CS 181: Natural Language Processing

Lecture 15: Semantic Representations & Deduction

Kim Bruce
Pomona College
Spring 2008

Disclaimer: Slide contents borrowed from many sources on web!

Deduction

- Many formal systems of deduction.
- Deduction system consists of a set of logical axioms, Λ, and a set of rules of inference for deducing a new formula from old ones.
- Write $\Gamma \vdash \varphi$ iff there is a sequence $\varphi_0, \ldots, \varphi_n = \varphi$ such that each φ_i is a logical axiom, is in Γ, or follows from previous members of the sequence using a deduction rule.

Semantics & Deduction

- Write $\Gamma \models \phi$ iff for every model M, if M satisfies every $\Upsilon \in \Gamma$, then $M \models \phi$.
- Soundness Theorem: If $\Gamma \vdash \phi$ then $\Gamma \models \phi$
- Completeness Theorem: If $\Gamma \models \phi$ then $\Gamma \vdash \phi$
- Want both to hold so can use deduction in place of satisfaction.

Definitions

- Need to define substitution of terms for free variables.
- Define φ_x^t as follows:
  - If φ is atomic, obtain φ_x^t by replacing all occurrences of x by t.
  - $(\neg\varphi)_x^t = \neg(\varphi_x^t)$
  - $(\varphi\wedge\psi)_x^t = \varphi_x^t \wedge \psi_x^t$, and same for ∨, ⇒
  - $(\forall y.\varphi)_x^t = \begin{cases} \forall y.\varphi, \text{ if } x = y \\ \forall y.(\varphi_x^t), \text{ if } x \neq y \end{cases}$
  - ... and same for ∃

Definitions

- t is substitutable for x in ϕ iff x does not get captured by a quantifier in ϕ
- A tautology of FOL is a formula obtainable of propositional logic by replacing each sentence symbol by a formula of FOL
  - $A \rightarrow B \rightarrow A$ is tautology of propositional logic
  - $\exists x.\forall y.D(x,y) \rightarrow \forall x.E(x) \rightarrow \exists x.\forall y.D(x,y)$ is a tautology of FOL.

Deductive System

- Logical Axioms:
  - All tautologies
  - $\forall x.\ \varphi \rightarrow \varphi_x^t$, where t is *substitutable* for x in φ
  - $\forall x.(\varphi \rightarrow \psi) \rightarrow (\forall x.\varphi \rightarrow \forall x.\ \psi)$
  - $\varphi \rightarrow \forall x.\varphi$, if x is not free in φ.
- Rule:
  - Modus ponens: From φ, $\varphi \rightarrow \psi$, deduce ψ.

Deductive Systems

- Easy for mathematicians, not so great for computers.
- Forward chaining may get unnecessary results.
- Backward chaining may help direct search

Deductive Systems

- Finding contradictions easier:
  - To show, $\varphi \vdash \psi$, show $\varphi\wedge\neg\psi$ yields a contradiction.
  - Resolution theorem proving: Write formulas in canonical form, use "resolution" (which involves unification) to deduce new formulas.

Properties of FOL

- Axioms and proofs must be decidable in any useful system.
- Let Γ be consistent &decidable set of hypotheses. Then $P = \{ \phi \mid \Gamma \vdash \phi \}$ is effectively enumerable, but not decidable.
- By contrast, set of statements of FOL true of natural numbers is *not* effectively enumerable.

Multi-Sorted FOL

- Restrict quantifiers to particular domains.
  - $\forall$x:Student. $\exists$y: Chair. isi(x,y)
- Definable if have unary predicates:
  - $\forall x{:}Student.\ \phi \equiv \forall x.(Student(x) \Rightarrow \phi(x))$
  - $\exists y{:}\ Chair.\ \phi \equiv \exists y.(Chair(y) \wedge \phi)$
- Can be more important with generalized quantifiers.

Expressiveness: Extending the Reach

- Not obvious how get expressiveness needed.
- Going beyond FOL lose important properties in deduction systems.

Categories

- Predicates: Child(x)
- Tallest(y, Child) -- *not legal*
- Reify, by making relation an object
  - Now more complicated.
  - Need elt of relation: isA(x,Child)
  - Subset: contains(Child, Person) -- rules

Events

- Can say more about events if reify into an object.
- Can extract info from event about thematic roles.
- *Alternatives involve using many-placed predicates and adding coherence conditions.*
- *Alternatively add lots of quantifiers, but then hard to fit together statements.*

Example

- I threw the ball to Sam
- ∃w.ISA(w,Throwing) ∧ Thrower(w,Speaker) ∧Thrown(w,ball) ∧ Receiver(w,Sam)
- Add as many roles as like

Time

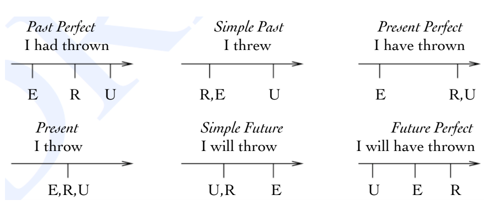
- Tenses convey temporal info and may also refer to them explicitly.
- Differences:
  - I threw the ball
  - I am throwing the ball
  - I will throw the ball
  - I have thrown the ball
  - I had thrown the ball

Explicit Representation

- I threw the ball
  - ∃i,e,w. ISA(w,Throwing) ∧ Thrown(w,ball) ∧ IntervalOf(w,i) ∧ EndPt(i,e) ∧ Precedes(e,Now)
- I am throwing the ball
  - ∃i,e,w. ISA(w,Throwing) ∧ Thrown(w,ball) ∧ IntervalOf(w,i) ∧ MemberOf(i,Now)
- I will throw the ball
  - ∃i,b,w. ISA(w,Throwing) ∧ Thrown(w,ball) ∧ IntervalOf(w,i) ∧ StartPt(i,b) ∧ Precedes(Now,b)

Reichenbach

- Keep track of event time, utterance time, reference time:



- Time can also be expressed with modal operators.

Modal Operators

- I believe the basketball team will win.
- The basketball team might win.
- The basketball team must win.
- It is possible that the basketball team will win.
- It is necessary that the basketball team will win.

How to Make Sense of Modals

- Try event-based approach
  - I believe that the basketball team will win.
  - ∃u,v.(IsA(u,believing) ∧ IsA(v,winning) ∧ Believer(u,Speaker) ∧ BelievedProp(u,v) ∧ Winner(v, the basketball team) ∧ ...)
- Implies there was a winning, v.

Modal Operators

- Extend FOL to add modal operators:
  - <I believe>(the basketball team will win)
  - <Necessary>(the basketball team will win)
  - <Possibility>(the basketball team will win)
- Interpret modal operators as applying over “possible worlds”.

Referential Transparency

- RT: It is possible to replace equals by equals without changing truth values:
  - (3 * 7) - 2 = 21 - 2
- Fails for modal operators:
  - I know batman is Bruce Wayne.
  - I know batman is batman.
  - I know that the top student will be valedictorian.
  - I know that Liz Adams will be valedictorian

Problems in Translation

- Many pitfalls in translation
  - Say that again and I’ll be angry!
  - If you are interested, there is a good colloquium speaker on Thursday.

Alternatives to FOL

- Slot-filler representations:
  - Semantic networks
  - Frames

Believing	
Believer:	Speaker
Believed:	Winning
	Winner: BB team

Semantic Web

W3C Semantic Web

“I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A ‘Semantic Web’, which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The ‘intelligent agents’ people have touted for ages will finally materialize.”

—Tim Berners-Lee, 1999

What's the Problem

- Typical web page has rendering info and links
- Semantic content accessible to humans, but not so easy for computers.
- Can mark up with semantic tags
  - address, phone number, papers, ...
  - but content still inaccessible

Solution?

- Add semantics by
  - Obtaining external agreement on meaning of annotations. E.g., XML standards within communities on meanings.
    - Problems: inflexible and limited in expressiveness
  - Use ontologies to specify meaning of annotations
    - Provide ways of building new terms from old
    - Formally specify meaning
    - Specify relations between terms in multiple ontologies

Ontology in CS

- Specific vocabulary to describe an area
- Explicit assumption on intended meaning of vocabulary
- Want it to be formal and machine manipulable.

Ontology Languages

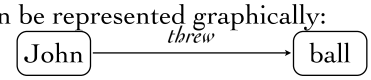
- Graphical:
  - Semantic networks,
  - UML
  - RDF (Resource Description Framework)
- Logic-based:
  - Description logics (OWL)
  - Rules
  - FOL
  - Higher-order logics
  - Non-classical logics

Ontology Languages

- Many based on OO concepts
  - Objects / instances / individuals
  - Types / classes / concepts
  - Relations / properties / roles

RDF

- Statements: <subject, predicate, object>
  - <John, threw, ball>
- Can be represented graphically:



- Statements represent properties of resources.
- Resources are objects that can be pointed to by a URI (properties too!)

RDF

- Objects linked together by properties:
  - <ball, isOwnedBy, Mary>
- Everything identified by URI -- not easily readable by humans.

RDF Schema

- Provides meaning for types and subclass relations.
- Define vocabulary and relations with other vocabulary -- specifies meaning
- Examples:
  - <Person, type, Class>
  - <threw, type, Property>
  - <Child, subClassOf, Person>
  - <John, type, Child>
  - <threw, domain, Person> *similarly for range*

RDF

- Problems:
  - No localized range and domain constraints
  - No existence or cardinality constraints
  - No way to specify transitive, inverses, or symmetric.
- Desirable features
  - Extends existing web standards (XML, RDF)
  - Easy to understand and use
  - Formally specified
  - Expressive
  - Automated reasoning possible

OWL

- Based on description logics
  - Well-defined semantics
  - Formal properties well understood (complexity, decidability)
  - Reasoning algorithms
  - Optimized implemented systems

OWL Class Constructors

- IntersectionOf, UnionOf, ComplementOf, oneOf, allValuesFrom, someValuesFrom, maxCardinality, minCardinality
- Example:
  - $\exists$hasAge.nonNegativeInteger
  - represents $\{x \mid \exists y.(x \text{ hasAge } y) \wedge y \in \text{nonNegativeInteger}\}$
- Define constraints on classes using above relations.

Computational Semantics

Semantic Analysis

- From representation to analysis.
- Syntax-driven semantic analysis.
- From meaning of words to meaning of phrases and sentences.
- Assume given legal parse tree
- Represent meaning of sentence in isolation.

Any Questions?