lor P_{2,1}) \]
\[ R_3: B_{2,1} \iff (P_{1,1} \lor P_{2,2} \lor P_{3,1}) \]
\[ R_4: \neg B_{1,1} \]
\[ R_5: B_{2,1} \]

**And-Elimination**

\[
\frac{\alpha \land \beta}{\alpha}
\]

**Modus Ponens**

\[
\frac{\alpha \Rightarrow \beta, \ \alpha}{\beta}
\]

**Bidirectional elimination**

\[
\frac{\alpha \iff \beta}{(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)}
\]

\[
\frac{(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)}{\alpha \iff \beta}
\]
Soundness and Completeness

⇒ Propositional logic is **sound**
  ⇒ Everything we inferred is entailment (i.e., correct)

⇒ It is also **complete**
  ⇒ Can be proven

⇒ It is possible to formulate the inference as a search algorithm
  ⇒ However, complexity is still high
  ⇒ Deciding entailment is co-NP-complete
  ⇒ Believed to be as hard as NP-complete
A Little Bit on First-Order Logic

⇒ Propositional logic is fairly limited
  ⇒ We must specify each object (proposition) individually
    ⇒ We have to explicitly define $P_{i,j}$ for all $i, j$ pairs
  ⇒ Not very expressive

⇒ First order logic addresses some of the limitations
  ⇒ Everything from propositional logic
  ⇒ Adds variables $x_1, x_2, \ldots$
  ⇒ Adds equality $=$
  ⇒ Adds predicates, $P(\ldots)$, taking 0+ variables
    ⇒ E.g., $Red(Ball)$
  ⇒ Adds functions, $F(\ldots)$, taking 0+ variables
    ⇒ E.g., $Father(x) = y$
  ⇒ Adds quantifiers $\exists$ (exists) and $\forall$ (for all)
    ⇒ Allows assigning property to more than one thing at a time
    ⇒ E.g., $\forall x. Red(x)$

⇒ Much higher computational complexity, but nevertheless sound and complete
Gödel's Incompleteness Theorem

⇒ First order logic is sound and complete
  ⇒ However still limited in expressivity
  ⇒ E.g., does not support elementary arithmetic

⇒ However, first order logic + arithmetic cannot be both consistent and complete
  ⇒ A logical inference system is consistent if it does not contain contradictions
  ⇒ E.g., cannot contain both $A$ and $\neg A$
  ⇒ Gödel’s first incompleteness theorem says that first order logic + arithmetic is incomplete – there are statements in the system that cannot be proved or disproved.