Today

• Previously:
  – Search, search and more search
  – Game playing
  – Constraint satisfaction problems

• Today: Knowledge-based agents
  – Representing knowledge
  – Logic in general
  – Propositional (Boolean) logic
  – First-order logic
Playing board games

How would you represent states in these games?

Playing more complex games

“Clash of Clans is an online multiplayer game in which players build a community, train troops, and attack other players to earn gold and elixir, and Dark Elixir, which can be used to build defenses that protect the player from other players’ attacks, and to train and upgrade troops.”

How would you represent states in this game?

Need a more general notion of knowledge:
• How to represent it.
• How to reason about it.
Knowledge representation

• An intelligent agent needs to be able to perform many tasks:
  – Perception: what is my state?
  – Cognition: what action should I take?

• State recognition implies some form of representation.
• Choosing the right action implies some form of inference.

Two levels to think about:
  – Implementation level: how is knowledge represented?
  – Knowledge level: what does the agent know?

Knowledge bases

The long-term dream of AI. A declarative approach to building agents.

Assumes agents have two different parts:
  – The knowledge base, which contains a set of facts expressed in some formal, standard language.
  – The inference engine, with general rules for deducing new facts and drawing conclusions.
An example: Wumpus World

Percepts: Breeze, Glitter, Stench, (not the full board.)
Actions: Turn-left, Turn-right, Forward, Grab, Shoot
Goal: Get Gold back to Start

without entering Pit or Wumpus square

Rules of the environment:

• Squares adjacent to Wumpus are smelly.
• Squares adjacent to Pit are breezy.
• Glitter if and only if Gold is in the same square.
• Shooting kills the Wumpus if you are facing it.
• Shooting uses up the only Arrow.
• Grabbing picks up the Gold if in the same square.

Wumpus World Characteristics

• The world is static: positions of the pits, gold, and monster do not change during the course of a game.

• The actions have deterministic effects.

• This domain is partially observable.
  – Agent does not know the map beforehand, must figure it out based on local perception.

• Belief state for this domain?
Exploring a Wumpus World

A = Agent
B = Breeze
S = Smell
P = Pit
W = Wumpus
OK = Safe
V = Visited
G = Glitter
Exploring a Wumpus World

A = Agent
B = Breeze
S = Smell
P = Pit
W = Wumpus
OK = Safe
V = Visited
G = Glitter
Exploring a Wumpus World

A = Agent
B = Breeze
S = Smell
P = Pit
W = Wumpus
OK = Safe
V = Visited
G = Glitter

Exploring a Wumpus World

A = Agent
B = Breeze
S = Smell
P = Pit
W = Wumpus
OK = Safe
V = Visited
G = Glitter
Exploring a Wumpus World

A = Agent
B = Breeze
S = Smell
P = Pit
W = Wumpus
OK = Safe
V = Visited
G = Glitter
Another example: Using actions to get knowledge

- Smell in (1, 1) ⇒ cannot move
- Can use a strategy of coercion:
  - Shoot straight ahead
  - Wumpus was there ⇒ dead ⇒ safe
  - Wumpus wasn’t there ⇒ safe

What knowledge representation supports this reasoning?

Logic

- Logics are formal languages for representing information such that conclusions can be drawn.
- Logic:
  - Syntax defines which sentences are allowed in the language.
  - Semantics define the "meaning" of sentences.
    - I.e. which sentences are true in a given world.

An analogy: The language of arithmetic

\[ x + 2 \geq y \] is a sentence
\[ x^2 + y > \] 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 \( x=0, y=6 \)
Types of logic

- Logics are characterized by what they allow as “primitives”
  - Ontological commitment: what exists? (facts/objects/time/beliefs)
  - Epistemological commitment: what do we know? (states of knowledge)

<table>
<thead>
<tr>
<th>Language</th>
<th>Ontological commitment</th>
<th>Epistemological comm.</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>

Propositional logic

- Propositional logic is the simplest logic.

- Syntax:
  - Atomic sentences $S_1$, $S_2$, etc. are sentences.
  - If $S$ is a sentence, $\neg S$ is a sentence.
  - If $S_1$ and $S_2$ are sentences, $S_1 \land S_2$ is a sentence.
  - If $S_1$ and $S_2$ are sentences, $S_1 \lor S_2$ is a sentence.
  - If $S_1$ and $S_2$ are sentences, $S_1 \Rightarrow S_2$ is a sentence.
  - If $S_1$ and $S_2$ are sentences, $S_1 \Leftrightarrow S_2$ is a sentence.
Propositional logic: Semantics

• A model specifies true/false value for each propositional symbol.
  E.g.  \( m = \{A=\text{True}, B=\text{True}, C=\text{False}\} \)

• Rules for evaluating truth with respect to model \( m \).
  \[
  \begin{align*}
  \neg S & \quad \text{is true iff} \quad S \text{ is false} \\
  S_1 \land S_2 & \quad \text{is true iff} \quad S_1 \text{ is true} \quad \text{and} \quad S_2 \text{ is true} \\
  S_1 \lor S_2 & \quad \text{is true iff} \quad S_1 \text{ is true} \quad \text{or} \quad S_2 \text{ is true} \\
  S_1 \Rightarrow S_2 & \quad \text{is true iff} \quad S_1 \Rightarrow S_2 \text{ is true} \quad \text{and} \quad S_2 \Rightarrow S_1 \text{ is true}
  \end{align*}
  \]

Interpretations: True or False

Terminology:

• A sentence is valid if it is true in all interpretations.

• A sentence is satisfiable if it is true in at least one interpretation.

• A sentence is unsatisfiable if it is false in all interpretations.

• The truth of a sentence may depend on its interpretation.
  – A sentence may be true in one interpretation and false in another.

E.g. Sentence = “I finished the AI homework”. Is this valid? satisfiable?
Examples

• For each of the following logical statements, say whether it is Valid, Satisfiable or Unsatisfiable:

1. \( A \lor \neg B \)  
   Satisfiable

2. \((A \land B) \lor (\neg A \lor \neg B)\)  
   Valid

3. \( \neg A \Rightarrow A \)  
   Satisfiable

4. \( B \land (B \Rightarrow \neg B) \)  
   Unsatisfiable

Entailment

=> Want a rule for generating new sentences that are always true.

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

   \[ KB \models \alpha \]

E.g., KB containing “I finished the AI homework” and “I am happy”.

What sentences are entailed by this?

   “I finished the AI homework or I am happy”.
Validity and satisfiability

- To check **validity** via inference:
  \[ KB \models \alpha \quad \text{if and only if} \quad (KB \Rightarrow \alpha) \text{ is valid.} \]

- To check **satisfiability** via inference:
  \[ KB \models \alpha \quad \text{if and only if} \quad (KB \land \neg \alpha) \text{ is unsatisfiable.} \quad \text{Proof by contradiction!} \]

Inference procedure

- \( KB \models_{i} \alpha \)
  means sentence \( \alpha \) can be derived from KB by inference procedure \( i \)

- Desired qualities of procedure \( i \):  
  
  **Soundness**: \( i \) is sound if, whenever \( KB \models_{i} \alpha \) is true, it is also true that \( KB \models \alpha \).
  
  *In other words, we only infer necessary truths.*

  **Completeness**: \( i \) is complete if, whenever \( KB \models \alpha \) is true, it is also true that \( KB \models_{i} \alpha \).
  
  *In other words, we can generate all the necessary truths.*
Two kinds of inference methods

1. **Model checking:**
   - Truth table enumeration

2. Application of inference rules

---

Propositional inference

- Truth table method:

  Let \( \alpha = A \lor B \) and \( KB = (A \lor C) \land (B \lor \neg C) \)

  Is it the case that \( KB \models \alpha \) ?

  Check all possible models: \( \alpha \) must be true whenever \( KB \) is true.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>A \lor C</th>
<th>B \lor \neg C</th>
<th>KB</th>
<th>\alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Propositional inference

• Truth table method:

Let \( \alpha = A \lor B \) and \( KB = (A \lor C) \land (B \lor \neg C) \)

Is it the case that \( KB \models \alpha \) ?

Check all possible models: \( \alpha \) must be true whenever \( KB \) is true.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>A \lor C</th>
<th>B \lor \neg C</th>
<th>KB</th>
<th>\alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</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>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</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>
<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>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

Example: Entailment in wumpus world

• Situation after detecting Nothing in [1,1], moving right, and detecting Breeze in [2,1].

• Consider world models for KB assuming only pits.

• 3 Boolean choices \( \Rightarrow \) 8 possible models.
Example: Models in wumpus world

Example: KB in wumpus world

- $\text{KB} = \text{wumpus-world rules} + \text{observations}$
Example: Model checking in wumpus world

- KB = wumpus-world rules + observations
- $\alpha_1 = \text{"[1,2] is safe"}$, KB $\models \alpha_1$, proved by model checking.

$\Rightarrow$ Enumerate states and check.

Recall: Knowledge base KB entails sentence $\alpha$ if and only if $\alpha$ is true in all worlds where KB is true.

Example: Model checking in wumpus world

- KB = wumpus-world rules + observations
- $\alpha_2 = \text{"[2,2] is safe"}$, KB does not entail $\alpha_2$.

Recall: Knowledge base KB entails sentence $\alpha$ if and only if $\alpha$ is true in all worlds where KB is true.
Properties of truth table method

• Soundness? ✅

• Complete? ✅ Checks truth in all possible models.

• Time / space complexity? Very inefficient! \(2^n\) models for \(n\) literals.

Two kinds of inference methods

1. Model checking:
   – Truth table enumeration (sound and complete, but expensive)

2. Application of inference rules:
   – Legitimate (sound) generation of new sentences from old.
   – A **proof** is a sequence of inference rule applications.
   – Can use inference rules as operators in a standard search algorithm.
Normal forms

Application of inference rules often requires sentences to be expressed in standardized forms.

**Conjunctive Normal Form (CNF):** Conjunction of disjunctions of literals
E.g. \((A \lor \neg B) \land (B \lor \neg C \lor \neg D)\)

**Disjunctive Normal Form (DNF):** Disjunction of conjunctions of literals
E.g. \((A \land B) \lor (A \land \neg C) \lor (A \land \neg D) \lor (\neg B \land \neg C) \lor (\neg B \land \neg D)\)

**Horn Form:** Conjunction of Horn clauses (clauses with \(\leq 1\) positive literal
E.g. \((A \lor \neg B) \land (B \lor \neg C \lor \neg D)\)
Often written as set of implications: \(B \Rightarrow A\) and \((C \land D) \Rightarrow B\)

---

Inference rules for propositional logic

- **Resolution (for CNF):** complete for propositional logic

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

- **Modus Ponens (for Horn Form):** complete for Horn KBs

\[
\frac{\alpha_1, \alpha_2, \ldots, \alpha_n, (\alpha_1 \land \alpha_2 \land \ldots \land \alpha_n \Rightarrow \beta)}{\beta}
\]

- Rules can be used with **forward search** or **backward search**.
Forward chaining

- When a new sentence $p$ is added to the KB
  - Look for all sentences that share literals with $p$.
  - Perform resolution.
  - Add new sentences to the KB and continue.

- Two important properties:
  - Forward chaining is data-driven. E.g. inferring properties and categories for new percepts.
  - Forward chaining is an eager method: new facts are inferred as soon as possible.

Backward chaining

- When a query $q$ is asked of the KB
  - If $q$ is in the KB already, return true.
  - Else use resolution for $q$ with other sentences in the KB, and continue from the result.

- Two important properties:
  - Backward chaining is goal-driven, as it centers the reasoning around the query being asked.
  - It is a lazy reasoning method: new facts are only inferred as needed, and only to the extent that they help answer the question.
Forward vs backward chaining: Which is better?

- It depends on the problem at hand.
- **Backward chaining:**
  - parsimonious in the amount of computation done.
  - does not grow the KB as much.
  - focuses on the proof that needs to be generated, so generally more efficient.
  - does nothing until questions are asked.
  - used in proofs by contradiction.

- **Forward chaining:**
  - extends the KB.
  - improves our understanding of the world.
  - used in tasks where focus is on finding model of the world.

Other useful rules

- **And-elimination:**
  \[
  \alpha_1 \land \alpha_2 \land \ldots \land \alpha_n \\
  \overline{\alpha_1, \alpha_2, \ldots, \alpha_n}
  \]

- **Implication elimination:**
  \[
  \alpha \Rightarrow \beta \\
  \overline{\neg \alpha \lor \beta}
  \]

- **De Morgan’s law:**
  \[
  \neg (\alpha \lor \beta) \Leftrightarrow (\neg \alpha) \land (\neg \beta) \\
  \neg (\alpha \land \beta) \Leftrightarrow (\neg \alpha) \lor (\neg \beta)
  \]
Knowledge base:

- HaveAILecture $\Rightarrow$ (TodayIsTuesday $\lor$ TodayIsThursday)
- $\neg$TodayIsThursday
- HaveAILecture $\lor$ HaveNoClass
- HaveNoClass $\Rightarrow$ Sad
- $\neg$Sad

Can you infer what day it is?

\[
\neg$Sad\n\] \hspace{1cm} \neg$HaveNoClass$ \Rightarrow $Sad
\hspace{1cm} $\neg$HaveNoClass
\hspace{1cm} HaveAILecture $\lor$ HaveNoClass
\hspace{1cm} HaveAILecture
\hspace{1cm} (TodayIsTuesday $\lor$ TodayIsThursday)
\hspace{1cm} $\neg$TodayIsThursday
\hspace{1cm} TodayIsTuesday $\lor$ TodayIsThursday
\hspace{1cm} TodayIsTuesday!

Example
Complexity of inference

- What is the complexity of verifying the validity of a sentence of $n$ literals?
  \[ 2^n \]
- What if our knowledge is expressed only in terms of Horn clauses?
  
  **Inference time is polynomial!**
  
  - Every Horn clause establishes exactly one fact.
  - We can state all new facts implied by the KB in $n$ passes.
  - This is why Horn clauses are often used in expert systems.

Example: The Wumpus World

- KB:
  \[
  \neg S_{1,1} \quad \neg S_{2,1} \quad S_{1,2} \\
  \neg B_{1,1} \quad B_{2,1} \quad \neg B_{2,1}
  \]
- Knowledge about the environment:
  \[
  \neg S_{1,1} \implies \neg W_{1,1} \land \neg W_{1,2} \land \neg W_{2,1} \\
  \neg S_{2,1} \implies \neg W_{1,1} \land \neg W_{2,1} \land \neg W_{2,2} \land \neg W_{3,1} \\
  S_{1,2} \implies W_{1,1} \lor W_{1,2} \lor W_{2,2} \lor W_{1,3}
  \]

Now we can use inference rules to find out where the Wumpus is!
Summary of propositional logic

- Propositional logic is very simple!

- Inference is simple: KB + a few basic rules + search algorithm.

- Writing out the KB can be hard!

Important news

- Homework 1 grades and solution will be posted today.


- Midterm is confirmed for Feb.23 at the usual class time.
  - Last name A-O go to MCMED 504.
  - Last name P-Z go to MCMED 521.