Propositional Logic

CPSC 470 – Artificial Intelligence
Brian Scassellati
Constraint Satisfaction Problems

T W O
+ T W O
-------
F O U R

2
1

4
3

2
1

1
2
3
4

4
3
2
1

Western Australia
Northern Territory
Queensland
South Australia
New South Wales
Victoria
Tasmania
# World Characterization

<table>
<thead>
<tr>
<th></th>
<th>Search</th>
<th>CSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully Observable</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Episodic</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Static</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Discrete</td>
<td>Yes</td>
<td>Mostly</td>
</tr>
<tr>
<td></td>
<td>Search</td>
<td>CSP</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>-----</td>
</tr>
<tr>
<td>Fully Observable</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Episodic</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Static</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Discrete</td>
<td>Yes</td>
<td>Mostly</td>
</tr>
</tbody>
</table>
The Wumpus World

- Grid-like world
- Noble hero
- Horrible wumpus
- Bottomless pits
- Gold
- Breeze
- Stench

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="stench" /></td>
<td><img src="image2" alt="gold" /></td>
<td><img src="image3" alt="breeze" /></td>
</tr>
<tr>
<td><img src="image4" alt="stench" /></td>
<td><img src="image5" alt="breeze" /></td>
<td><img src="image6" alt="pit" /></td>
</tr>
<tr>
<td><img src="image7" alt="stench" /></td>
<td><img src="image8" alt="breeze" /></td>
<td><img src="image9" alt="pit" /></td>
</tr>
<tr>
<td><img src="image10" alt="hero" /></td>
<td><img src="image11" alt="breeze" /></td>
<td><img src="image12" alt="pit" /></td>
</tr>
</tbody>
</table>
## Actions in the Wumpus World

<table>
<thead>
<tr>
<th>N</th>
<th>S</th>
<th>E</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Goals:
- find the gold
- kill the wumpus
- go home

### Actions
- Move N,S,E,W
- Grab
- Shoot(N,S,E,W)
  - Only one arrow!
The Wumpus World

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>stench</strong></td>
<td></td>
<td><strong>breeze</strong></td>
</tr>
<tr>
<td><strong>stench</strong></td>
<td></td>
<td><strong>breeze</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>stench</strong></td>
<td></td>
<td><strong>breeze</strong></td>
</tr>
<tr>
<td><strong>stench</strong></td>
<td><strong>breeze</strong></td>
<td><strong>breeze</strong></td>
</tr>
</tbody>
</table>

- If we had complete knowledge of the world, then we could simply build a search tree.
- What if our perceptions are limited?
Incomplete Knowledge of the World

- Agent’s percepts:
  - Stench
  - Breeze
  - Glitter
  - Bump
  - Scream

- Other than the agent, the world is static
# Our First Wumpus Hunt

<table>
<thead>
<tr>
<th></th>
<th>scent</th>
<th>breeze</th>
<th>glitter</th>
<th>bump</th>
<th>scream</th>
</tr>
</thead>
<tbody>
<tr>
<td>East</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>West</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>North</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>East</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>North</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Shoot(W)</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>West</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>North</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>East</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Grab</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Today we will see how to build an agent that can perform this reasoning
Representing Beliefs

- In most programming languages, it is easy to specify statements like this…
  - There is a pit in square [3,1]
- But it is difficult to specify statements like these…
  - There is a pit in either square [3,1] or [2,2]
  - There is no wumpus in square [2,2]
  - Because there was no breeze in square [1,2], there is a pit in square [3,1]
- Require an agent that can represent this knowledge and perform the reasoning to infer new conclusions
Components of a Logic

• A formal system for representing the state of affairs
  – A sentence is a representation of a fact about the world
  – A syntax that describes how to make sentences
  – A semantics that gives constraints on how sentences relate to the state of affairs
  – A proof theory – a set of rules for deducing the entailments of a set of sentences

Entailment means that one thing follows from another
Properties of Logical Inference

• Inference is **complete** if it can find a proof for any sentence that is entailed.

• A sentence is **valid** or necessarily true if and only if it is true under all possible interpretations in all possible worlds.

  *There is a stench in [1,1] or there is not a stench in [1,1]*

• A sentence is **satisfiable** if and only if there is some interpretation in some world for which it is true.

  *There is a wumpus at [1,1]*
Types of Commitment

<table>
<thead>
<tr>
<th>Language</th>
<th>Ontological Commitment (What exists in the world)</th>
<th>Epistemological Commitment (What an agent believes about facts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propositional logic</td>
<td>facts</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>First-order logic</td>
<td>facts, objects, relations</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>Temporal logic</td>
<td>facts, objects, relations, times</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>Probability theory</td>
<td>facts</td>
<td>degree of belief 0…1</td>
</tr>
<tr>
<td>Fuzzy logic</td>
<td>degree of truth</td>
<td>degree of belief 0…1</td>
</tr>
</tbody>
</table>

- We make assumptions about
  - the world (**ontological commitments**)  
  - the beliefs that an agent can hold (**epistemological commitments**)
Propositional Logic Syntax

• Basic Units (sentences)
  – *True* and *False*
  – Propositions $P$, $Q$, …

• Connectives
  
  $P \land Q$ and (conjunction)
  
  Returns true if both $P$ and $Q$ are true

  $P \lor Q$ or (disjunction)
  
  Returns true if either $P$ or $Q$ is true

  $P \Rightarrow Q$ implication
  
  If $P$ is true then $Q$ is also true

  $P \Leftrightarrow Q$ equivalence
  
  $P$ is true exactly when $Q$ is true

  $\neg P$ negation
  
  Returns true when $P$ is false
Propositional Logic Grammar

• BNF (Backus-Naur form) for PL Grammar:

\[
\text{Sentence} \rightarrow \text{AtomicSentence} | \text{ComplexSentence} \\
\text{AtomicSentence} \rightarrow \text{True} | \text{False} | P | Q | ... \\
\text{ComplexSentence} \rightarrow (\text{Sentence}) | \\
\quad \text{Sentence Connective Sentence} | \\
\quad \neg \text{Sentence} \\
\text{Connective} \rightarrow \land | \lor | \Rightarrow | \Leftrightarrow \\
\]

• Also require an order of precedence

From highest to lowest: \( \neg \land \lor \Rightarrow \Leftrightarrow \)
Propositional Logic Semantics

• Propositions can have any semantic meaning:
  \( P = \) “Paris is the capital of France”
  \( Q = \) “The wumpus is dead”
  \( R = \) “Bill Gates is the US President”

• Compound functions can be derived from a truth table:

<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 \iff 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>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>
Validity and Inference

\[ ((P \lor H) \land \neg H) \Rightarrow P \]

- Truth tables can also be used to test validity of a sentence
- Remember to read implications as conditionals: \( P \Rightarrow Q \) is read as "if P then Q"
Inference Rules for Propositional Logic

• Modus Ponens (Implication-Elimination)
  – From an implication and its premise, infer conclusion
    \[ \alpha \Rightarrow \beta , \alpha \]
    \[ \frac{\beta}{\beta} \]

• And-Elimination
  – From a conjunction, you can infer any conjunct
    \[ \alpha_1 \land \alpha_2 \land \alpha_3 \land \ldots \land \alpha_n \]
    \[ \frac{\alpha_i}{\alpha_i} \]
Inference Rules for Propositional Logic

• And-Introduction
  – From a list of sentences, you can infer the conjunct
    \[
    \alpha_1, \alpha_2, \alpha_3, \ldots \alpha_n \quad \frac{\alpha_1, \alpha_2, \alpha_3, \ldots \alpha_n}{\alpha_1 \land \alpha_2 \land \ldots \land \alpha_n}
    \]

• Or-Introduction
  – From a sentence, infer its disjunction with anything
    \[
    \alpha_i \quad \frac{\alpha_i}{\alpha_1 \lor \alpha_2 \lor \ldots \lor \alpha_n}
    \]
Inference Rules for Propositional Logic

• Double-Negative Elimination
  – From a double negation, infer the positive sentence

\[ \neg
\neg \alpha \quad \Rightarrow \quad \alpha \]

• Unit Resolution
  – From a disjunction in which one is false, then you can infer the other is true

\[ \alpha \vee \beta \land \neg \beta \quad \Rightarrow \quad \alpha \]
Inference Rules for Propositional Logic

• Resolution
  – Since beta cannot be both true and false, one of the disjuncts must be true

\[
\frac{\alpha \lor \beta, \lnot \beta \lor \gamma}{\alpha \lor \gamma}
\]

– Implication is transitive

\[
\frac{\lnot \alpha \Rightarrow \beta, \beta \Rightarrow \gamma}{\lnot \alpha \Rightarrow \gamma}
\]
Truth Table for Resolution

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\gamma$</th>
<th>$\alpha \lor \beta$</th>
<th>$\neg \beta \lor \gamma$</th>
<th>$\alpha \lor \gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

- Truth tables can also be used to verify the inference rules

\[
\frac{\alpha \lor \beta, \neg \beta \lor \gamma}{\alpha \lor \gamma}
\]
Logical Agents

- Input sentences can come from the user perceiving the world, or from a machine-readable representation of the world.
- Infer new statements about the world that are valid.
An Agent for the Wumpus World

- Convert perceptions into sentences:
  “In square [1,1], there is no breeze and no stench” … becomes…
  \( \neg B_{11} \land \neg S_{11} \)

- Start with some knowledge of the world (in the form of rules)
  
  \( R1 : \neg S_{11} \Rightarrow \neg W_{11} \land \neg W_{12} \land \neg W_{21} \)
  
  \( R2 : \neg S_{21} \Rightarrow \neg W_{11} \land \neg W_{21} \land \neg W_{22} \)

  ....

  \( R4 : S_{12} \Rightarrow W_{13} \lor W_{12} \lor W_{22} \lor W_{11} \)
Finding the Wumpus

1. Apply modus ponens and and-elimination to $\neg S_{11} \Rightarrow \neg W_{11} \land \neg W_{12} \land \neg W_{21}$ to get
   $\neg W_{11} \land \neg W_{12} \land \neg W_{21}$

2. Apply modus ponens and and-elimination to $\neg S_{21} \Rightarrow \neg W_{11} \land \neg W_{21} \land \neg W_{22}$ to get
   $\neg W_{22} \land \neg W_{21} \land \neg W_{31}$

3. Apply modus ponens to $S_{12} \Rightarrow W_{13} \lor W_{12} \lor W_{22} \lor W_{11}$ to get
   $W_{13} \lor W_{12} \lor W_{22} \lor W_{11}$

4. Apply unit resolution to #3 and #1
   $W_{13} \lor W_{22}$

5. Apply unit resolution to #4 and #2
   $W_{13}$

The wumpus is in square [1,3]!!!
Problems with Propositional Logic

• Too many propositions!
  – How can you encode a rule such as “don’t go forward if the wumpus is in front of you”?
  – In propositional logic, this takes (16 squares * 4 orientations) = 64 rules!

• Truth tables become unwieldy quickly
  – Size of the truth table is $2^n$ where $n$ is the number of propositional symbols
More Problems with Propositional Logic

• No good way to represent changes in the world
  – How do you encode the location of the agent?

• What kinds of practical applications is this good for?
  – Relatively little
Coming Up...

• More powerful logic!
  – First-order logic (also known as First Order Predicate Calculus)