Without facts, the decision cannot be made logically.

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 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

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
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

Knowledge Base Example

TELL

Knowledge Base:
- Victim was Professor Plum
- 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

ASK

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 TELLed 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

The Wumpus World

- **Performance measure:**
  - +1000 for picking up gold, -1000 for death
  - -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
    - Death for meeting a live wumpus or falling into a pit
  - 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 adjacent to the 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 (in terms of actions)?
- 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]

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]

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].

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].
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.

Entailment Applied to the Wumpus World

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 \( KB \models \alpha_1 \).

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].

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 \( KB \models_i \alpha \)
Correspondence to the Real World

If the KB is true in the real world, then any sentence derived from the KB by a sound inference procedure is also true in the real world.

Grounding

- Defined as the connection, if any, between logical reasoning processes and the real environment in which the agent exists
- How do we know that the KB is true in the real world?
- Deep philosophical question
- We’ll respond with the following:
  - Rely on sensors to accurately perceive the world
  - Learning produces general rules (derived from perceptual experience). Learning can be fallible but it has the potential to fix its mistakes.

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