CSE560 Artificial Intelligence

- http://www.csee.ogi.edu/class/cse560
- heemanp@ohsu.edu
- Textbook:
  Computational Intelligence: A Logical Approach
  David Poole, Alan Mackworth and Randy Goebel
  Oxford University Press. 1998

Overview

⇒ Agent Approach
- Symbolic Reasoning
- Declarative Approach
- Example Problem
- Bookkeeping

Artificial Intelligence

- Goal:
  - Understand how intelligent behavior is possible
  - i.e. Come up with a theory that explains intelligent behavior
- Methodology:
  - Design, build and experiment with computation systems that perform tasks commonly viewed as intelligent

Agent Approach

- This course focuses on building agents
- Agent is something that acts in an environment
  - It has sensors to determine the state (or partial state) of the environment
  - It has effectors to manipulate the environment
- Examples
  - Robot
  - Expert system (uses language as its input and output)
- An intelligent agent is one that behaves intelligently
  - Intelligently $\Rightarrow$ Purposely
  - We should be able to ascribe to it some goal
  - Its actions should be directly towards achieving its goal

Mapping Inputs to Outputs

- For agents with simple goals in simple domains
  Could just map current sensor values to actions
- Mapping could be done by a neural network,
  - assuming there is training data (mapping inputs to outputs)
  - and enough of it (to adequately cover enough permutations of the inputs)

Overview

- Agent Approach

⇒ Symbolic Reasoning

- Declarative Approach
- Example Problem
- Bookkeeping

Internal State

- For more complex goals in more complicated domains
  Agent might need more than current sensor values
- Might want to know previous sensor values
  - Agent in a maze cannot sense entire world each time
- Might want to know its previous actions
  - Want to avoid trying the same thing over and over again
- Might want to make assumptions
  - Reason and make assumptions about other agents' beliefs and goals
- Might want to commit itself to future actions
  - In chess, might want to form a strategy and follow through with it

Representing and Updating Internal State

- Problems:
  - How do we represent the internal state?
  - How do we update the state and determine the next action?
- Need a theory (formal specification)
  - and implementation of this theory

Procedural Approach

- Make a big program
- Emphasis is on making the code work
- Intelligence of the agent is manifest by the code
- But software engineering nightmare
  - Program can quickly become huge: difficult to maintain and add to

Overview

- Agent Approach
- Symbolic Reasoning

⇒ Declarative Approach

- Example Problem
- Bookkeeping

Another Solution

- Break problem broken down into some subproblems
  - Isn't this how computer science has evolved?
  - How many layers are there in Internet communication?
- Although might not be most efficient way of doing things, some redundancy can make complex problems tractable

Declarative Approach: Symbols

- Model the internal state with symbols
  - Symbols will have meaning to us (us = the designer)
  - Meaning of symbols should be unambiguous, unlike English
- Need to express complex relations with minimum of symbols
  - Need language for representing the internal state
- Example

  have(milk) $\wedge$ *have(cereal)* $\wedge$ *want(sugar)*
  - *A* $\wedge$ *B* means A and B are both true for the agent
  - *have(X)* means agent has X in its procession
  - *want(X)* means agent wants X in its procession
  - Don't need a symbol such as *havemilkhavecerealwantsugar*

Declarative Approach: Rules

- Not only can facts be represented with symbols
  But also more general knowledge can be represented
- Examples:
  - having cereal means having food
    $have(cereal) \rightarrow have(food)$
    - + Use additional connector in representing rules
  - if X is connected to Y and there is a path from Y to Z, then there is a path from X to Z
    $connected(X,Y) \wedge path(Y,Z) \rightarrow path(X,Z)$
    - + Use tokens that start with an uppercase letter for variables
  - if X is connected to Y, then there is a path between them
    $connected(X,Y) \rightarrow path(X,Y)$

Declarative Approach: Reasoning Algorithm

- Rules encode how new symbols are created from existing ones
- From rules and facts, we should be able to make conclusions that follow from internal state
  - Facts that are not explicitly represented
  - Assumptions that seem reasonable
  - Plans of actions
  - Action to perform right now
- Reasoning algorithm
  - Makes conclusions from rules and facts

Declarative Approach

- Intelligence is in
  - Having an appropriate language for representing internal state
  - Being able to reason about symbols to form new symbols
- Knowledge engineer:
  - Decides the set of facts and rules for a particular domain
- Programmer:
  - Constructs algorithms that can take arbitrary sets of facts and rules to make conclusions
  - Can reuse algorithm over and over again for any domain

Fundamental Issues

- What are good languages for representing
  - the facts of an agent's internal state?
  - the rules that define the agent's reasoning?
- What are good algorithms that can produce the conclusions that correspond to reasoning?
  - What do we mean by a 'good' algorithm?
  - What constraints are needed on the language that allow good algorithms?

Overview

- Agent Approach
- Symbolic Reasoning
- Declarative Approach

⇒ Example Problem

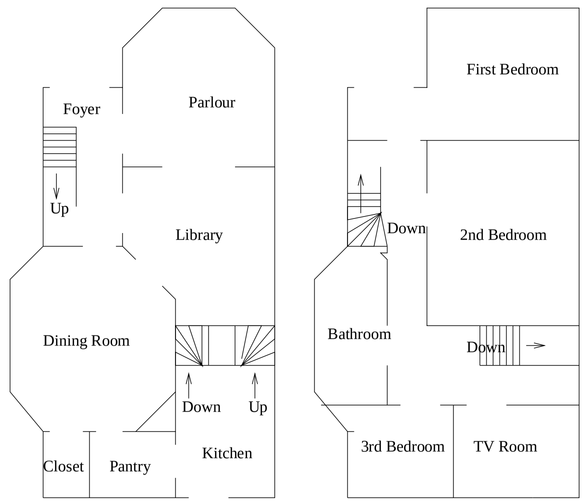
- Bookkeeping

Burglar Alarm

- Most burglar alarms require user to explicitly arm them
  - Turn motion detectors on for the first floor
- Suboptimal
  - Must remember to arm when going to bed, and leaving the house
  - Entire first floor is either armed or disarmed
- Would be nice if it could reason about motion in a house
  - Decide if it is expected or unexpected motion
  - If unexpected, challenge person to enter code

The House

- Detectors in
  Foyer, Library,
  Dining room,
  Kitchen,
  Backstairs
  Hallway



Reasoning About Motion

- If motion in dining room and recent motion in foyer, library, or kitchen, probably just someone walking between rooms
- If motion in dining room and no recent motion in foyer, library or kitchen, not good
- But how recent is recent enough?
- Lets make some assumptions

Motionless

- Motion detectors only notice major motion, like walking
  - if someone is sleeping, typing, reading, might not be noticed
- For each room, lets give an upper bound for how long someone will not be detected by motion detector

2nd Bedroom	8 hours
Library	1 hour
Diningroom	30 minutes
Kitchen	10 minutes
Foyer	5 minutes
Hallway	5 minutes
Stairs	5 minutes

Movement Between Rooms

More Assumptions ...

- When someone is moving out of a room with a motion detector, the detector will notice them
- When someone is moving into a room with a motion detector, detector will notice them
- When moving between rooms, a person will be out of view of the motion detector for at most 10 seconds

Duration in Rooms With No Detector

More Assumptions ...

- Motion detectors do not cover all rooms
- When figuring out if person in kitchen might have went through pantry to go to dining room, take into account how long they might spend in pantry

Pantry	20 minutes
Closet	1 minute
1st bedroom	10 hours
3rd bedroom	10 hours
TV room	90 minutes
Parlour	30 minutes
Bathroom	1 hour

Paths Through House

- Luckily, enough motion detectors that someone can not cycle between rooms ($A \rightarrow B \rightarrow A$) without being detected

Towards a Sophisticated Alarm

- When a motion detector triggers in a certain room
  Need to reason about whether it is expected or not
  - Could someone already in the house have caused detector to go off?
- Is there a path in the house
  from the past motion to present motion
  within the time frame specified by the assumptions
  without going by another motion detector

  path: sequence of rooms where each is connected to the next
- Above is another rule that is part of the internal state of the agent

Action rules

- If there is motion and the motion is not expected
  then go into challenge mode
- If there is motion
  then remember which room it occurred in (for future reasoning)
- If code entered
  then go out of challenge mode
- If in challenge mode for 10 seconds
  then sound alarm

Declarative Approach

Determine a **language** that is rich enough to specify

- Knowledge about the house
  - + what rooms are connected to what; which rooms have detectors
- Assumptions
  - + how long motionless in each room with detector; how long in rooms without
  - + 10 seconds between rooms
- Expected motion rule
  - + path through house within the time frames without going by motion detector

Need **algorithm** that can reason about the internal state to apply expected motion rule and the action rules

Overview

- Agent Approach
- Symbolic Reasoning
- Declarative Approach
- Example Problem

⇒ Bookkeeping

Course Outline

- Knowledge and Reasoning
  - First Order Logic: Syntax, Semantics, Inference
  - Search procedures
- Planning
  - Agents might have goals, have knowledge about actions
- Non-monotonic Reasoning
  - Making assumptions and learning new information
- Belief and Knowledge
  - Representing and reasoning about beliefs/knowledge of other agents
- Building Agents
  - Tie together concepts into a system

Grading

Assignments and Projects	50%
Midterm	25%
Final	25%

- Assignments due at beginning of class, in hardcopy
  - Mixture of written answers and programming assignments
  - Programming in Tcl and a small amount in Prolog
- First assignment has some Tcl programming
  - You can start this assignment already
  - Information on the Web about downloading Tcl
- Bulleton board: http://www.cslu.ogi.edu/forum

Academic Integrity

- See http://www.ogi.edu/students/integrity.shtml
- You can do the homeworks with your colleagues
  - But, you cannot bring any part of your homework into the meeting
  - You cannot bring anything written out of your meeting
- After the meeting, you rehash the solution from scratch
  If you can do this, then you have learned,
  Which is the point of taking the course.
- Corollary
  - Unless both people have photographic memories, homework assignments should look different

Reading Assignments

- We will be following the textbook closely
- You are responsible for material in the textbook
- Reading assignments are posted on the course website
  - Read chapter 1, 2.1-2.5 for next class