Agents that Reason Logically

Lecture 4

Outline

Outline

The Wumpus-less Wumpus World

- Version 1: Location of gold is known.

	G		
A			

- Possible strategy: Use greedy search with h as city block distance.

The Wumpus-less Wumpus World

- Version 2: Location of gold is unknown.
- But agent can sense a glitter when it is in the same cell as the gold.
- Will search help in this case?
- What should the agent do?
  - Perhaps, systematically visit all cells.

The Wumpus World(s)

S		B	P
$\mathfrak{W}$	S G B	P	B
S		B	
A	B	P	B

Goal

- The agent goal is to find the gold and bring it back to the start, where it will climb out of the cave, as quickly as possible, and without getting killed.
- Performance measure:
  - $+1000$ for climbing out of the cave with the gold.
  - -1 for each action performed.
  - -10000 for getting killed.

Percepts

- In the cell containing the wumpus, and in the adjacent cells, the agent will perceive a stench.
- In the cells adjacent to a pit, the agent will perceive a breeze.
- In the cell containing the gold, the agent will perceive a glitter.
- If the agent walks into a wall, it perceives a bump.
- When the wumpus is killed, the agent perceives a scream.
- The agent cannot perceive its location.

Thus, a percept is a quintuple

$$\langle\ Stench, Breeze, Glitter, Bump, Scream\ \rangle$$

Actions

- Forward.
- Right.
- Left.
- Grab.
- Shoot.
- Climb.

Can search help the wumpus agent?

What is a Wumpus-World State?

- States encode deeper knowledge of the environment.
- For example, sensing no stench while in [1,1], the agent should *know* that the wumpus is neither in [1,2] nor in [2,1].
- Given the percept sequence, a state would include things like:
  - Cell [i,j] is safe.
  - There is a pit in cell [i,j].
  - The wumpus is in cell [i,j] or cell [k,l].

Logical Reasoning

- In order to update the state, given a new percept, the agent needs the following:
  1. Background knowledge linking percepts to possible contents of cells.
  2. Some way of representing this knowledge.
  3. Some way of mapping percepts to structures of the same representation.
  4. A method that manipulates these structures to update the agent's knowledge of the state of the environment.
- That is, the agent needs full-fledged logical reasoning.

Outline

What is Logic?

- There are different uses of the word “logic”.
- But think of “a logic” as a language used to represent knowledge.
  - For example, knowledge of the Wumpus-world agent.
- It is an artificial language, it should have precise syntax and semantics.
- In addition, it should have a proof theory.

Syntax

- The syntax of a language specifies valid *forms* of expression in the language.
- We typically specify the syntax in terms of how expressions will look on paper.

In case of computer implementation:

- The real syntax is in the form of certain configurations of data structures in the agent program.
- Ultimately, it is a pattern of electrons in computer memory.

Example

How is a sentence like $\forall x(P(x) \Rightarrow \exists y Q(x, y))$ represented inside a computer?

Semantics

- The semantics of a language determines the entities, in the world, expressions of the language denote.
- Logics have compositional semantics: the meaning of an expression is a function of the meaning of its parts.
- Some expressions denote individuals in the world. Others (sentences) denote facts.
- Each sentence makes a claim about the world.
- A sentence is true if what it claims about the world is indeed the case.
- The agent knowledge is represented as a set of sentences: a knowledge base (KB).
- If a configuration corresponding to a sentence exists in the agent's knowledge base, we say that the agent *believes* the sentence.

Proof Theory

- The proof theory identifies a set of inference rules that license the derivation of sentences with a certain form from sentences of certain forms.

Example

The rule

$$\frac{\phi \Rightarrow \psi, \phi}{\psi}$$

licenses the derivation of a sentence ψ from a sentence ϕ and the sentence $\phi \Rightarrow \psi$

- We write $KB \vdash_I \phi$ whenever ϕ may be derived from the knowledge base, KB, using the rules of inference I.
- Typically, sentences derived from the KB are added to the KB.

Logical Implication

- A sentence ϕ logically implies (or entails) a sentence ψ if every state of affairs in which ϕ is true is a state of affairs in which ψ is true.

Example

"All men are mortal and Socrates is a man" logically implies "Socrates is mortal"

- Logical implication does not depend at all on the proof theory, only on the semantics of the language.
- We write $KB \models \phi$ whenever the conjunction of all sentences in KB logically implies ϕ.

$\models$ and $\vdash$

- An agent that reasons logically needs to compute the logical implications of its KB.
  - That is, it should be able to generate new sentences that are necessarily true, assuming the truth of what is in the KB.
  - Note: the agent will then be rational, but not necessarily well-informed.
- However, computing using $\models$ requires access to the world; you cannot put the world inside the agent.
- Rather, all reasoning should be done on a representation of the world: the KB.
- Move from $\models$ to $\vdash$.

Soundness and Completeness

- A logic is sound if $KB \vdash_I \phi$ implies $KB \models \phi$.
  - The rules of inference in I should be sensitive to the semantics; they should be truth preserving.
- A logic is complete if $KB \models \phi$ implies $KB \vdash_I \phi$.
- In practice, soundness is easier to achieve.

A Knowledge-Based Agent

function KB-AGENT(*percept*) **returns** action
    static: *KB*
        t //A counter, initially 0

    TELL(*KB*, MAKE-PERCEPT-SENTENCE(*percept, t*))
    action $\longleftarrow$ ASK(*KB*, MAKE-ACTION-QUERY(*t*))
    TELL(*KB*, MAKE-ACTION-SENTENCE(*action, t*))
    $t \longleftarrow t + 1$
    return *action*

Outline

FOL Vocabulary

- The language of FOL is made up of the following symbols.
  1. Constants: $A, B, John, Wumpus, Pit1$.
     - Each constant denotes an individual.
  2. Variables: $x, y, t, z, \ldots$
  3. Predicate symbols: $Alive, Breezy, Adjacent, Likes$.
     - Each predicate symbol has an arity.
     - Unary predicate symbols denote properties.
     - n-ary predicate symbols ($n > 1$) denote relations.
  4. Function symbols: $LocationOf, OrientationOf$.
     - Each function symbol has an arity.
     - n-ary function symbols denote functions.
  5. Syncategorematic Symbols: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow, \forall, \exists, =, (,), ,$

FOL Terms

- FOL expressions are either terms or sentences.
- A term is either
  1. a constant,
  2. a variable,
  3. an expression of the form $f(t_1, t_2, \ldots, t_n)$, where f is an n-ary function symbol and t_i is a term.

Example

What are the terms of a FOL with two constants, *Wumpus* and *Agent*; two variables, x and y; and two function symbols, *LocationOf* *OrientationOf*?

FOL Sentences

- Sentences have several forms:
  1. Atomic: $P(t_1, t_2, \ldots, t_n)$, where P is an n-ary predicate symbol and t_i is a term.
  2. $t_1 = t_2$, where t_1 and t_2 are terms.
  3. $\neg\phi$, where ϕ is a sentence.
  4. $\phi\ op\ \psi$, where ϕ and ψ are sentences and $op \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$.
  5. $Q\ v\ (\phi)$, where ϕ is a sentence, v is a variable, and $Q \in \{\forall, \exists\}$.

Informal Semantics

- Interpreting FOL expressions is restricted to certain set of individuals: the domain of discourse.
  1. $P(t_1, t_2, \ldots, t_n)$ is true iff the relation denoted by P holds among the individuals denoted by $t_1, t_2, \ldots, t_n$
  2. $t_1 = t_2$ is true iff the individual denoted by t_1 is identical to the individual denoted by t_2.
  3. $\neg\phi$ is true iff ϕ is false.
  4. ϕ *op* ψ, where *op* $\in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$, is true as per the truth table of *op*.
  5. $\forall\, v\ (\phi)$ is true iff ϕ is true no matter which individual in the domain is referred to by v.
  6. $\exists\, v\ (\phi)$ is true iff ϕ is true for at least one choice (from the domain) for a referent of v.

Commitments of FOL

- Every logic has both ontological and epistemological commitments.
- Ontological commitments refer to what the logic assumes to be the structure of reality.
  - FOL assumes reality to consist of individuals having (or not) properties and standing (or not) in relations to one another.
- Epistemological commitments refer to what the logic assumes to be the states of knowledge of an agent.
  - FOL assumes the agent either believes a sentence to be true, believes it to be false, or has no clue.
- Contrast to propositional, probabilistic, and fuzzy logic.

Outline

Objective

- We would like to build a knowledge base for reasoning about sets.
- We, thus, need two things:
  1. A suitable language of FOL to represent knowledge about sets.
  2. A collection of axioms, written in this language, that represent what we know about the domain of sets.

Ontological Commitments

- The world consists of individuals. Some of them are sets. One of them is the empty set.
- Sets are either empty or non-empty.
- There is exactly one empty set.
- Individuals stand in membership relations to sets.
- Sets stand in subset relations to other sets.

Language: Variables

- We shall use x, y, x_i, y_i, and s_i ($i \in \mathbb{N}$) as variables.
- We shall consistently use s_i to refer to sets.
- However, this is just a matter of readability:
  - We can use any variable to refer to sets.
  - A variable s_i by itself does not exclusively refer to a set; we have to explicitly state that it does.
  - (In a *sorted logic* we do not have to.)

Language: Constants

- Should introduce as many constants as there are individuals in the domain we would like to distinguish.
- In our case, there is only one such individual: the empty set.
- Thus, introduce one constant, *EmptySet*.
- Once we have encoded our simplistic set theory, we might want to reason with it about particular sets; we shall introduce constants to name these on the fly. (Typically, *A*, *B*, *C*, etc.)

Language: Predicate Symbols

- Which properties and relations are interesting to us?
  - Properties: being a set.
  - Relations: membership and subset.
- Thus,
  - *Set*, where $Set(t)$ means that t denotes a set.
  - *Member*, where $Member(t_1, t_2)$ means that t_1 denotes a member of whatever t_2 denotes.
  - *Subset*, where $Subset(t_1, t_2)$ means that t_1 denotes a subset of whatever t_2 denotes.

Language: Function Symbols

- We mainly need functions to be able to construct sets.
- There are (at least) three ways to construct sets:
  1. By adding an element to a set.
  2. By unioning two sets.
  3. By intersecting two sets.
- Thus,
  - *Adjoin*, where $Adjoin(t_1, t_2)$ denotes the set resulting from adding the referent of t_1 to the set referred to by t_2.
    - What if t_2 does not refer to a set?
  - *Union*, where $Union(t_1, t_2)$ denotes the union of the sets denoted by t_1 and t_2.
  - *Intersection*, $Intersection(t_1, t_2)$ denotes the intersection of the sets denoted by t_1 and t_2.

Axioms

1. $\forall s[Set(s) \Leftrightarrow ((s = EmptySet) \vee (\exists x, s_2[Set(s_2) \wedge s = Adjoin(x, s_2)]))]$
2. $\neg\exists x, s[Adjoin(x, s) = EmptySet]$
3. $\forall x, s[Member(x, s) \Leftrightarrow s = Adjoin(x, s)]$
4. $\forall x, s[Member(x, s) \Leftrightarrow \exists y, s_2[s = Adjoin(y, s_2) \wedge (x = y \vee Member(x, s_2))]]$
5. $\forall s_1, s_2[Subset(s_1, s_2) \Leftrightarrow \forall x[Member(x, s_1) \Rightarrow Member(x, s_2)]]$
6. $\forall s_1, s_2[(s_1 = s_2) \Leftrightarrow (Subset(s_1, s_2) \wedge Subset(s_2, s_1))]$
7. $\forall x, s_1, s_2[Member(x, Union(s_1, s_2)) \Leftrightarrow (Member(x, s_1) \vee Member(x, s_2))]$
8. $\forall x, s_1, s_2[Member(x, Intersection(s_1, s_2)) \Leftrightarrow (Member(x, s_1) \wedge Member(x, s_2))]$

Theorems

- Given the above axioms and a reasonable set of sound inference rules, we can prove interesting theorems:
    - $\forall s_1, s_2 Subset(Intersection(s_1, s_2), s_1)$.
    - $\forall s_1, s_2 Subset(s1, Union(s_1, s_2))$.
    - $\forall s Subset(EmptySet, s)$
- Can we prove De Morgan's laws?