Title: Logical Agents
AIMA: Chapter 7 (Sections 7.1, 7.2, and 7.3)

Introduction to Artificial Intelligence
CSCE 476-876, Fall 2019
URL: cse.unl.edu/~choueiry/F19-476-876

Berthe Y. Choueiry (Shu-we-ri)
(402)472-5444
Outline

• Wumpus world: motivating example

• Knowledge bases

• Logic for Knowledge Representation & Reasoning
  – Syntax
  – Semantics
  – Inference mechanisms: complexity, completeness

Propositional logic/sentential logic
Predicate logic/first-order logic
Motivating example: The Wumpus world

Early computer game

Agent explores a cave with:

- bottomless pits
- a beast that eats anyone who enters the room, and
- heap of gold to trap
PEAS description of the Wumpus world

Performance measure: gold +1000, death -1000, -1 per step, -10 for using the arrow

Environment: Squares adjacent to Wumpus are smelly
Squares adjacent to pit are breezy
Glitter iff gold is in the same square
Shooting kills Wumpus if you are facing it
Shooting uses up the only arrow
Grabbing picks up gold if in same square
Releasing drops the gold in same square

Sensors: Breeze, Glitter, Smell

Actuators: Left turn, Right turn, Forward, Grab, Release, Shoot
Wumpus World: Characterization

Is the world:

- Observable? No, only local perception
- Deterministic? Yes, outcome exactly specified
- Episodic? No, sequential at the level of actions
- Static? Yes, Wumpus/Pits don’t move
- Discrete? Yes
- Single-agent? Yes, Wumpus considered a natural feature
Empirical evaluations: single/multiple configuration

An agent can do well in a single environment: learns the environment, executes rules.

Agent must be tested in a complete class of environments and its average performance must be determined → empirical experiments

- Constraints: start from position [1,1], limited to 4×4 grid
- Location of Wumpus and Gold chosen randomly with a uniform distribution (all squares are possible except [1,1])
- Each square, except [1,1], can be a pit with probability 0.2
- Terribly bad cases: gold in a pit or surrounded by pits
Wumpus World: Acting & Reasoning

- After receiving initial percepts, agent knows it is in [1,1] and it is OK
- No stench or breeze in [1,1] ⇒ [1,2] and [2,1] are danger-free
- Cautious agent moves only to square it knows it is OK
- Agent moves only to square [2,1], detects breeze ⇒ ∃ a pit in neighboring squares [1,1], [2,2] and [3,1]. Agent knows no pit in [1,1] → Pit indicated in [2,2] and [3,1] with P?
- Not visited OK squares? Only [1,2]. Agent goes to [1,1], proceeds to [1,2]
Wumpus World: Acting & Reasoning

1,4  2,4  3,4  4,4

1,3  2,3  3,3  4,3

1,2  2,2  3,2  4,2

A  B  OK

1,1  A  OK  OK

(a)

1,4  2,4  3,4  4,4

1,3  2,3  3,3  4,3

1,2  2,2  P?  4,2

1,1  V  OK

(b)
Wumpus World: Acting & Reasoning

- Agents detects stench in $[1,2]$ ⇒ Wumpus nearby!
  Possibilities: $[1,1]$, $[1,3]$ or $[2,2]$.
  Agent knows $[1,1]$ is Wumpus-free (Agent was there!)
  Agent can infer $[2,2]$ is Wumpus-free ($\not\exists$ stench in $[2,1]$)
  Agent infers Wumpus is in $[1,3]$ ($\overline{W}$?)

- Lack of breeze in $[1,2]$ ⇒ $[2,2]$ is pit-free
  But, $\exists$ a pit in either $[2,2]$ or $[3,1]$ ⇒ $\exists$ pit in $[3,1]$ ($P!$)
  Inference combines knowledge gained at different times and places, beyond the abilities of most animals, but Logical Inference can handle this

- Since $[2,2]$ is OK and not visited, Agent moves there

- etc.
### Wumpus World: Acting & Reasoning

- **Agent** = A
- **Breeze** = B
- **Glitter, Gold** = G
- **Safe square** = OK
- **Pit** = P
- **Stench** = S
- **Visited** = V
- **Wumpus** = W

#### (a)

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

#### (b)

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

---

- Starting point (v, h) = (1, 1)
- Gold on (4, 1)
- Agent enters (1, 1)
- Action: Move right
- Results: Movement right, Wumpus!
- Location of agent (1, 2)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 1)
- Action: Investigate
- Results: S, G
- Location of agent (1, 1)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
- Location of agent (1, 2)
- Action: Move right
- Results: Movement right, OK
- Location of agent (1, 3)
- Action: Investigate
- Results: S, G
- Location of agent (1, 3)
- Action: Move left
- Results: Movement left, OK
- Location of agent (1, 2)
- Action: Investigate
- Results: Wumpus!
The point of the Wumpus world

In each case where the agent draws a conclusion from the available information, that conclusions is guaranteed to be correct if the available information is correct.

→ Fundamental property of logical reasoning.
Knowledge Base

A fact in the world: A representation of a fact in the world
    A sentence= a representation of a fact in the world in a
    formal language

A Knowledge Based (KB): A set sentences
    A set (of representations) of facts about the world

Issues: Access to KB, Representation (language), Reasoning
        (inference)
Level of Knowledge

Agents can be viewed at various levels:

1. **Epistemological**:  
   Abstract description of what the agent knows about the world

2. **Logical**:  
   Encoding of knowledge into sentences

3. **Implementation**:  
   Actual implementation (lists, arrays, hash tables, etc.)
   - Very important for performance of agent
   - Irrelevant for higher levels of knowledge
A simple KB-agent

function KB-AGENT(percept) returns an action

static: $KB$, a knowledge base
$t$, a counter, initially 0, indicating time

$TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))$

$action \leftarrow ASK(KB, MAKE-ACTION-QUERY(t))$

$TELL(KB, MAKE-ACTION-SENTENCE(action, t))$

$t \leftarrow t + 1$

return $action$

The agent must be able to:

represent states, actions, etc.
incorporate new percepts
update internal representations of the world
deduce hidden properties of the world
deduce appropriate actions
Knowledge-Based Agent

function KB-AGENT(\textit{percept}) \textbf{returns} an \textit{action} \\
\textbf{static}: \textit{KB}, a knowledge base \\
\hspace{1cm} \textit{t}, a counter, initially 0, indicating time \\

\textsc{Tell}(\textit{KB, Make-Percept-Sentence}(\textit{percept, t})) \\
\textit{action} \leftarrow \textsc{Ask}(\textit{KB, Make-Action-Query}(\textit{t})) \\
\textsc{Tell}(\textit{KB, Make-Action-Sentence}(\textit{action, t})) \\
\textit{t} \leftarrow \textit{t} + 1 \\
\textbf{return} \textit{action}

\textbf{Perceives:} Tells \textit{KB} about new percepts (new sentences) \\
\textbf{Representation:} \textsc{Make-Percept-Sentence}

\textbf{Access to \textit{KB}:} Asks \textit{KB} about actions to take (inference) \\
Two primitives: \textsc{Ask} and \textsc{Tell} hide reasoning details

\textbf{Acts:} Tells \textit{KB} about actions (new sentences) \\
\textbf{Representation:} \textsc{Make-Action-Sentence}, \\
\textsc{Make-Action-Query}
Logic in general

**Logics** are formal languages for representing information such that conclusions can be drawn

**Syntax** defines the sentences in the language (grammar)

**Semantics** define the “meaning” of sentences; *i.e.*, define truth of a sentence in a world

Example: the language of arithmetic

- **Syntax:** $x + 2 \geq y$ is a sentence; $x^2 + y >$ is not a sentence

- **Semantics:**
  - $x + 2 \geq y$ is true iff the number $x + 2$ is no less than the number $y$
  - $x + 2 \geq y$ is true in a world where $x = 7, y = 1$
  - $x + 2 \geq y$ is false in a world where $x = 0, y = 6$
Types of logic

Logics are characterized by what they commit to as “primitives”

**Ontological commitment:**
what exists—facts? objects? time? beliefs?

**Epistemological commitment:**
what states of knowledge?

<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>
Knowledge representation & reasoning

Facts: in the world

Representations: in the computer

Reasoning: process of constructing new representations from old ones

Proper Reasoning: ensures new representations correspond to facts that actually follow from facts in the world
Entailment

Entailment means that one thing follows from another:

\[(\text{KB} \models \alpha)\]

Knowledge base KB \textbf{entails} sentence \(\alpha\)

iff \(\alpha\) is true in all worlds where KB is true

Example: KB: \{\(a \land b\)\}, then

\[\text{KB} \models a; \text{KB} \models b; \text{KB} \models a \lor b\]

Entailment is a relationship between sentences (i.e., syntax) that is based on semantics

\[(\alpha \models \beta): \text{the truth of } \beta \text{ contains the truth of } \alpha\]

For example: \((x + y = 4) \models (4 = x + y), (x + y = 4) \models (4 \geq x + y), (x + y \geq 4) \not\models (4 = x + y),\]
Models

Logicians typically think in terms of models, which are formally structured worlds with respect to which truth can be evaluated.

We say \( m \) is a model of a sentence \( \alpha \) if \( \alpha \) is true in \( m \).

\( M(\alpha) \) is the set of all models of \( \alpha \).

Then \( KB \models \alpha \) if and only if \( M(KB) \subseteq M(\alpha) \).
**Entailment** in the Wumpus world

Situation: Agent detected nothing in [1,1], breeze in [2,1]

\[2^3 = 8\] possible models

Percepts + the PEAS description = KB

Agent wonders whether pit is in [1,2], [2,2], and [3,1]:

Only 3 models where the KB is true

\[\alpha_1 = \text{no pit in [1,2]}:\]

\[\alpha_1\] is true in 4 models.
Entailment in the Wumpus world

Consider: $\alpha_1 = \text{no pit in [1,2]}$, $\alpha_2 = \text{no pit in [2,2]}$

Model checking: $\text{KB} \models \alpha_1$, $\text{KB} \not\models \alpha_2$

Given KB, agent cannot conclude whether $\alpha_2$ holds or not

Entailment can be used to derive conclusions: Inference

Inference here is done by model checking
Inference

KB ⊢_i α ≡ α is derived from KB by procedure i

Consequences of KB are a haystack; α is a needle.
Entailment = needle in haystack; inference = finding it

**Soundness:** i is sound if
whenever KB ⊢_i α, it is also true that KB |= α

**Completeness:** i is complete if
whenever KB |= α, it is also true that KB ⊢_i α

That is, the procedure will answer any question whose answer follows from what is known by the KB

The record of operation of a sound inference procedure is a **proof**

Next, propositional logic: syntax, semantics, and inference