Logical Agents

Outline

I. Knowledge-based agents

II. The Wumpus world

III. Logic

IV. Syntax of proportional logic

* Figures/images are from the textbook site unless sources are specifically cited.
I. Knowledge-Based Agents

- Problem solving agents do not know general facts.

  An 8-puzzle agent does not know that two tiles cannot occupy the same space.

- Their atomic representations are very limited.

  e.g., a list of all possible concrete states.
I. Knowledge-Based Agents

- Problem solving agents do not know general facts.
  
  An 8-puzzle agent does not know that two tiles cannot occupy the same space.

- Their atomic representations are very limited.
  
  e.g., a list of all possible concrete states.

- Intelligent agents need *knowledge about the world* in order to carry out reasoning for good decision making.
I. Knowledge-Based Agents

- Problem solving agents do not know general facts.
  
  An 8-puzzle agent does not know that two tiles cannot occupy the same space.

- Their atomic representations are very limited.
  
  e.g., a list of all possible concrete states.

- Intelligent agents need *knowledge about the world* in order to carry out reasoning for good decision making.
  
  - Represent states, actions, etc.
  - Incorporate new percepts.
  - Update internal representation of the world.
  - Deduce hidden properties of the world.
  - Deduce appropriate actions.
A *knowledge base (KB)* is a set of sentences that represents some assertion about the world.

An *axiom* is such a sentence that is taken to be true without being derived from other sentences.
A *knowledge base (KB)* is a set of sentences that represents some assertion about the world.

An *axiom* is such a sentence that is taken to be true without being derived from other sentences.
A *knowledge base (KB)* is a set of sentences that represents some assertion about the world.

An *axiom* is such a sentence that is taken to be true without being derived from other sentences.

**TELL**: Add new sentences to the KB.
A knowledge base (KB) is a set of sentences that represents some assertion about the world.

An axiom is such a sentence that is taken to be true without being derived from other sentences.

**TELL**: Add new sentences to the KB.

**ASK**: Query the KB.
A *knowledge base (KB)* is a set of sentences that represents some assertion about the world.

An *axiom* is such a sentence that is taken to be true without being derived from other sentences.

**TELL**: Add new sentences to the KB.

**ASK**: Query the KB.

**Inference**: Derive new sentences from old.
**Generic Knowledge-Based Agent**

```latex
function KB-AGENT(\textit{percept}) returns an \textit{action}

\textbf{persistent:} \( KB \), a knowledge base
\hspace{2cm} \( t \), a counter, initially 0, indicating time

\texttt{TELL}(\textit{KB}, \texttt{MAKE-\textit{PERCEPT-SENTENCE}}(\textit{percept}, \textit{t})) // asks what action
\texttt{action} \leftarrow \texttt{ASK}(\textit{KB}, \texttt{MAKE-\textit{ACTION-QUERY}}(\textit{t})) // it should perform.
\texttt{TELL}(\textit{KB}, \texttt{MAKE-\textit{ACTION-SENTENCE}}(\textit{action}, \textit{t})) // tells what action
\texttt{t} \leftarrow \texttt{t} + 1 // was chosen.
\texttt{return \textit{action}}
```
II. The Wumpus World

Cave consisting of connected rooms.

- Some rooms contain pits that will trap whoever enters them.
- The wumpus lurks in one room ready to eat whoever enters the room.
- The wumpus can be shot by the agent, who has only one arrow.
- A heap of gold is in a different room than where the wumpus lurks.
II. The Wumpus World

Cave consisting of connected rooms.

- Some rooms contain pits that will trap whoever enters them.
- The wumpus lurks in one room ready to eat whoever enters the room.
- The wumpus can be shot by the agent, who has only one arrow.
- A heap of gold is in a different room than where the wumpus lurks.

Goal: Find the gold and bring it back to the start without getting killed.
Task Environment

Performance measure

- $+1000$ (climbing out of the cave with the gold)
- $-1000$ (falling into a pit or being eaten by the wumpus)
- $-1$ (each action taken)
- $-10$ (using up the arrow)
Task Environment

Performance measure
- +1000 (climbing out of the cave with the gold)
- −1000 (falling into a pit or being eaten by the wumpus)
- −1 (each action taken)
- −10 (using up the arrow)

Environment
- 4 × 4 grid surrounded by walls
- [1, 1]: the starting square for the agent, who faces east
- locations of the gold and the wumpus:
  - ≠ [1, 1]
  - otherwise randomly generated under uniform distribution
- 0.2 probability for a square other than [1, 1] and without gold or wumpus to be a pit
Actuators:

1) *Forward, TurnLeft by 90°, TurnRight by 90°*
   - Death of the agent if it enters a square containing a pit or a live wumpus.
   - No movement if bumping into a wall.

2) *Grab*
   - Picks up the gold if it in the same square as the agent.

3) *Shoot*
   - Fire an arrow in the direction the agent is facing.
   - The arrow continues until hitting the wumpus (who gets killed consequently) or a wall.

4) *Climb*
   - Climb out of the cave if at [1, 1].
Sensors

5 Sensors, each providing one bit of information:

1) *Stench*
   - in the squares directly (not diagonally) adjacent to the wumpus

2) *Breeze*
   - in the squares directly (not diagonally) adjacent to a pit

3) *Glitter*
   - in the square where the gold is

4) *Bump*
   - when the agent walks into a wall

5) *Scream*
   - when the wumpus is killed
Sensors

5 Sensors, each providing one bit of information:

1) *Stench*
   - in the squares directly (not diagonally) adjacent to the wumpus

2) *Breeze*
   - in the squares directly (not diagonally) adjacent to a pit

3) *Glitter*
   - in the square where the gold is

4) *Bump*
   - when the agent walks into a wall

5) *Scream*
   - when the wumpus is killed

Percepts in the form of a 5-vector:

  e.g., \([\text{Stench, Breeze, None, None, None}]\)
Characteristics of WW

- Deterministic, discrete, static, and single-agent
Characteristics of WW

- Deterministic, discrete, static, and single-agent

Outcome specified.
Characteristics of WW

- Deterministic, discrete, static, and single-agent

  Outcome specified.  The wumpus does not move.
Characteristics of WW

- Deterministic, discrete, static, and single-agent
  - Outcome specified.
  - The wumpus does not move.

- Partially observable
  - Locations of the pits and the wumpus are unknown.
Characteristics of WW

- Deterministic, discrete, static, and single-agent
  - Outcome specified.
  - The wumpus does not move.

- Partially observable
  - Locations of the pits and the wumpus are unknown.

**Challenge:** The agent needs to use logical reasoning to overcome its initial lack of knowledge about the environment’s configuration.
Solution by a Knowledge-Based Agent

\[
\begin{array}{cccc}
1,4 & 2,4 & 3,4 & 4,4 \\
1,3 & 2,3 & 3,3 & 4,3 \\
1,2 & 2,2 & 3,2 & 4,2 \\
1,1 & 2,1 & 3,1 & 4,1 \\
\end{array}
\]

\(A\) = Agent
\(B\) = Breeze
\(G\) = Glitter, Gold
\(OK\) = Safe square
\(P\) = Pit
\(S\) = Stench
\(V\) = Visited
\(W\) = Wumpus
Solution by a Knowledge-Based Agent

Percept: [None, None, None, None, None]

[Stench, Breeze, Glitter, Bump, Scream]

Percept: [None, None, None, None, None]
Solution by a Knowledge-Based Agent

[Stench, Breeze, Glitter, Bump, Scream]

Percept: [None, None, None, None, None]

\[1,2\] and [2, 1] are free of dangers.
Solution by a Knowledge-Based Agent

Percept: [None, None, None, None, None]

[1,2] and [2, 1] are free of dangers.
Solution by a Knowledge-Based Agent

Percept: [None, None, None, None, None]

[1,2] and [2,1] are free of dangers.
Solution by a Knowledge-Based Agent

Percept: \([\text{None, None, None, None, None}]\)

\([1,2]\) and \([2, 1]\) are free of dangers.

[Stench, Breeze, Glitter, Bump, Scream]
**Solution by a Knowledge-Based Agent**

Percept: `[None, None, None, None, None]`

- `1,2` and `[2, 1]` are free of dangers.

---

Percept: `[None, Breeze, None, None, None]`

- `[1, 1]` has just been visited.

---

Percept: `[Stench, Breeze, Glitter, Bump, Scream]`
Solution by a Knowledge-Based Agent

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

Percept: [None, None, None, None, None]

[1,2] and [2, 1] are free of dangers.

A pit in [1,1], [2,2], or [3, 1].

[None, Breeze, None, None, None]

A pit in [2,2] or [3, 1].

[Stench, Breeze, Glitter, Bump, Scream]
Next Step

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

Only one unexplored square [1,2] is OK.
Only one unexplored square [1,2] is OK.

Be prudent: Turn around, go back to [1,1] and move onto [1, 2].
Only one unexplored square [1,2] is OK.

Be prudent: Turn around, go back to [1,1] and move onto [1, 2].
Only one unexplored square [1,2] is OK.

Be prudent: Turn around, go back to [1,1] and move onto [1, 2].
Only one unexplored square [1,2] is OK.

Be prudent: Turn around, go back to [1,1] and move onto [1,2].
Only one unexplored square [1,2] is OK.

Be prudent: Turn around, go back to [1,1] and move onto [1, 2].
Next Step

Only one unexplored square [1,2] is OK.

Be prudent: Turn around, go back to [1,1] and move onto [1,2].

[Stench, None, None, None, None, None]

The wumpus is in [1,1], [2,2], or [1,3]. [1,1] is OK [2,2] is impossible because no stench was detected at [2,1].
Only one unexplored square [1,2] is OK.

Be prudent: Turn around, go back to [1,1] and move onto [1, 2].

The wumpus is in [1,1], [2,2], or [1, 3].

[Stench, None, None, None, None, None]

[1,1] is OK

[2,2] is impossible because no stench was detected at [2, 1].

The wumpus is in [1, 3].
Next Step

Only one unexplored square [1,2] is OK.

Be prudent: Turn around, go back to [1,1] and move onto [1, 2].

The wumpus is in [1, 1], [2, 2], or [1, 3].

- [1,1] is OK
- [2,2] is impossible because no stench was detected at [2, 1].
- The wumpus is in [1, 3].

**Stench, None, None, None, None, None**
The wumpus is in [1, 3].
The wumpus is in [1, 3].

[Stench, None, None, None, None]

No breeze in [1,2].

[2, 2] is OK.
The wumpus is in [1, 3].

No breeze in [1,2].

[2, 2] is OK.

A pit in [2,2] or [3,1].

A pit in [3,1]

Stench, None, None, None, None
The wumpus is in [1, 3].
No breeze in [1,2].

[2, 2] is OK.

A pit in [2,2] or [3,1].

A pit in [3,1]

- Moves to (2, 2).
### More Inference

The wumpus is in [1, 3].

- No breeze in [1,2].

- [2, 2] is OK.

- A pit in [2,2] or [3,1].

- A pit in [3,1]

- Moves to (2, 2).

- Assume then it turns and moves to (2, 3) based on percept at (2,2).
More Inference

<table>
<thead>
<tr>
<th></th>
<th>1,4</th>
<th>2,4</th>
<th>3,4</th>
<th>4,4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,3</td>
<td>W!</td>
<td>2,3</td>
<td>3,3</td>
<td>4,3</td>
</tr>
<tr>
<td>1,2</td>
<td>S</td>
<td>2,2</td>
<td>3,2</td>
<td>4,2</td>
</tr>
<tr>
<td>1,1</td>
<td>V</td>
<td>2,1</td>
<td>3,1</td>
<td>4,1</td>
</tr>
</tbody>
</table>

- **Moves to (2, 2).**
- **Assume then it turns and moves to (2, 3) based on percept at (2,2).**
- **Grab the gold and return home.**

**Stench, None, None, None, None**

The wumpus is in [1, 3].
No breeze in [1,2].
[2, 2] is OK.
A pit in [2,2] or [3, 1].
A pit in [3,1]

[Stench, None, None, None, None]

• Moves to (2, 2).
• Assume then it turns and moves to (2, 3) based on percept at (2,2).
• Grab the gold and return home.
The wumpus is in [1, 3].

- No breeze in [1,2].
- [2, 2] is OK.
- A pit in [2,2] or [3,1].
- A pit in [3,1]

- Moves to (2, 2).
- Assume then it turns and moves to (2, 3) based on percept at (2,2).
- Grab the gold and return home.

A conclusion drawn is guaranteed if the available information is correct.
III. Logic

- A systematic study of rules of inference.
- A formal language for representing information such that conclusions can be drawn.
III. Logic

• A systematic study of rules of inference.

• A formal language for representing information such that conclusions can be drawn.

  ♦ *Syntax* – what expressions are legal (*well-formed* sentences)
III. Logic

• A systematic study of rules of inference.

• A formal language for representing information such that conclusions can be drawn.

♦ Syntax – what expressions are legal (well-formed sentences)

“\( x + y = 4 \)” is a sentence but “\( x4y + = \)” is not.
III. Logic

- A systematic study of rules of inference.
- A formal language for representing information such that conclusions can be drawn.

- **Syntax** – what expressions are legal (*well-formed* sentences)
  
  “\( x + y = 4 \)” is a sentence but “\( x4y + = \)” is not.

- **Semantics** – what the “meanings” of sentences are.
III. Logic

- A systematic study of rules of inference.
- A formal language for representing information such that conclusions can be drawn.

- **Syntax** – what expressions are legal (well-formed sentences)
  
  “$x + y = 4$” is a sentence but “$x4y +=$” is not.

- **Semantics** – what the “meanings” of sentences are.
  
  Truth of a sentence w.r.t. each possible world (model).
III. Logic

- A systematic study of rules of inference.
- A formal language for representing information such that conclusions can be drawn.
  
  ♦ **Syntax** – what expressions are legal *(well-formed)* sentences
    
    “$x + y = 4$” is a sentence but “$x4y +=$” is not.
  
  ♦ **Semantics** – what the “meanings” of sentences are.
    
    Truth of a sentence w.r.t. each possible world *(model)*.
    
    “$x + y = 4$” is true in a world where $x$ is 2 and $y$ is 2, but false in a world where $x$ is 1 and $y$ is 1.
III. Logic

- A systematic study of rules of inference.

- A formal language for representing information such that conclusions can be drawn.

  - **Syntax** – what expressions are legal (well-formed sentences)
    
    “$x + y = 4$” is a sentence but “$x4y +$” is not.

  - **Semantics** – what the “meanings” of sentences are.

    Truth of a sentence w.r.t. each possible world (model).

    “$x + y = 4$” is true in a world where $x$ is 2 and $y$ is 2, but false in a world where $x$ is 1 and $y$ is 1.

    Every sentence must be either true or false in each possible world.
Model $m$: assigns values to variables.

$m$ satisfies a sentence $\alpha$, or $m$ is a model of $\alpha$, if $\alpha$ is true in $m$. 
Model $m$: assigns values to variables.

$m$ satisfies a sentence $\alpha$, or $m$ is a model of $\alpha$, if $\alpha$ is true in $m$.

$M(\alpha)$: set of all models of $\alpha$. 
Model and Reasoning

Model $m$: assigns values to variables.

$m$ satisfies a sentence $\alpha$, or $m$ is a model of $\alpha$, if $\alpha$ is true in $m$.

$M(\alpha)$: set of all models of $\alpha$.

Logical entailment

\[
\alpha \models \beta
\]

“The sentence $\alpha$ entails the sentence $\beta$.”
Model and Reasoning

Model $m$: assigns values to variables.

$m$ satisfies a sentence $\alpha$, or $m$ is a model of $\alpha$, if $\alpha$ is true in $m$.

$M(\alpha)$: set of all models of $\alpha$.

Logical entailment

$\alpha \models \beta$ if and only if every model of $\alpha$ is also a model of $\beta$.

“The sentence $\alpha$ entails the sentence $\beta$.”
Model and Reasoning

Model \( m \): assigns values to variables. 

\( m \) satisfies a sentence \( \alpha \), or \( m \) is a model of \( \alpha \), if \( \alpha \) is true in \( m \).

\( M(\alpha) \): set of all models of \( \alpha \).

Logical entailment

\[ \alpha \models \beta \quad \text{if and only if every model of } \alpha \text{ is also a model of } \beta. \]

“The sentence \( \alpha \) entails the sentence \( \beta \).”

Equivalently,

\[ \alpha \models \beta \quad \text{if and only if } M(\alpha) \subseteq M(\beta) \]
Model and Reasoning

Model \( m \): assigns values to variables.

\( m \) satisfies a sentence \( \alpha \), or \( m \) is a model of \( \alpha \), if \( \alpha \) is true in \( m \).

\( M(\alpha) \): set of all models of \( \alpha \).

Logical entailment

\( \alpha \models \beta \) if and only if every model of \( \alpha \) is also a model of \( \beta \).

“The sentence \( \alpha \) entails the sentence \( \beta \).”

Equivalently,

\( \alpha \models \beta \) if and only if \( M(\alpha) \subseteq M(\beta) \)

\( \alpha \) is a stronger assertion than \( \beta \).
Model and Reasoning

Model $m$: assigns values to variables.

$m$ satisfies a sentence $\alpha$, or $m$ is a model of $\alpha$, if $\alpha$ is true in $m$.

$M(\alpha)$: set of all models of $\alpha$.

Logical entailment

\[ \alpha \models \beta \quad \text{if and only if every model of } \alpha \text{ is also a model of } \beta. \]

“The sentence $\alpha$ entails the sentence $\beta$.”

Equivalently,

\[ \alpha \models \beta \quad \text{if and only if } M(\alpha) \subseteq M(\beta) \]

$\alpha$ is a stronger assertion than $\beta$.

Example $x = 0$ entails $xy = 0$. 
Back to the Wumpus World

Knowledge base (KB) includes

- All the rules.
- Percepts:
  - [None, None, None, None, None] in [1,1]
  - [None, Breeze, None, None, None] in [2,1]
Back to the Wumpus World

Knowledge base (KB) includes

- All the rules.
- Percepts:
  
  [None, None, None, None, None] in [1,1]
  [None, Breeze, None, None, None] in [2,1]

Q. Does any of the three squares [1,2], [2,2] and [3, 1] contain pits?
**Back to the Wumpus World**

Knowledge base (KB) includes

- All the rules.
- Percepts:
  
  - [None, None, None, None, None] in [1,1]
  - [None, Breeze, None, None, None] in [2,1]

Q. Does any of the three squares [1,2], [2,2] and [3, 1] contain pits?

8 possibilities if ignoring the KB.
Which Neighbors Contain a pit?

8 possible models for the presence of pits in squares [1, 2], [2, 2], and [3, 1].
Which Neighbors Contain a pit?

8 possible models for the presence of pits in squares [1, 2], [2, 2], and [3, 1].
3 models in which the KB is true given the two percepts.
Which Neighbors Contain a pit?

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

8 possible models for the presence of pits in squares [1, 2], [2, 2], and [3, 1].
3 models in which the KB is true given the two percepts.

\[ \alpha_1 = \text{"There is no pit in } [1, 2]." \]
Which Neighbors Contain a pit?

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

True in 4 models.
Which Neighbors Contain a pit?

8 possible models for the presence of pits in squares [1, 2], [2, 2], and [3, 1].
3 models in which the KB is true given the two percepts.

\[ \alpha_1 = \text{“There is no pit in [1, 2].”} \]

True in 4 models.

\[ KB \models \alpha_1 \text{ since } M(KB) \subseteq M(\alpha_1) \]
8 possible models for the presence of pits in squares [1, 2], [2, 2], and [3, 1].
3 models in which the KB is true given the two percepts.

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

True in 4 models.

\[ KB \models \alpha_1 \text{ since } M(KB) \subseteq M(\alpha_1) \]
Which Neighbors Contain a pit?

\[ \alpha_1 = \text{"There is no pit in } [1, 2] \text{."} \]
True in 4 models.

\[ \alpha_2 = \text{"There is no pit in } [2, 2] \text{."} \]
True in 4 models.

8 possible models for the presence of pits in squares \([1, 2], [2, 2], \text{and } [3, 1]\).
3 models in which the KB is true given the two percepts.

\[ KB \models \alpha_1 \text{ since } M(KB) \subseteq M(\alpha_1) \]
Which Neighbors Contain a pit?

\( \alpha_1 = \) “There is no pit in [1, 2].”

\( \alpha_2 = \) “There is no pit in [2, 2].”

True in 4 models.

\( \neg \alpha_1 \) since \( M(KB) \subseteq M(\alpha_1) \)

\( \neg \alpha_2 \) since \( M(KB) \not\subseteq M(\alpha_2) \)

8 possible models for the presence of pits in squares [1, 2], [2, 2], and [3, 1].

3 models in which the KB is true given the two percepts.

Model checking
Derivation vs. Entailment

Inference is like finding a needle entailed by (known to be in) a haystack (KB).
Derivation vs. Entailment

Inference is like finding a needle entailed by (known to be in) a haystack (KB).

\[ KB \vdash_i \alpha \]
Derivation vs. Entailment

Inference is like finding a needle entailed by (known to be in) a haystack (KB).

\[ KB \vdash_i \alpha \quad \text{“\( \alpha \) is derived from \( KB \) by (the inference algorithm) \( i \).”} \]
Derivation vs. Entailment

Inference is like finding a needle entailed by (known to be in) a haystack (KB).

\[ KB \vdash_i \alpha \]

“\( \alpha \) is derived from \( KB \) by (the inference algorithm) \( i \).”

“\( i \) derives \( \alpha \) from \( KB \).”
Derivation vs. Entailment

Inference is like finding a needle entailed by (known to be in) a haystack (KB).

\[ KB \vdash_i \alpha \]  
"\( \alpha \) is derived from KB by (the inference algorithm) \( i \)."

"\( i \) derives \( \alpha \) from KB."

An inference algorithm \( i \) is sound or truth-preserving if it derives only entailed sentences, that is.

\[ KB \models \alpha \]  whenever \[ KB \vdash_i \alpha \]
Inference is like finding a needle entailed by (known to be in) a haystack (KB).

\[ KB \vdash_i \alpha \quad \text{“} \alpha \text{ is derived from } KB \text{ by (the inference algorithm) } i. \text{”} \]

\[ i \text{ derives } \alpha \text{ from } KB. \]

• An inference algorithm \( i \) is sound or truth-preserving if it derives only entailed sentences, that is.

\[ KB \models \alpha \quad \text{whenever} \quad KB \vdash_i \alpha \]

• It is complete if it can derive any sentence that is entailed, that is,

\[ KB \vdash_i \alpha \quad \text{whenever} \quad KB \models \alpha \]
Logical Reasoning

A process whose conclusions are guaranteed to be true in any world in which the premises (the $KB$ in this case) are true.

Correspondence between world and representation
Logical Reasoning

A process whose conclusions are guaranteed to be true in any world in which the premises (the $KB$ in this case) are true.

Correspondence between world and representation
IV. Syntax of Propositional Logic

An *atomic sentence* is a single *proposition symbol*.

standing for a proposition that has to be either true or false but not both.

\[P, Q, R, W_{1,3}, \text{FacingEast}\]
IV. Syntax of Propositional Logic

An *atomic sentence* is a single *proposition symbol*. Standing for a proposition that has to be either true or false but not both.

\[ P, Q, R, W_{1,3}, \text{FacingEast} \]

The wumpus is in [1,3].
IV. Syntax of Propositional Logic

An *atomic sentence* is a single *proposition symbol*. Standing for a proposition that has to be either true or false but not both.

\[ P, Q, R, W_{1,3}, \text{FacingEast} \]

The wumpus is in [1,3].

*True*: always-true proposition

*False*: always-false proposition
A complex sentence is constructed from simpler sentences, using parentheses and logical connectives (5 in total).

- \( \neg \) (not).
  - \( \neg P \) is the *negation* of \( P \).
  - *literal*: either an atomic sentence or a negated atomic sentence.
Complex Sentences

A complex sentence is constructed from simpler sentences, using parentheses and logical connectives (5 in total).

- ¬ (not).
  - ¬P is the negation of P.
  - literal: either an atomic sentence or a negated atomic sentence.

\[ W_{1,3}, \neg Q \]
Complex Sentences

A *complex sentence* is constructed from simpler sentences, using parentheses and *logical connectives* (5 in total).

- \( \neg \) (not).
  - \( \neg P \) is the *negation* of \( P \).
  - *literal*: either an atomic sentence or a negated atomic sentence.

\[
W_{1,3}, \neg Q
\]

- \( \land \) (and).
  - \( W_{1,3} \land P_{3,1} \) is a *conjunction* whose parts \( W_{1,3} \) and \( P_{3,1} \) are *conjuncts*.

There is a pit in [3,1].
Complex Sentences

A complex sentence is constructed from simpler sentences, using parentheses and logical connectives (5 in total).

• \( \neg \) (not).
  - \( \neg P \) is the negation of \( P \).
  - literal: either an atomic sentence or a negated atomic sentence.
    \[ W_{1,3}, \neg Q \]

• \( \land \) (and).
  - \( W_{1,3} \land P_{3,1} \) is a conjunction whose parts \( W_{1,3} \) and \( P_{3,1} \) are conjuncts.
    \[ W_{1,3} \land P_{3,1} \]
    - There is a pit in [3,1].

• \( \lor \) (or).
  - \( (W_{1,3} \land P_{3,1}) \lor W_{2,2} \) is a disjunction whose parts \( (W_{1,3} \land P_{3,1}) \) and \( W_{2,2} \) are disjuncts.
More Logical Connectives

• $\Rightarrow$ (implies).

$\diamond (W_{1,3} \land P_{3,1}) \Rightarrow \neg W_{2,2}$ is an implication.
More Logical Connectives

- $\Rightarrow$ (implies).

$(W_{1,3} \land P_{3,1}) \Rightarrow \neg W_{2,2}$ is an implication.

premise or antecedent
More Logical Connectives

• \( \Rightarrow \) (implies).

\[ (W_{1,3} \land P_{3,1}) \Rightarrow \neg W_{2,2} \] is an implication.

premise or antecedent          conclusion or consequent

implies
More Logical Connectives

- \( \Rightarrow \) (implies).
  
  \[ (W_{1,3} \land P_{3,1}) \Rightarrow \neg W_{2,2} \]  
  is an implication.

- \( \Leftarrow \) (if and only if).
  
  \[ W_{1,3} \Leftarrow W_{2,2} \]  
  is a biconditional.
Grammar of Propositional Logic

Backus-Naur form (BNF):

\[
\begin{align*}
\text{Sentence} & \rightarrow \text{AtomicSentence} \mid \text{ComplexSentence} \\
\text{AtomicSentence} & \rightarrow \text{True} \mid \text{False} \mid P \mid Q \mid R \mid \ldots \\
\text{ComplexSentence} & \rightarrow ( \text{Sentence} ) \\
& \mid \neg \text{Sentence} \\
& \mid \text{Sentence} \land \text{Sentence} \\
& \mid \text{Sentence} \lor \text{Sentence} \\
& \mid \text{Sentence} \Rightarrow \text{Sentence} \\
& \mid \text{Sentence} \Leftrightarrow \text{Sentence}
\end{align*}
\]

Operator precedence:

\[
\neg, \land, \lor, \Rightarrow, \Leftrightarrow
\]

John Backus (IBM)
National Medal of Science (1975)
ACM Turing Award (1977)

Peter Naur (U. Copenhagen)
ACM Turing Award (2005)

* Photos from https://amturing.acm.org/bbyear.cfm.
Grammar of Propositional Logic

Backus-Naur form (BNF):

\[
\begin{align*}
\text{Sentence} & \rightarrow \text{AtomicSentence} \mid \text{ComplexSentence} \\
\text{AtomicSentence} & \rightarrow \text{True} \mid \text{False} \mid P \mid Q \mid R \mid \ldots \\
\text{ComplexSentence} & \rightarrow (\text{Sentence}) \\
& \mid \neg \text{Sentence} \\
& \mid \text{Sentence} \land \text{Sentence} \\
& \mid \text{Sentence} \lor \text{Sentence} \\
& \mid \text{Sentence} \Rightarrow \text{Sentence} \\
& \mid \text{Sentence} \Leftrightarrow \text{Sentence}
\end{align*}
\]

**Operator Precedence**:
\[\neg, \land, \lor, \Rightarrow, \Leftrightarrow\]

\[\neg A \lor B \land C \Rightarrow D \text{ is equivalent to } ((\neg A) \lor (B \land C)) \Rightarrow D\]