CS5100: Foundations of Artificial Intelligence

First Order Logic

Dr. Rutu Mulkar-Mehta

Lecture 7

Administrative

- EC1 (Constraint Satisfaction Problems) is graded and handed back to you
- Project 1: Grading is complete. You should have received an email with your score.
- Project 2 is out last week. Please start EARLY!
- Next Week – Guest Speaker from Voicebox Technologies (Dr. Phil Smith)
- In Class Assignment – Today and Next week!

Last Week

- First Order Logic
  - Syntax and Semantics
  - Quantifiers
  - Unification
  - Inference using forward chaining
  - Inference using backward chaining

Constants, Functions, Predicates

- Constant symbols, which represent individuals in the world
  - Mary
  - 3
  - Green
- Function symbols, which map individuals to individuals
  - father-of(Mary) = John
  - color-of(Sky) = Blue
- Predicate symbols, which map individuals to truth values
  - greater(5,3) = true
  - green(Grass) = true
  - color(Grass, Green) = true

Variables, Connectives and Quantifiers

- Variable symbols
  - They can hold any value
  - E.g., x, y, foo
- Connectives
  - Same as in PL: not (¬), and (^), or (v), implies (→), if and only if (⇔)
- Quantifiers
  - Universal (Ax)
  - Existential (Ex)

Quantifier Scope

- Switching the order of universal quantifiers does not change the meaning:
  (Ax)(Ay)P(x,y) → (Ay)(Ax) P(x,y)
- Similarly, you can switch the order of existential quantifiers:
  (Ex)(Ey)P(x,y) → (Ey)(Ex) P(x,y)
- Switching the order of universals and existentials does change meaning:
  (Ax)(Ey) likes(x,y)
    Everyone likes someone
  (Ey)(Ax) likes(x,y)
    Someone is liked by everyone

Unification

Unification – takes two similar sentences and computes the substitution that makes them look the same

- Unify(α,β) = θ if subs(α,θ) = subs(β,θ)

p	q	θ
Knows(John,x)	Knows(John,Jane)	{x/Jane}}
Knows(John,x)	Knows(y,OJ)	{x/OJ,y/John}}
Knows(John,x)	Knows(y,Mother(y))	{y/John,x/Mother(John)}}
Knows(John,x)	Knows(x,OJ)	{fail}

Example knowledge base

- The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.
- Prove that Col. West is a criminal

Example knowledge base contd.

... it is a crime for an American to sell weapons to hostile nations:
American(x) ∧ Weapon(y) ∧ Sells(x,y,z) ∧ Hostile(z) ⇒ Criminal(x)
Nono … has some missiles, i.e., ∃x Owns(Nono,x) ∧ Missile(x):
Owns(Nono,M_1) and Missile(M_1)
… all of its missiles were sold to it by Colonel West
Missile(x) ∧ Owns(Nono,x) ⇒ Sells(West,x,Nono)
Missiles are weapons:
Missile(x) ⇒ Weapon(x)
An enemy of America counts as "hostile“:
Enemy(x,America) ⇒ Hostile(x)
West, who is American …
American(West)
The country Nono, an enemy of America …
Enemy(Nono,America)

Today

- Knowledge Representation and Reasoning
  - Ontological Engineering
  - Taxonomies
  - Representing Events
- In Class Assignment in SWI-Prolog

This Class

- How to use first order logic to represent the most important aspects of the real world – action, space, time, etc.

Ontological Engineering

- **Ontology** – A formal Conceptualization of the world
- We need general and flexible notations to describe complex domains
- We need to be able to reuse our ontological representations for new domains
- High level concepts exists in most domains
  - e.g. The concepts of *time*, *space* can be reused in the domain of Chemical Reactions or for representing a game of Football
- Representing abstract concepts in Logic is known as **Ontological Engineering**

Jobs in Ontological Engineering

Principal Taxonomist, Browse Experience
Amazon Corporate LLC - ★★★★☆ 9,625 reviews - Seattle, WA
Evangelize the role of Taxonomy and Ontology at Amazon. Influence across a broad range of engineers, designers, business owners, and stakeholders....
30+ days ago

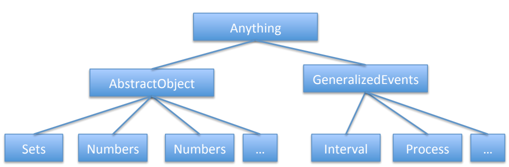
Knowledge Representation Researcher
Samsung - ★★★★☆ 1,820 reviews - Mountain View, CA
Samsung Research America (SRA) located in Mountain View, California is currently recruiting world-class R&D engineers who share our "Innovation through Passion"...
14 days ago - save job - more...

Data Modeler
Human Longevity - San Diego, CA
The Data Modeler will provide leadership insights into choosing data reference models, data mapping and terminology models for various content types such as...
30+ days ago

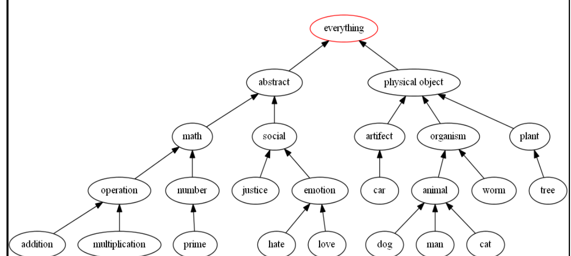
Upper Ontology

- General Framework of Concepts
- Reuse/inherit these generic concepts to model more specific concepts
- Similar to concepts of “inheritance” in programming languages
- FOL is one way of representing this knowledge

Example Upper Ontology



Wordnet Upper Ontology



Characteristics of a general purpose ontology

- It should be applicable to any special purpose domain
- Different area of knowledge must be unified
- e.g.A robot circuit repair system needs to know about:
  - electrical connectivity
  - physical layout
  - time
  - Sentences describing time must be able to work with sentences describing circuit design

Real World Ontologies

“Every Ontology is a treaty – a social agreement- among people with some common motive for sharing”

- Team of Trained Ontologists:
  - Cyc (Lenat and Guha 1990)
- Importing knowledge from existing databases
  - DBPedia (Bizer et. al. 2007)
- Parsing documents and Information Extraction
  - TextRuner (Banki and Etzioni 2008)
- Crowdsourced Commonsense Extraction
  - OpenMind (Singh et. al 2002, Chlovsky and Gil 2005)

Real World Ontologies

- Cyc (Lenat and Guha 1990)
- Decades worth of commonsense knowledge that is hand coded by humans
  - "An AI with 30 Years' Worth of Knowledge Finally Goes to Work"
  - "Cyc has been given many thousands of facts, including lots of information that you wouldn't find in an encyclopedia because it seems self-evident. It knows, for example, that that Sir Isaac Newton is a famous historical figure who is no longer alive. But more important, Cyc also understands that if you let go of an apple it will fall to the ground; that an apple is not bigger than a person; and that a person cannot throw an apple into space."

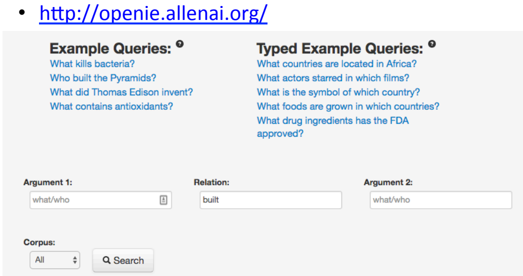
Real World Ontologies

- DBPedia (Bizer et. al. 2007)

DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web. DBpedia allows you to ask sophisticated queries against Wikipedia, and to link the different data sets on the Web to Wikipedia data. They hope that this work will make it easier for the huge amount of information in Wikipedia to be used in some new interesting ways. Furthermore, it might inspire new mechanisms for navigating, linking, and improving the encyclopedia itself.

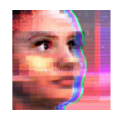
Real World Ontologies

- TextRuner (Banko and Etzioni 2008)
- http://openie.allenai.org/



Real World Ontologies

- OpenMind (Singh et. al 2002, Chklovsky and Gil 2005)
- Amateurs and volunteers proposed information about the world.
- Cons – Tay.ai – The Microsoft bot that learned what people told it
- It became controversial very quickly, based on what people were teaching it



CREATING ONTOLOGIES

Categories and Objects

- Why is categorization important?
  - It helps with faster retrieval, human information understanding, disambiguation, amongst other things
- Categories organize knowledgebases through inheritence
  - e.g. cars are a vehicle, Camry is a car
  - Camry inherits properties of vehicle
- Subclass relations organize categories into a taxonomy

is-a relation

- All horses are animals
  - Ax horse(x) → animal(x)
  - isa(horse, animal)
- NaOH is a chemical
  - Ax naoh(x) → chemical(x)
  - isa(naoh, chemical)

Categories and Objects

- Disjoint Relations: Two or more categories are disjoint if they have no members in common
  - e.g. a player can be part of only one team at a time
- Exhaustive Decomposition: All the possible ways in which a given category can be decomposed
  - e.g. North America -> USA, Canada, Mexico

Categories and Objects: PartOf

- PartOf relation (or Part_of etc.) is one way of capturing the *partonymic* or *meronymic* relationship between objects

Part-of Relations

Types of Part-Of Relations (Girju 2002)

1. COMPONENT–INTEGRAL OBJECT
   - e.g kitchen–apartment
2. MEMBER–COLLECTION
   - e.g. tree-forest, player-team
3. PORTION–MASS
   - e.g. slice-pie
4. STUFF–OBJECT
   - e.g. alcohol-wine
5. FEATURE–ACTIVITY
   - e.g. paying-shopping
6. PLACE–AREA.
   - e.g. Seattle-Washington

Properties of Part-of Relations

Part of Relations are Transitive

part_of(x,y) ^ part_of(y,z) **→** *part_of(x,z)*

Part of Relations are reflexive

part_of(x,x)

Measurements

- Scientific as well as commonsense theories have numerical information such as height, mass, weight, cost, etc.
  - e.g.
    - length(L_1) = inches(1.5) = centimeters(3.81)
    - centimeters(2.54 * d) = inches(d)
    - list_price(basketball) = dollar(19)

Objects: Things and Stuff

- Butter and Pencils
  - Butter cannot be cut in half, Pencil can
  - The former is a mass noun and the latter is a count noun
- Intrinsic Properties
  - These properties belong to the substance of the object, rather than the object as a whole. e.g. butter vs amount of butter
  - e.g. tasty(x) ^ butter(x)
  - (The butter is tasty)
- Extrinsic Properties
  - Weight, shape length etc, belong to the object as a whole. These properties do not remain after the object has been subdivided
  - e.g. size_of(x, small) ^ stick_of(x, y) ^ butter(y)
  - (the small stick of butter)

Events

Continuous Events

- Fluent – A condition that can change over time

Discrete Events

- Time Intervals

Event Representation

What are the different ways of representing events?

- Situation Calculus – Representing situations as predicates
- Event Calculus – Representing points of time rather than situations as predicates

Situation Calculus

- Shankar is traveling to DC from SFO
  - *traveling(Shankar, SFO, DC)*
  - *traveler(Shankar) ^ origin(SFO) ^ destination(DC)*
- Limitations:
  - No temporal information is captured
  - Process is not described
  - How to describe the event of "*traveling*"

Event Calculus

- Reification:
  - Process of making a concept more concrete.
  - Turning a predicate into an object
- Shankar is traveling to DC from SFO
  - *traveling(**e1**, Shankar, SFO, DC)*
  - *traveler(**e1**, Shankar) ^ origin(**e1**, SFO) ^ destination(**e1**, DC)*
- Now we can describe the journey
  - *long(**e1**)*
  - *delayed(**e1**)*

Event Calculus

T(f, t)	Fluent f is true for time t
Happens(e, i)	Event e happens over time interval i
Initiates(e, f, t)	Event e causes fluent f to start to hold at time t
Terminates(e, f, t)	Event e causes fluent f to start to hold at time t
Clipped(f, i)	Fluent f ceases to be true at some point during time interval i
Restored(f, i)	Fluent f becomes true sometime during time interval i

Events: Time Intervals

- Moments
  - Intervals that happen in an instant
  - Time = 0
  - *event(i) ^ duration(i, j) ^ seconds(j, 0)*
- Intervals
  - Have a duration
  - Have a start and end time

Date and Time

Interval(i) => Duration(i) = Time(end(i)) – Time(beginning(i))
Time(Begin(AD1900)) = Seconds(0)
Time(Begin(AD2001)) = Seconds(3187324800)
Time(Begin(AD2001)) = Seconds(3218860800)
Duration(AD2001) = Seconds(31536000)

Date(Begin(AD2001)) = Date(0, 0, 0, 1, Jan, 2001)
Date(0, 20, 21, 24, 1, 1995) = Seconds(3000000000)

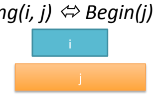
Event Intervals

Meet(i,j) ⇔ End(i) = Begin(j)

Before(i,j) ⇔ End(i) < Begin(j)
After(j, i) ⇔ Before(i,j)

Event Intervals

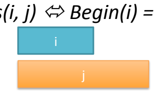
During(i, j) ⇔ Begin(j) < Begin(i) < End(i) < End(j)



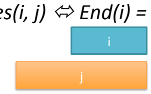
Overlap(i, j) ⇔ Begin(i) <Begin(j) < End(i) < End(j)

Event Intervals

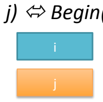
Begins(i, j) ⇔ Begin(i) = Begin(j)



Finishes(i, j) ⇔ End(i) = End(j)



Equals(i, j) ⇔ Begin(i) = Begin(j) ^ End(i) = End(j)

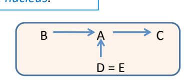


Real World Application

What if we can develop a system that can cure cancer after reading millions of research papers?

XPD appears to be degraded in wild-type embryos
between
prophase and metaphase of the first cell division
after
the onset of zygotic gene expression,
which coincides with
a redistribution of CDK7 from the cytoplasm to the nucleus.

A
between
B and C
after
D
which coincides with
E.



Time and Causality

We can capture knowledge from written language in real time using logic based axioms

	Causal Axioms
1	**in**-in(e4,x2) & **response**-nn(x2) & **to**-in(x2,x3) → CAUSES(x3,e4) & BEFORE(x3,e4)
2	**result**-vb(e0,x0) & **in**-in(e0,x8) →CAUSES(x0,x8) & BEFORE(x0,x8)
3	**lead**-vb(e1,x1) & **to**-in(e1,x3) → CAUSES(x1,x3) & BEFORE(x1,x3)
4	**in**-in(e4,x2) & **order**(x2) & **to**-in(x2,x3) → CAUSES(e4,x3) & BEFORE(e4,x3)
5	**upon**(e14,x15) → CAUSES(x15,e14) & BEFORE(x15,e14)
	Time based Axioms
6	**start**-vb'(e1,x1,e2) → BEGINS(x1,e2)
7	**progression**-nn(x1) & **into**-in(x1,x2) → BEGINS(x1,x2)
8	**onset**-nn(x1) & **of**-in(x1,x2) → BEGINS(x1,x2)

Inference in FOL

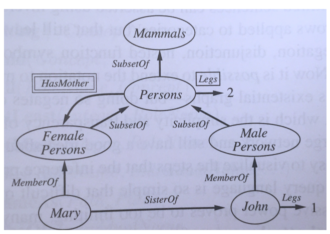
exposure-nn(x2) & …& **result-vb'(e0,x2) & in-in(e0,x8)** & …& activation-nn(x8) & …

BEFORE(x2,x8) & CAUSES(x2,x8)

Final Logical Form: exposure-nn(x2) & …& BEFORE(x2,x8) & CAUSES(x2,x8) & …& activation-nn(x8) & …

Visualizing Ontologies

- Tools you can use:
  - graphviz



Limitation of FOL

- One drawback of FOL is that it is not able to capture uncertainty or exceptions to the norm
  - e.g. most tomatoes are red, but not all are red
  - More in the next section of this class (Statistics and Uncertainty)