Lecture 12 of 42

First-Order Logic: Syntax and Semantics
Discussion: FOL Sentences in English

Wednesday, 20 September 2006

William H. Hsu
Department of Computing and Information Sciences, KSU

KSOL course page: http://snipurl.com/v9v3
Course web site: http://www.kddresearch.org/Courses/Fall-2006/CIS730
Instructor home page: http://www.cis.ksu.edu/~bhsu

Reading for Next Class:
Section 8.1 – 8.2, p. 240 – 253, Russell & Norvig 2nd edition

Lecture Outline

- Reading for Next Class: Section 8.1 – 8.2, R&N 2e
- Recommended : Nilsson and Genesereth (Chapter 5 online)
- Next Week's: Chapter 8 & first half of Chapter 9, R&N
- Today
  - Syntax of first-order predicate calculus (FOPC, *aka* "first-order logic")
  - Semantics
  - Role of automated deduction in AI
- This Week
  - Monday: Propositional Resolution, Soundness, Completness
  - Today: First-order logic (FOL): predicates, functions, quantifiers
  - Friday: Knowledge Engineering (KE) and theorem proving
- Coming Soon
  - Next week: Resolution, constraint logic, Prolog
  - Week of 04 Oct 2006: knowledge representation, ontologies

Logical Agents: Review

Logical agents apply inference to a knowledge base
to derive new information and make decisions

Basic concepts of logic:
- syntax: formal structure of sentences
- semantics: truth of sentences wrt models
- entailment: necessary truth of one sentence given another
- inference: deriving sentences from other sentences
- soundess: derivations produce only entailed sentences
- completeness: derivations can produce all entailed sentences

Wumpus world requires the ability to represent partial and negated information, reason by cases, etc.

Propositional logic suffices for some of these tasks

Adapted from slides by S. Russell, UC Berkeley

Predicate Logic and FOL Road Map

- **Predicate Logic**
  - **Enriching language**
    - **Predicates**
    - **Functions**
  - **Syntax and semantics of predicate logic**
- **First-Order Logic (FOL, FOPC)**
  - **Need for quantifiers**
  - **Relation to (unquantified) predicate logic**
  - **Syntax and semantics of FOL**
- **Fun with Sentences**
- **Wumpus World in FOL**

Syntax of FOL: Basic Elements

Constants	$KingJohn,\ 2,\ UCB, \ldots$
Predicates	$Brother,\ >, \ldots$
Functions	$Sqrt,\ LeftLegOf, \ldots$
Variables	$x,\ y,\ a,\ b, \ldots$
Connectives	$\wedge\ \vee\ \neg\ \Rightarrow\ \Leftrightarrow$
Equality	$=$
Quantifiers	$\forall\ \exists$

Adapted from slides by
S. Russell, UC Berkeley

Fun with Sentences: Family Feud

- Brothers are Siblings
  - ∀ *x*, *y* . ***Brother*** (*x*, *y*) ⇔ ***Sibling*** (*x*, *y*)
- Siblings (i.e., Sibling Relationships) are Reflexive
  - ∀ *x*, *y* . ***Sibling*** (*x*, *y*) ⇔ ***Sibling*** (*y*, *x*)
- One's Mother is One's Female Parent
  - ∀ *x*, *y* . ***Mother*** (*x*, *y*) ⇔ ***Female*** (*x*) ∧ ***Parent*** (*x*, *y*)
- A *First Cousin* Is A Child of A Parent's Sibling
  - ∀ *x*, *y* . ***First-Cousin*** (*x*, *y*) ⇔
    ∃ *p*, *ps* . ***Parent*** (*p*, *x*) ∧ ***Sibling*** (*p*, *ps*) ∧ ***Parent*** (*ps*, *y*)

Adapted from slides by
S. Russell, UC Berkeley

Exercise [1]: First-Order Logic Sentences

- **"Every Dog Chases Its Own Tail"**
  - ∀ *d* . Chases (*d*, *tail-of* (d))
  - Alternative Statement: ∀ *d* . ∃ *t* . *Tail-Of* (*t*, *d*) ∧ *Chases* (*d*, *t*)
  - Prefigures concept of Skolemization (Skolem variables / functions)
- **"Every Dog Chases Its Own (Unique) Tail"**
  - $\forall d \,.\, \exists^1 t \,.\, \text{Tail-Of}(t, d) \wedge \text{Chases}(d, t) \equiv$
    $\forall d \,.\, \exists t \,.\, \text{Tail-Of}(t, d) \wedge \text{Chases}(d, t) \wedge [\forall t' \text{ Chases}(d, t') \Rightarrow t' = t]$
- **"Only The Wicked Flee when No One Pursueth"**
  - ∀ *x* . *Flees* (*x*) ∧ [¬∃ *y Pursues* (*y*, *x*)] ⇒ *Wicked* (*x*)
  - Alternative : ∀ *x* . [∃ *y* . *Flees* (*x*, *y*)] ∧ [¬∃ *z* . *Pursues* (*z*, *x*)] ⇒ *Wicked* (*x*)
- **Offline Exercise: What Is An *n*th Cousin, *m* Times Removed?**

Exercise [2]: First-Order Logic Sentences

Validity and Satisfiability

A sentence is valid if it is true in all models
e.g., $A \lor \neg A$, $A \Rightarrow A$, $(A \land (A \Rightarrow B)) \Rightarrow B$

Validity is connected to inference via the Deduction Theorem:
$KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

A sentence is satisfiable if it is true in some model
e.g., $A \lor B$, C

A sentence is unsatisfiable if it is true in no models
e.g., $A \land \neg A$

Satisfiability is connected to inference via the following:
$KB \models \alpha$ if and only if $(KB \land \neg\alpha)$ is unsatisfiable
i.e., prove α by *reductio ad absurdum*

FOL: Atomic Sentences (Atomic Well-Formed Formulae)

Atomic sentence = $predicate(term_1, \ldots, term_n)$
or $term_1 = term_2$

Term = $function(term_1, \ldots, term_n)$
or $constant$ or $variable$

E.g., $Brother(KingJohn, RichardTheLionheart)$
$> (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn))$

Adapted from slides by
S. Russell, UC Berkeley

FOL: Complex Sentences (Well-Formed Formulae)

Complex sentences are made from atomic sentences using connectives

$\neg S, \quad S_1 \wedge S_2, \quad S_1 \vee S_2, \quad S_1 \Rightarrow S_2, \quad S_1 \Leftrightarrow S_2$

E.g. $Sibling(KingJohn, Richard) \Rightarrow Sibling(Richard, KingJohn)$
$>(1,2) \vee \leq(1,2)$
$>(1,2) \wedge \neg>(1,2)$

Adapted from slides by
S. Russell, UC Berkeley

Truth in FOL

Sentences are true with respect to a model and an interpretation

Model contains objects and relations among them

Interpretation specifies referents for
constant symbols → objects
predicate symbols → relations
function symbols → functional relations

An atomic sentence $predicate(term_1, \ldots, term_n)$ is true
iff the objects referred to by $term_1, \ldots, term_n$
are in the relation referred to by *predicate*

Adapted from slides by
S. Russell, UC Berkeley

Taking Stock: FOL Inference

- **Previously: Logical Agents and Calculi**
- **FOL in Practice**
  - **Agent "toy" world: Wumpus World in FOL**
  - **Situation calculus**
  - **Frame problem and variants (see R&N sidebar)**
    - **Representational vs. inferential frame problems**
    - **Qualification problem: "what if?"**
    - **Ramification problem: "what else?" (side effects)**
  - **Successor-state axioms**
- **FOL Knowledge Bases**
- **FOL Inference**
  - **Proofs**
  - **Pattern-matching: unification**
  - **Theorem-proving as search**
    - **Generalized Modus Ponens (GMP)**
    - **Forward Chaining and Backward Chaining**

Automated Deduction (Chapters 8-10 R&N)

Sound inference: find α such that $KB \models \alpha$.
Proof process is a search, operators are inference rules.

E.g., Modus Ponens (MP)

$$\frac{\alpha, \quad \alpha \Rightarrow \beta}{\beta} \qquad \frac{At(Joe, UCB) \quad At(Joe, UCB) \Rightarrow OK(Joe)}{OK(Joe)}$$

E.g., And-Introduction (AI)

$$\frac{\alpha \quad \beta}{\alpha \wedge \beta} \qquad \frac{OK(Joe) \quad CSMajor(Joe)}{OK(Joe) \wedge CSMajor(Joe)}$$

E.g., Universal Elimination (UE)

$$\frac{\forall x \ \alpha}{\alpha\{x/\tau\}} \qquad \frac{\forall x \ At(x, UCB) \Rightarrow OK(x)}{At(Pat, UCB) \Rightarrow OK(Pat)}$$

τ must be a ground term (i.e., no variables)

Adapted from slides by S. Russell, UC Berkeley

Summary Points

- Applications of Knowledge Bases (KBs) and Inference Systems
- "Industrial Strength" KBs
  - Building KBs
  - Components
    - Ontologies
    - Fact and rule bases
    - Knowledge Engineering (KE) and protocol analysis
    - Inductive Logic Programming (ILP) and other machine learning techniques
  - Using KBs
- Systems of Sequent Rules: GMP/AI/UE, Resolution
- Methodology of Inference
  - Inference as search
  - Forward and backward chaining
  - Fan-in, fan-out

Terminology

- Logical Frameworks
  - Knowledge Bases (KB)
  - Logic in general: representation languages, syntax, semantics
  - Propositional logic
  - First-order logic (FOL, FOPC)
  - Model theory, domain theory: possible worlds semantics, entailment
- Normal Forms
  - Conjunctive Normal Form (CNF)
  - Disjunctive Normal Form (DNF)
  - Horn Form
- Proof Theory and Inference Systems
  - Sequent calculi: rules of proof theory
  - Derivability or provability
  - Properties
    - Soundness (derivability implies entailment)
    - Completeness (entailment implies derivability)