Introduction to Knowledge Representation and Prolog

Welcome

- CST8503: Knowledge Representation and Reasoning
- Meet your Professor

  Todd Kelley

  Office: T315

  Phone: 613-727-4723 x7474

  Email: kelleyt@algonquincollege.com
- Contact your Professor
  - email me with enquiries (can expect reply same or next day)
  - email me to arrange office-hour style meetings

Welcome

- Student in the University of Toronto Knowledge Representation group '91-'99
  - Presented papers at
    - KR '96 in Boston
    - Commonsense '96 at Stanford

- Research Interests
  - Automated Reasoning about Physical Systems

Logistics

- Weekly schedule
    - Lecture Thursdays 2-4pm in A1120
    - Section 301 Labs Thursdays 11:30am – 1:30pm in J210
    - Section 302 Labs Tuesdays 5:00-7:00pm in B220
    - One hour (average) of asynchronous (Hybrid) activity
    - Let's look at the Weekly Schedule on Brightspace
- Let's look at the submission requirements on Brightspace (Weekly Schedule)
- Late lab/assignment submissions are not accepted but you have a grace period (8:00am following business day) for submissions.
- Demonstrations must be done before the end of the lab period following the due date.

Lesson Overview (Agenda)

In the following lesson, we will explore:

1. Knowledge Representation vs Machine Learning vs Artificial Intelligence
2. Declarative Knowledge
3. Declarative Programming
4. Prolog Programming

Knowledge Representation

When we are first learning about the following topics, including their similarities and differences, we will notice some distracting words:

Artificial Intelligence

Knowledge Representation

Machine Learning

We all have intuitions about the definitions of "intelligence", "knowledge", and "learning", but those intuitions are not necessarily helpful in distinguishing between what these topics mean in practice

Artificial Intelligence

Artificial Intelligence Software: a category of software solutions to problems that have often been thought to require or benefit from intelligence.

But what is intelligence?

Short answer: you already know vaguely what intelligence is, and it's difficult to know more precisely.

Alan Turing tried to characterize what intelligence is as mathematically as he could, using what we know as the Turing Test for Artificial Intelligence

Turing Test for intelligence

Alan Turing tried to characterize intelligence as mathematically as he could with the Turing Test:

Under certain conditions, if the software system convinces you that it has intelligence, then that software system passes the Turing Test for intelligence.

Passing the Turing Test is thought to require intelligence, because even if the intelligence is faked, it takes intelligence to successfully fake intelligence

The above sentence puts a large burden on "successfully fake"

Artificial Intelligence (for our purposes)

A vague definition of intelligence (what you already think it is) is sufficient for our purposes, and the Turing Test does not help us much.

Artificial Intelligence Software is a broad category of software solutions that involve techniques from some combination of

- Machine Learning (other courses in the Program of Study)
    - Supervised
    - Unsupervised
    - Reinforcement Learning (Level II)
- Knowledge Representation (this course)

Machine Learning versus Knowledge Representation

At some level, Machine Learning is about **finding patterns in data**:

- Supervised Learning: labeled data sets containing explicit examples are used to train the system what patterns to look for
- Unsupervised Learning: the system looks for patterns in the data using an algorithm, but without explicit examples of what to look for
- Reinforcement Learning: an agent-based paradigm where repeatedly an action is chosen in a state according to a policy to receive a reward
    - data is the history of rewards, observations, actions
    - the patterns in the data are learned by trial and error as the history progresses

Machine Learning versus Knowledge Representation

Knowledge Representation is about **declarative** forms of knowledge suitable for processing by dedicated reasoning engines.

Declarative knowledge : facts and rules (knowledge) that are written in a language (declared)

KR: the patterns (knowledge) is given to the system directly in the form of facts and rules – the system (interpreter) knows how to make use of the patterns without learning them

Machine learning: data comes first - the patterns are in the data set with the labels in the case of supervised learning or often simulators in the case of reinforcement learning, and algorithms discover the patterns

Time to check your learning!

Let’s see how many key concepts you recall by answering the following questions!

- How does the computer system obtain the relevant patterns in the data:
  - with supervised Machine Learning?
  - with unsupervised Machine Learning?
  - with Declarative programming?

Declarative Knowledge

If you're setting out to declare knowledge, then what language should you use to declare things? There are languages that are designed for that purpose: knowledge representation languages.

KR languages are often based on logic

Some of the design considerations of KR languages:

- Need to be able to represent the following things
    - State of the world
    - Actions that can happen
    - Results of Actions in changing the state of the world

Declarative vs Procedural

Imagine a pseudo-program to make a cup of instant coffee:

Declarative Program (a set of facts/rules):

- The solution is a cup with hot brown water that tastes like coffee
- Boiling water in a kettle results in hot water
- Putting a spoonful of instant coffee into hot water results in brown water that tastes like coffee
- Pouring boiling water in a cup results in a cup with hot water

Etc… Question: does changing the order of these bullets change the solution? No, in the sense that facts are true no matter the order

Declarative vs Procedural (cont'd)

Procedural Solution to Coffee problem:

1. Take cup out of cupboard and place on counter
2. Fill kettle with water
3. Plug in kettle
4. Place spoon of instant coffee into cup on the counter
5. Wait for kettle to boil
6. Pour water from kettle into cup

Question: how does the meaning of step 6 change when we swap the order of steps 5 and 6?

Declarative Programming

Declarative knowledge has a procedural interpretation.

If we combine declarative knowledge with properly implemented system with a theoretical foundation, for example a theorem prover like Prolog, we can consider the declarative knowledge to be a program.

We run our program by issuing a query to the theorem prover (Prolog in our case). Prolog uses the declarative knowledge (the prolog program) to derive by inference, the conditions (variable bindings) under which the query is true.

Time to check your learning!

Let’s see how many key concepts from topic 2 you recall by answering the following questions!

- What are three things that Knowledge Representation languages need to represent?
- What else is needed besides Declarative Knowledge in order for a Declarative program to run?

Facts and rules Programming

Forall X, Y Parent(X,Y)->Ancestor(X,Y).

Forall X,Y Parent(X,Y) and Ancestor(Y,Z)->Ancestor(X,Z).

Parent(john,joe).

Parent(joe,bill).

Parent(bill,jill).

Ancestor(joe,sally).

Conclusion

In this lesson, you learned…

- Knowledge Representation and Machine Learning differ in where the patterns come from
- Knowledge Representation languages can be expressed in the form of Prolog programs
- Prolog programs combine with a prolog interpreter to form a running program