Inf2D 06: Logical Agents: Knowledge Bases and the Wumpus World

Cristina Alexandru

School of Informatics, University of Edinburgh

26/01/18

Slide Credits: Jacques Fleuriot, Michael Rovatsos, Michael Herrmann
Knowledge-based agents
Wumpus world
Logic in general – models and entailment
Propositional (Boolean) logic
Equivalence, validity, satisfiability
## Knowledge bases

<table>
<thead>
<tr>
<th>Inference engine</th>
<th>← domain-independent algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge base</td>
<td>← domain-specific content</td>
</tr>
</tbody>
</table>

- **Knowledge base (KB):** Set of *sentences* in a formal language
- **Declarative** approach to building a KB:
  - Tell it what it needs to know
  - Then the agent can Ask the KB what to do
    - answers should follow from the KB
- KB can be part of agent or be accessible to many agents
- The agent’s KB can be viewed at the **knowledge level**
  - i.e., what it knows, regardless of how implemented
- Or at the **implementation level**
  - i.e., data structures in KB and algorithms that manipulate them
A simple knowledge-based agent

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

```
function KB-AGENT(percept) returns an action

persistent

  KB, a knowledge base
  t, a counter, initially 0, indicating time

  TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
  action ← ASK(KB, MAKE-ACTION-QUERY(t))
  TELL(KB, MAKE-ACTION-SENTENCE(action, t))
  t ← t + 1

return action
```
Wumpus World PEAS description

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

- **Environment**
  - Squares adjacent to Wumpus are smelly
  - Squares adjacent to pits 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

- **Actuators:** Left turn, Right turn, Forward, Grab, Release, Shoot

- **Sensors:** Stench, Breeze, Glitter, Bump, Scream
Wumpus world characterization

- **Fully Observable?** No – only local perception
- **Deterministic?** Yes – outcomes exactly specified
- **Episodic?** No – sequential at the level of actions
- **Static?** Yes – Wumpus and Pits do not move
- **Discrete?** Yes
- **Single-agent?** Yes – Wumpus is essentially a natural feature
Exploring a Wumpus world
Exploring a Wumpus world

A B

OK OK

A

OK

A
Exploring a Wumpus world
Exploring a Wumpus world

Diagram of a Wumpus world:

- A: Agent
- B: Bumpus
- S: Stench
- P: Pit
- OK: Safe

The agent is exploring the environment, avoiding dangers like the Wumpus and pits.
Exploring a Wumpus world

![Diagram of a Wumpus world grid with symbols for P, B, A, S, and W]
Exploring a Wumpus world

[Diagram of a Wumpus world grid]

- P: Pit
- B: Breeze
- A: Air
- S: Smell
- W: Wumpus

Cristina Alexandru

Inf2D 06: Logical Agents: Knowledge Bases and the Wumpus World
Exploring a Wumpus world

Cristina Alexandru
Exploring a Wumpus world
Logic in general

- **Logics** are formal languages for representing information such that conclusions can be drawn.
- **Syntax** defines the sentences in the language.
- **Semantics** defines the “meaning” of sentences, i.e., define truth of a sentence in a world.

E.g., the language of arithmetic:

- $x + 2 \geq y$ is a sentence; $x2 + y >$ is not a sentence.
- $x + 2 \geq y$ 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 $x = 0$, $y = 6$. 

Cristina Alexandru

Inf2D 06: Logical Agents: Knowledge Bases and the Wumpus World
Entailment means that one thing follows from another:

\[ KB \models \alpha \]

Knowledge base KB entails sentence \( \alpha \) if and only if \( \alpha \) is true in all worlds where KB is true.

- e.g., the KB containing “Celtic won” and “Hearts won” entails “Celtic won or Hearts won”
- Considering only worlds where Celtic plays Hearts (and no draws) it entails “Either Celtic won or Hearts won”
- e.g., \( x + y = 4 \) entails \( 4 = x + y \)
- Entailment is a relationship between sentences (i.e., syntax) that is based on semantics
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

\[ KB \models \alpha \text{ iff } M(KB) \subseteq M(\alpha) \]

The stronger an assertion, the fewer models it has.
Entailment in the Wumpus world

Situation after detecting nothing in [1,1], moving right, breeze in [2,1]

Consider possible models for KB assuming only pits

3 Boolean choices $\Rightarrow$ 8 possible models

Mid-lecture Exercise: What are these 8 models?
Wumpus models
$KB = \text{Wumpus-world rules} + \text{observations}$
Wumpus models

- $KB = \text{Wumpus-world rules} + \text{observations}$
- $\alpha_1 = \text{"[1,2] has no pit"}$, $KB \models \alpha_1$, proved by model checking
  - In every model in which $KB$ is true, $\alpha_1$ is also true.
Wumpus models

\[ KB = \text{Wumpus-world rules} + \text{observations} \]
\( KB = \) Wumpus-world rules + observations

\( \alpha_2 = \) "[2,2] has no pit", \( KB \not\models \alpha_2 \)

- In some models in which \( KB \) is true, \( \alpha_2 \) is also true
Inference

- $KB \vdash_i \alpha$: sentence $\alpha$ can be derived from $KB$ by an inference procedure $i$
- **Soundness**: $i$ is sound if whenever $KB \vdash_i \alpha$, it is also true that $KB \models \alpha$
- **Completeness**: $i$ is complete if whenever $KB \models \alpha$, it is also true that $KB \vdash_i \alpha$
- Preview: we will define first-order logic:
  - expressive enough to say almost anything of interest,
  - sound and complete inference procedure exists.
  - But first...
Propositional logic is the simplest logic – illustrates basic ideas:

- The proposition symbols $P_1$, $P_2$ etc. are sentences.
- If $S$ is a sentence, $\neg S$ is a sentence (negation).
- If $S_1$ and $S_2$ are sentences, $S_1 \land S_2$ is a sentence (conjunction).
- If $S_1$ and $S_2$ are sentences, $S_1 \lor S_2$ is a sentence (disjunction).
- If $S_1$ and $S_2$ are sentences, $S_1 \Rightarrow S_2$ is a sentence (implication).
- If $S_1$ and $S_2$ are sentences, $S_1 \Leftrightarrow S_2$ is a sentence (biconditional).
Propositional logic: Semantics

- Each model specifies true/false for each proposition symbol, e.g. $P_{1,2}$, $P_{2,2}$, $P_{3,1}$, false, true, false

With these symbols, 8 possible models, can be enumerated automatically.

- Rules for evaluating truth with respect to a model $m$:
  \begin{align*}
  \neg S & \quad \text{is true iff } S \text{ is false} \\
  S_1 \land S_2 & \quad \text{is true iff } S_1 \text{ is true and } S_2 \text{ is true} \\
  S_1 \lor S_2 & \quad \text{is true iff } S_1 \text{ is true or } S_2 \text{ is true} \\
  S_1 \implies S_2 & \quad \text{is true iff } S_1 \text{ is false or } S_2 \text{ is true} \\
  \text{i.e.} & \quad \text{is false iff } S_1 \text{ is true and } S_2 \text{ is false} \\
  S_1 \iff S_2 & \quad \text{is true iff } S_1 \implies S_2 \text{ is true and } S_2 \implies S_1 \text{ is true}
  \end{align*}

- Simple recursive process evaluates an arbitrary sentence, e.g.,
  \[
  \neg P_{1,2} \land (P_{2,2} \lor P_{3,1}) = \text{true} \land (\text{true} \lor \text{false}) = \text{true} \land \text{true} = \text{true}
  \]
### Truth tables for connectives

<table>
<thead>
<tr>
<th>$P$</th>
<th>$Q$</th>
<th>$\neg P$</th>
<th>$P \wedge Q$</th>
<th>$P \vee Q$</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>false</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>
Let $P_{i,j}$ be true if there is a pit in $[i,j]$.
Let $B_{i,j}$ be true if there is a breeze in $[i,j]$.

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

"Pits cause breezes in adjacent squares"

$B_{1,1} \iff (P_{1,2} \lor P_{2,1})$
$B_{2,1} \iff (P_{1,1} \lor P_{2,2} \lor P_{3,1})$

Recall: $\alpha_1 = "[1,2] \text{ has no pit}"$, 
Truth tables for inference

<table>
<thead>
<tr>
<th>$B_{1,1}$</th>
<th>$B_{2,1}$</th>
<th>$P_{1,1}$</th>
<th>$P_{1,2}$</th>
<th>$P_{2,1}$</th>
<th>$P_{2,2}$</th>
<th>$P_{3,1}$</th>
<th>$KB$</th>
<th>$\alpha_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>
**Inference by enumeration**

- Depth-first enumeration of all models is sound and complete

```plaintext
function TT-ENTAILS?(KB, α) returns true or false
    symbols ← a list of the proposition symbols in KB and α
    return TT-CHECK-ALL(KB, α, symbols, [])

function TT-CHECK-ALL(KB, α, symbols, model) returns true or false
    if EMPTY?(symbols) then
        if PL-TRUE?(KB, model) then return PL-TRUE?(α, model)
        else return true
    else do
        P ← FIRST(symbols); rest ← REST(symbols)
        return TT-CHECK-ALL(KB, α, rest, EXTEND(P, true, model)) and
        TT-CHECK-ALL(KB, α, rest, EXTEND(P, false, model))
```

- PL-TRUE? returns true if a sentence holds within a model
- EXTEND(P, val, model) returns a new partial model in which P has value val
- For n symbols, time complexity: $O(2^n)$, space complexity: $O(n)$
Logical equivalence

- Two sentences are logically equivalent iff true in the same models: \( \alpha \equiv \beta \) iff \( \alpha \models \beta \) and \( \beta \models \alpha \)

\[
\begin{align*}
(\alpha \land \beta) & \equiv (\beta \land \alpha) \quad \text{commutativity of } \land \\
(\alpha \lor \beta) & \equiv (\beta \lor \alpha) \quad \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) & \equiv (\alpha \land (\beta \land \gamma)) \quad \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) & \equiv (\alpha \lor (\beta \lor \gamma)) \quad \text{associativity of } \lor \\
\neg(\neg \alpha) & \equiv \alpha \quad \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) & \equiv (\neg \beta \Rightarrow \neg \alpha) \quad \text{contraposition} \\
(\alpha \Rightarrow \beta) & \equiv (\neg \alpha \lor \beta) \quad \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) & \equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination} \\
\neg(\alpha \land \beta) & \equiv (\neg \alpha \lor \neg \beta) \quad \text{de Morgan} \\
\neg(\alpha \lor \beta) & \equiv (\neg \alpha \land \neg \beta) \quad \text{de Morgan} \\
(\alpha \land (\beta \lor \gamma)) & \equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) \quad \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) & \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) \quad \text{distributivity of } \lor \text{ over } \land
\end{align*}
\]
Validity and satisfiability

- A sentence is **valid** if it is true in all models,
  
  e.g. true, $A \lor \neg A$, $A \Rightarrow A$, $(A \land (A \Rightarrow B)) \Rightarrow B$

- Validity is connected to inference via the **Deduction Theorem**:

  $$KB \vDash \alpha \text{ if and only if } (KB \Rightarrow \alpha) \text{ is valid}$$

- A sentence is **satisfiable** if it is true in some model

  e.g., $A \lor B$, $C$

- A sentence is **unsatisfiable** if it is true in no models

  e.g., $A \land \neg A$

- Satisfiability is connected to inference via the following:

  $$KB \vDash \alpha \text{ if and only if } (KB \land \neg \alpha) \text{ is unsatisfiable}$$
Proof methods divide into (roughly) two kinds:

- **Application of inference rules**
  - Legitimate (sound) generation of new sentences from old
  - **Proof** = a sequence of inference rule applications.
  - Can use inference rules as operators in a standard search algorithm
  - Typically require transformation of sentences into a normal form
  - Example: resolution

- **Model checking**
  - truth table enumeration (always exponential in \( n \))
  - improved backtracking, e.g., Davis-Putnam-Logemann-Loveland (DPLL) method
  - heuristic search in model space (sound but incomplete)
    - e.g., min-conflicts-like hill-climbing algorithms
Logical agents apply inference to a knowledge base to derive new information and make decisions.

Basic concepts of logic:

- **syntax**: formal structure of sentences
- **semantics**: truth of sentences w.r.t. models
- **entailment**: necessary truth of one sentence given another
- **inference**: deriving sentences from other sentences
- **soundness**: derivations produce only entailed sentences
- **completeness**: derivations can produce all entailed sentences

Wumpus world requires the ability to represent partial and negated information, reason by cases, etc.

Does propositional logic provide enough expressive power for statements about the real world?