Knowledge-based Agents

• Can represent knowledge
• And reason with this knowledge
• How is this different from the knowledge used by problem-specific agents?
  – More general
  – More flexible
Outline

1. Knowledge-based Agents
2. The Wumpus World
3. Logic

Knowledge-based Agents

• Knowledge of problem solving agents is specific and inflexible
• Knowledge-based agents can benefit from knowledge expressed in very general forms, combining information in different ways to suit different purposes
• Knowledge-based agents can combine general knowledge with current percepts to infer hidden aspects of the current state
Knowledge-based Agents

Flexibility of knowledge-based agents:

• Accept new tasks in the form of explicitly described goals
• Achieve competence quickly by being told or learning new knowledge about the environment
• Adapt to changes in the environment by updating the relevant knowledge

Knowledge is definite

• Knowledge of logical agents is always definite
• That is, each proposition is entirely true or entirely false
• Agent may be agnostic about some propositions
• Logic doesn’t handle uncertainty well
The Knowledge Base (KB)

- A knowledge base is a set of “sentences”
- Each sentence is expressed in a knowledge representation language and represents some assertion about the world

The Knowledge Base (KB)

- Need to add new sentences to the knowledge base (this task is called TELL)
- Need to query what is known (this task is called ASK)
Knowledge Base Example

Knowledge Base:
Murderer wasn’t Colonel Mustard
Murderer wasn’t Miss Scarlett
Weapon wasn’t the Gun
Weapon wasn’t the Candlestick
Room wasn’t the Library

When you discover a new fact like “The murder room wasn’t the study”, you would TELL the KB.
You can then ASK the KB what to ask next.

Inference

- Inference: deriving new sentences from old ones
- Must obey fundamental requirement: when one ASKS a question of the knowledge base, answer should follow from what has been TELLeled to the KB previously.
A Generic Knowledge-based Agent

Input: Percept

Knowledge Base

Output: Action

Starts out with background knowledge
A Generic Knowledge-based Agent

1. TELL the KB what it perceives
2. ASK the KB what action it should perform
3. TELL the KB that the action was executed

Input: Percept

Output: Action

Knowledge Base

The Wumpus World

- Wumpus eats anyone that enters its room
- Wumpus can be shot by an agent, but agent has one arrow
- Pits trap the agent (but not the wumpus)
- Agent’s goal is to pick up the gold
The Wumpus World

• **Performance measure:**
  – +1000 for picking up gold, -1000 for death (meeting a live wumpus or falling into a pit)
  – -1 for each action taken, -10 for using arrow

• **Environment:**
  – 4x4 grid of rooms
  – Agent starts in (1,1) and faces right
  – Geography determined at the start:
    • Gold and wumpus locations chosen randomly
    • Each square other than start can be a pit with probability 0.2

The Wumpus World

• **Actuators:**
  – Movement:
    • Agent can move forward
    • Turn 90 degrees left or right
  – Grab: pick up an object in same square
  – Shoot: fire arrow in straight line in the direction agent is facing
The Wumpus World

• **Sensors:**
  – Returns a 5-tuple of five symbols eg. [stench, breeze, glitter, bump, scream] (note that in this 5-tuple, all five things are present. We indicate absence with the value None)
  – In squares adjacent to the wumpus, agent perceives a stench
  – In squares adjacent to a pit, agent perceives a breeze
  – In squares containing gold, agent perceives a glitter
  – When agent walks into a wall, it perceives a bump
  – When wumpus is killed, it emits a woeful scream that is perceived anywhere

The Wumpus World

• Biggest challenge: Agent is ignorant of the configuration of the 4x4 world
• Needs logical reasoning of percepts in order to overcome this ignorance
• Note: retrieving gold may not be possible due to randomly generated location of pits
• Initial knowledge base contains:
  – Agent knows it is in [1,1]
  – Agent knows it is a safe square
The Wumpus World Environment Properties

- Fully or Partially observable?
- Deterministic or stochastic?
- Episodic or sequential?
- Static or dynamic?
- Discrete or continuous?
- Single agent or multiagent?
Wumpus World Example

1st percept is:
[None, None, None, None, None]
(Corresponding to [Stench, Breeze, Glitter, Bump, Scream])

Agent concludes squares [1,2], [2,1] are safe. We mark them with OK. A cautious agent will move only to a square that it knows is OK.

Agent now moves to [2,1]

Wumpus World Example

2nd percept is:
[None, Breeze, None, None, None]

Must be a pit at [2,2] or [3,1] or both. We mark this with a P?.

Only one square that is OK, so the agent goes back to [1,1] and then to [1,2]
Wumpus World Example

3rd percept is:
[Stench, None, None, None, None]

Wumpus must be nearby. Can’t be in [1,1] (by rules of the game) or [2,2] (otherwise agent would have detected a stench at [2,1])

Therefore, Wumpus must be in [1,3]. Indicate this by W!.

Lack of breeze in [1,2] means no pit in [2,2], so pit must be in [3,1].

Wumpus World Example

Note the difficulty of this inference:

• Combines knowledge gained at different times and at different places.
• Relies on the lack of a percept to make one crucial step

At this point, the agent moves to [2,2].
Wumpus World Example

We’ll skip the agent’s state of knowledge at [2,2] and assume it goes to [2,3].

Agent detects a glitter in [2,3] so it grabs the gold and ends the game

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

Logic

Logic must define:

1. Syntax of the representation language
   • Symbols, rules, legal configurations

2. Semantics of the representation language
   • Loosely speaking, this is the “meaning” of the sentence
   • Defines the truth of each sentence with respect to each possible world
   • Everything is either true or false, no in between
Models

• We will use the word model instead of “possible world”
• “m is a model of α” means that sentence α is true in model m
• Models are mathematical abstractions which fix the truth or falsehood of every relevant sentence
• Think of it as the possible assignments of values to the variables
  – E.g. the possible models for x + y = 4 are all possible assignments of numbers to x and y such that they add up to 4

Entailment

α |= β means α entails β i.e. β follows logically from α, where α and β are sentences

Mathematically, α |= β if and only if in every model in which α is true, β is also true.

Another way: if α is true, then β must also be true.
Entailment Applied to the Wumpus World

- Suppose the agent moves to [2,1]
- Agent knows there is nothing in [1,1] and a breeze in [2,1]
- These percepts, along with the agent’s knowledge of the rules of the wumpus world constitute the KB
- Given this KB, agent is interested if the adjacent squares [1,2], [2,2] and [3,1] contain pits.

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

\[2^3 = 8\] possible models because [1,2], [2,2] and [3,1] can take each take values true or false that there is a pit there

The 3 models inside the line marked KB are those in which the KB is true
Entailment Applied to the Wumpus World

Let us consider the models that support the conclusion $\alpha_1 = \text{“There is no pit in [1,2].”}$. We draw a line marked with $\alpha_1$ around these models. In every model in which KB is true, $\alpha_1$ is also true. Therefore $\text{KB } \models \alpha_1$

Entailment applied to the Wumpus World

Now let us consider the models that support the conclusion $\alpha_2 = \text{“There is no pit in [2,2].”}$. We draw a line marked with $\alpha_2$ around these models. In some models in which KB is true, $\alpha_2$ is false. Therefore KB $\not\models \alpha_2$ and the agent cannot conclude that there is no pit in [2,2]
### Modified Wumpus World

- Breeze occurs in squares directly or diagonally adjacent to a pit

### CW: Modified Wumpus Exercise

- KB includes modified rules plus:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>V</td>
<td>OK</td>
</tr>
<tr>
<td>V</td>
<td>OK</td>
<td></td>
</tr>
</tbody>
</table>

- Want to reason about squares [2,2], [2,3], [1,3]. Are these sentences entailed?
  - S1: There is a wumpus in [2,2].
  - S2: There is a pit in [1,3].
Logical inference

• Entailment can be applied to derive conclusions (we call this carrying out logical inference)
• Model checking: enumerates all possible models to check that \( \alpha \) is true in all models in which \( KB \) is true
• If an inference algorithm \( i \) can derive \( \alpha \) from the \( KB \), we write \( KB \models i \alpha \)
• The above is pronounced “\( \alpha \) is derived from \( KB \) by \( i \)” or “\( i \) derives \( \alpha \) from \( KB \)”

Soundness

• An inference algorithm that derives only entailed sentences is called sound or truth-preserving
• Soundness is a good thing!
• If an inference algorithm is unsound, you can make things up as it goes along and derive basically anything it wants to
Completeness

- An inference algorithm is complete if it can derive any sentence that is entailed
- For some KBs, the number of sentences can be infinite
- Can’t exhaustively check all of them, need to rely on proving completeness

In Summary

- Soundness: $i$ is sound if whenever $KB \models_i \alpha$, it is also true that $KB \models \alpha$
- Completeness: $i$ is complete if whenever $KB \models \alpha$, it is also true that $KB \models_i \alpha$
Propositional Logic: Syntax and Semantics

Syntax: Backus-Naur Form grammar of sentences in propositional logic

Sentence → AtomicSentence | ComplexSentence
AtomicSentence → True | False | Symbol
Symbol → P | Q | R | …
ComplexSentence → ¬ Sentence
| ( Sentence ∧ Sentence )
| ( Sentence ∨ Sentence )
| ( Sentence ⇒ Sentence )
| ( Sentence ⇔ Sentence )
Atomic Sentences

• The indivisible syntactic elements
• Consist of a single propositional symbol
e.g. P, Q, R that stands for a proposition
that can be true or false e.g. P=true, Q=false
• We also call an atomic sentence a literal
• 2 special propositional symbols:
  – True (the always true proposition)
  – False (the always false proposition)

Complex Sentences

• Made up of sentences (either complex or atomic)
• 5 common logical connectives:
  – \( \neg \) (not): negates a literal
  – \( \land \) (and): conjunction e.g. \( P \land Q \) where P and Q are
called the conjuncts
  – \( \lor \) (or): disjunction e.g. \( P \lor Q \) where P and Q are called
the disjuncts
  – \( \Rightarrow \) (implies): e.g. \( P \Rightarrow Q \) where P is the
premise/antecedent and Q is the conclusion/consequent
  – \( \Leftrightarrow \) (if and only if): e.g. \( P \Leftrightarrow Q \) is a biconditional
Precedence of Connectives

- In order of precedence, from highest to lowest: ¬, ∧, ∨, ⇒, ⇔
- E.g. ¬P ∨ Q ∧ R ⇒ S is equivalent to 
  (¬P) ∨ (Q ∧ R) ⇒ S
- You can rely on the precedence of the connectives or use parentheses to make the order explicit
- Parentheses are necessary if the meaning is ambiguous

Semantics (Are sentences true?)

- Defines the rules for determining if a sentence is true with respect to a particular model
- For example, suppose we have the following model: P=true, Q=false, R=true
- Is (P ∧ Q ∧ R) true?
Semantics

For atomic sentences:
• True is true, False is false
• A symbol has its value specified in the model

For complex sentences (for any sentence S and model m):
• \( \neg S \) is true in m iff S is false in m
• \( S_1 \land S_2 \) is true in m iff \( S_1 \) is true in m and \( S_2 \) is true in m
• \( S_1 \lor S_2 \) is true in m iff \( S_1 \) is true in m or \( S_2 \) is true in m
• \( S_1 \Rightarrow S_2 \) is true in m iff \( S_1 \) is false in m or \( S_2 \) is true in m
  i.e., can translate it as \( \neg S_1 \lor S_2 \)
• \( S_1 \Leftrightarrow S_2 \) is true iff \( S_1 \Rightarrow S_2 \) is true in m and \( S_2 \Rightarrow S_1 \) is true in m

Note on implication

• \( P \Rightarrow Q \) seems weird…doesn’t fit intuitive understanding of “if P then Q”
• Propositional logic does not require causation or relevance between P and Q
• Implication is true whenever the antecedent is false (remember \( P \Rightarrow Q \) can be translated as \( \neg P \lor Q \))
  – Implication says “if P is true, then I am claiming that Q is true. Otherwise I am making no claim”
  – The only way for this to be false is if P is true but Q is false
Truth Tables for the Connectives

<table>
<thead>
<tr>
<th>P</th>
<th>¬P</th>
<th>P &amp; Q</th>
<th>P ∨ Q</th>
<th>P → Q</th>
<th>P ↔ Q</th>
</tr>
</thead>
<tbody>
<tr>
<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>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>true</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>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>

With the truth tables, we can compute the truth value of any sentence with a recursive evaluation e.g.

Suppose the model is P=false, Q=false, R=true

¬P ∧ (Q ∨ R) = true ∧ (false ∨ true) = true ∧ true = true

The Wumpus World KB (only dealing with knowledge about pits)

For each i, j:
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]

The KB contains the following sentences:

1. There is no pit in [1,1]:
   \[ R_1: \neg P_{1,1} \]

2. A square is breezy iff there is a pit in a neighboring square: (not all sentences are shown)
   \[ R_2: B_{1,1} \iff P_{1,2} \lor P_{2,1} \]
   \[ R_3: B_{2,1} \iff (P_{1,1} \lor P_{2,2} \lor P_{3,1}) \]
The Wumpus World KB

3. We add the percepts for the first two squares ([1,1] and [2,1]) visited in the Wumpus World example:
   \[ R_4: \neg B_{1,1} \]
   \[ R_5: B_{2,1} \]

The KB is now a conjunction of sentences \( R_1 \land R_2 \land R_3 \land R_4 \land R_5 \) because all of these sentences are asserted to be true.

Inference

• How do we decide if KB |= \( \alpha \)?
• Enumerate the models, check that \( \alpha \) is true in every model in which KB is true

<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>R_1</th>
<th>R_2</th>
<th>R_3</th>
<th>R_4</th>
<th>R_5</th>
<th>KB</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>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</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>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</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>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</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>
<td>true</td>
<td>true</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>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</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>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>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<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>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<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>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<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>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<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>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>

25
Inference

• Suppose we want to know if KB |= ¬P₁,₂?

• In the 3 models in which KB is true, ¬P₁,₂ is also true

<table>
<thead>
<tr>
<th>B₁,₁</th>
<th>B₂,₁</th>
<th>P₁,₁</th>
<th>P₁,₂</th>
<th>P₂,₁</th>
<th>P₂,₂</th>
<th>P₃,₁</th>
<th>R₁</th>
<th>R₂</th>
<th>R₃</th>
<th>R₄</th>
<th>R₅</th>
<th>KB</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>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</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>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</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>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>true</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>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

Things you should know

• Properties of a knowledge-based agent
• What a knowledge-base is
• What entailment and inference mean
• Desirable properties of inference algorithms such as soundness and completeness