INTRO TO FIRST-ORDER LOGIC:
SYNTAX AND SEMANTICS

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/CIS730
Instructor home page: http://www.cis.ksu.edu/~bhsu

Reading for Next Class:
Section 8.3 – 8.4, p. 253 - 266, Russell & Norvig 2nd edition
Handout, Nilsson & Genesereth, Logical Foundations of Artificial Intelligence

Lecture Outline
- Reading for Next Class: 8.3-8.4 (p. 253-266), 9.1 (p. 272-274), R&N 2nd
- Last Class: Propositional Logic, Sections 7.5-7.7 (p. 211-232), R&N 2nd
  - Properties of sentences (and sets of sentences, aka knowledge bases)
    - entailment
    - provability/derivability
    - validity: truth in all models (aka tautological truth)
    - satisfiability: truth in some models
  - Properties of proof rules
    - soundness: KB ⊢ α ⇒ KB ⊨ α (can prove only true sentences)
    - completeness: KB ⊨ α ⇒ KB ⊢ α (can prove all true sentences)
- Still to Cover in Chapter 7: Resolution, Conjunctive Normal Form (CNF)
- Today: Intro to First-Order Logic, Sections 8.1-8.2 (p. 240-253), R&N 2nd
  - Elements of logic: ontology and epistemology
  - Resolution theorem proving
    - First-order predicate calculus (FOPC) aka first order logic (FOL)
  - Coming Week: Propositional and First-Order Logic (Ch. 8 – 9)
Chapter 7
Concluded

- Knowledge-based agents
- Wumpus world
- Logic in general—models and entailment
- Propositional (Boolean) logic
- Equivalence, validity, satisfiability
- Inference rules and theorem proving
  - forward chaining
  - backward chaining
  - resolution

Inference: Review

\[ KB \vdash_i \alpha = \text{sentence } \alpha \text{ can be derived from } KB \text{ by procedure } i \]

Consequences of \( KB \) are a haystack; \( \alpha \) is a needle.
Entailment = needle in haystack; inference = finding it

Soundness: \( i \) is sound if whenever \( KB \vdash_i \alpha \), it is also true that \( KB \models \alpha \)

Completeness: \( i \) is complete if whenever \( KB \models \alpha \), it is also true that \( KB \vdash_i \alpha \)

Preview: we will define a logic (first-order logic) which is expressive enough to say almost anything of interest, and for which there exists a sound and complete inference procedure.

That is, the procedure will answer any question whose answer follows from what is known by the \( KB \).

**Validity and Satisfiability: Review**

A sentence is **valid** if it is true in all models,
e.g., \( \text{True}, \ A \lor \neg A, \ A \Rightarrow A, \ (A \land (A \Rightarrow B)) \Rightarrow B \)

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 \( \alpha \) by **reductio ad absurdum**

---

**Forward Chaining Example: Review**

\( n \): number of antecedents
(LHS conjuncts) still unmatched

Adapted from slides © 2004 S. Russell & P. Norvig. Reused with permission.
Backward Chaining Example: Review

Forward vs. Backward Chaining: Review

FC is data-driven, cf. automatic, unconscious processing, e.g., object recognition, routine decisions.

May do lots of work that is irrelevant to the goal.

BC is goal-driven, appropriate for problem-solving, e.g., Where are my keys? How do I get into a PhD program?

Complexity of BC can be much less than linear in size of KB.
Resolution [1]: Propositional Sequent Rule

Conjunctive Normal Form (CNF—universal)

\[
\text{conjunction of disjunctions of literals}
\]

E.g., \((A \lor \neg B) \land (B \lor \neg C \lor \neg D)\)

Resolution inference rule (for CNF): complete for propositional logic

\[
\ell_1 \lor \cdots \lor \ell_k, m_1 \lor \cdots \lor m_n
\]

\[
\ell_1 \lor \cdots \lor \ell_{i-1} \lor \ell_i+1 \lor \cdots \lor \ell_k \lor m_1 \lor \cdots \lor m_{j-1} \lor m_{j+1} \lor \cdots \lor m_n
\]

where \(\ell_i\) and \(m_j\) are complementary literals. E.g.,

\[
\frac{P_{1,2} \lor P_{2,2}, \neg P_{2,2}}{P_{1,3}}
\]

Resolution is sound and complete for propositional logic

Resolution [2]: Conversion to Conjunctive Normal Form (CNF)

\[
B_{1,1} \iff (P_{1,2} \lor P_{2,1})
\]

1. Eliminate \(\iff\), replacing \(\alpha \iff \beta\) with \((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)\).

\[
(B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1})
\]

2. Eliminate \(\Rightarrow\), replacing \(\alpha \Rightarrow \beta\) with \(\neg \alpha \lor \beta\).

\[
(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg (P_{1,2} \lor P_{2,1}) \lor B_{1,1})
\]

3. Move \(\neg\) inwards using de Morgan’s rules and double-negation:

\[
(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land ((\neg P_{1,2} \lor \neg P_{2,1}) \lor B_{1,1})
\]

4. Apply distributivity law (\(\lor\) over \(\land\)) and flatten:

\[
(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg P_{1,2} \lor B_{1,1}) \land (\neg P_{2,1} \lor B_{1,1})
\]

Based on slide © 2004 S. Russell & P. Norvig. Reused with permission.
Resolution [3]:
Algorithm

Proof by contradiction, i.e., show \( KB \land \neg \alpha \) unsatisfiable

```
function PL-RESOLUTION(KB, \alpha) returns true or false
  inputs: KB, the knowledge base, a sentence in propositional logic
          \alpha, the query, a sentence in propositional logic
  clauses — the set of clauses in the CNF representation of \( KB \land \neg \alpha \)
  new — {}
  loop do
    for each \( C_i, C_j \) in clauses do
      resolvents — PL-RESOLVE(\( C_i, C_j \))
      if resolvents contains the empty clause then return true
      new — new \cup resolvents
      if new \subseteq clauses then return false
    clauses — clauses \cup new
```

Resolution [4]:
Example

\[ KB = (B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})) \land \neg B_{1,1} \land \neg P_{1,2} \]

Based on slide © 2004 S. Russell & P. Norvig. Reused with permission.
Chapter 7: Summary

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
- souldess: 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.

Forward, backward chaining are linear-time, complete for Horn clauses
Resolution is complete for propositional logic
Propositional logic lacks expressive power

Chapter 8: Overview

- Why FOL?
- Syntax and semantics of FOL
- Fun with sentences
- Wumpus world in FOL

Propositional Logic: Pros and Cons

- Propositional logic is declarative: pieces of syntax correspond to facts
- Propositional logic allows partial/disjunctive/negated information (unlike most data structures and databases)
- Propositional logic is compositional: meaning of \( B_{1,1} \land P_{1,2} \) is derived from meaning of \( B_{1,1} \) and of \( P_{1,2} \)
- Meaning in propositional logic is context-independent (unlike natural language, where meaning depends on context)
- Propositional logic has very limited expressive power (unlike natural language)
  E.g., cannot say “pits cause breezes in adjacent squares” except by writing one sentence for each square

First-Order Logic (FOL)

Whereas propositional logic assumes world contains facts, first-order logic (like natural language) assumes the world contains

- **Objects**: people, houses, numbers, theories, Ronald McDonald, colors, baseball games, wars, centuries . . .
- **Relations**: red, round, bogus, prime, multistoried . . ., brother of, bigger than, inside, part of, has color, occurred after, owns, comes between, . . .
- **Functions**: father of, best friend, third inning of, one more than, end of . . .
### Logics in General: Ontological and Epistemic Aspects

<table>
<thead>
<tr>
<th>Language</th>
<th>Ontological Commitment</th>
<th>Epistemological Commitment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propositional logic</td>
<td>facts</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>First-order logic</td>
<td>facts, objects, relations</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>Temporal logic</td>
<td>facts, objects, relations, times</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>Probability theory</td>
<td>facts</td>
<td>degree of belief</td>
</tr>
<tr>
<td>Fuzzy logic</td>
<td>facts + degree of truth</td>
<td>known interval value</td>
</tr>
</tbody>
</table>

**Ontological commitment** – what entities, relationships, and facts exist in the world and can be reasoned about

**Epistemic commitment** – what agents can know about the world

---

### Syntax of FOL: Basic Elements

- **Constants**: KingJohn, 2, UCB, ...
- **Predicates**: Brother, >, ...
- **Functions**: Sqrt, LeftLegOf, ...
- **Variables**: x, y, a, b, ...
- **Connectives**: ∧, ∨, ¬, ⇒, ⇔
- **Equality**: =
- **Quantifiers**: ∀, ∃
Atomic Sentences

(aka Atoms, aka Atomic WFFs)

Atomic sentence = predicate(term₁, ..., termₙ)
or term₁ = term₂

Term = function(term₁, ..., termₙ)
or constant or variable

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

- Atomic sentence – smallest unit of a logic
  (aka “atom”, “atomic well-formed formula (atomic WFF)”)

Complex Sentences

Complex sentences are made from atomic sentences using connectives

¬S₁, S₁ ∧ S₂, S₁ ∨ S₂, S₁ ⊨ S₂, S₁ ↔ S₂

E.g., Sibling(KingJohn, Richard) ⇒ Sibling(Richard, KingJohn)
> (1, 2) ∨ ≤ (1, 2)
> (1, 2) ∧ ¬> (1, 2)
Truth in First-Order Logic

Sentences are true with respect to a model and an interpretation.

Model contains \( \geq 1 \) objects (domain elements) and relations among them.

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

An atomic sentence \( \text{predicate}(\text{term}_1, \ldots, \text{term}_n) \) is true if the objects referred to by \( \text{term}_1, \ldots, \text{term}_n \) are in the relation referred to by \( \text{predicate} \).

Models for FOL: Example

Models for FOL:  
Example

Consider the interpretation in which

- \textit{Richard} \rightarrow \text{Richard the Lionheart}
- \textit{John} \rightarrow \text{the evil King John}
- \textit{Brother} \rightarrow \text{the brotherhood relation}

Under this interpretation, \textit{Brother(Richard, John)} is true just in case Richard the Lionheart and the evil King John are in the brotherhood relation in the model.

Models for FOL: 
Lots!

Entailment in propositional logic can be computed by enumerating models

We can enumerate the FOL models for a given KB vocabulary:

For each number of domain elements \(n\) from 1 to \(\infty\)
  - For each \(k\)-ary predicate \(P_k\) in the vocabulary
    - For each possible \(k\)-ary relation on \(n\) objects
      - For each constant symbol \(C\) in the vocabulary
        - For each choice of referent for \(C\) from \(n\) objects...

Computing entailment by enumerating FOL models is not easy!
**Universal Quantification [1]: Definition**

\[ \forall \text{ (variables) (sentence)} \]

Everyone at Berkeley is smart:
\[ \forall x \; \text{At}(x, \text{Berkeley}) \Rightarrow \text{Smart}(x) \]

\[ \forall x \; P \text{ is true in a model } m \text{ iff } P \text{ is true with } x \text{ being each possible object in the model} \]

Roughly speaking, equivalent to the conjunction of instantiations of \( P \)

\[ (\text{At}(\text{KingJohn}, \text{Berkeley}) \Rightarrow \text{Smart}(\text{KingJohn})) \]
\[ \land (\text{At}(\text{Richard}, \text{Berkeley}) \Rightarrow \text{Smart}(\text{Richard})) \]
\[ \land (\text{At}(\text{Berkeley}, \text{Berkeley}) \Rightarrow \text{Smart}(\text{Berkeley})) \]
\[ \land \ldots \]

Based on slide © 2004 S. Russell & P. Norvig. Reused with permission.

---

**Universal Quantification [2]: Common Mistake to Avoid**

Typically, \( \Rightarrow \) is the main connective with \( \forall \)

Common mistake: using \( \land \) as the main connective with \( \forall \):

\[ \forall x \; \text{At}(x, \text{Berkeley}) \land \text{Smart}(x) \]

means “Everyone is at Berkeley and everyone is smart”
**Existential Quantification [1]: Definition**

\[ \exists \text{ (variables) (sentence)} \]

Someone at Stanford is smart:
\[ \exists x \ At(x, \text{Stanford}) \land \text{Smart}(x) \]

\[ \exists x \ P \] is true in a model \( m \) if \( P \) is true with \( x \) being some possible object in the model

Roughly speaking, equivalent to the disjunction of instantiations of \( P \)

\[ (At(\text{KingJohn}, \text{Stanford}) \land \text{Smart}(\text{KingJohn})) \]
\[ \lor (At(\text{Richard}, \text{Stanford}) \land \text{Smart}(\text{Richard})) \]
\[ \lor (At(\text{Stanford}, \text{Stanford}) \land \text{Smart}(\text{Stanford})) \]
\[ \lor \ldots \]

Based on slide © 2004 S. Russell & P. Norvig. Reused with permission.

---

**Existential Quantification [2]: Common Mistake to Avoid**

Typically, \( \land \) is the main connective with \( \exists \)

Common mistake: using \( \Rightarrow \) as the main connective with \( \exists \):

\[ \exists x \ At(x, \text{Stanford}) \Rightarrow \text{Smart}(x) \]

is true if there is anyone who is not at Stanford!

Based on slide © 2004 S. Russell & P. Norvig. Reused with permission.
Properties of Quantifiers

∀x ∀y is the same as ∀y ∀x (why??)
∃x ∃y is the same as ∃y ∃x (why??)
∃x ∀y is not the same as ∀y ∃x
∃x ∀y Loves(x, y)
"There is a person who loves everyone in the world"
∀y ∃x Loves(x, y)
"Everyone in the world is loved by at least one person"
Quantifier duality: each can be expressed using the other
∀x Likes(x, IceCream) ¬∃x ¬Likes(x, IceCream)

Fun With Sentences

Brothers are siblings
∀x, y Brother(x, y) ⇒ Sibling(x, y).

"Sibling" is symmetric
∀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 FirstCousin(x, y) ⇔ ∃p, ps Parent(p, x) ∧ Sibling(ps, p) ∧ Parent(ps, y)
**EQUALITY**

\[ \text{term}_1 = \text{term}_2 \] is true under a given interpretation if and only if \( \text{term}_1 \) and \( \text{term}_2 \) refer to the same object.

E.g., \( 1 = 2 \) and \( \forall x \times (\text{Sqrt}(x), \text{Sqrt}(x)) = x \) are satisfiable. \( 2 = 2 \) is valid.

E.g., definition of (full) \textit{Sibling} in terms of \textit{Parent}:

\[ \forall x, y \quad \text{Sibling}(x, y) \iff [-(x = y) \land \exists m, f \land -(m = f) \land \text{Parent}(m, x) \land \text{Parent}(f, x) \land \text{Parent}(m, y) \land \text{Parent}(f, y)] \]

*Interacting with FOL KBs*

Suppose a wumpus-world agent is using an FOL KB and perceives a smell and a breeze (but no glitter) at \( t = 5 \):

- \( \text{Tell}(KB, \text{Percept}([\text{Smell}, \text{Breeze}, \text{None}], 5)) \)
- \( \text{Ask}(KB, \exists a \quad \text{Action}(a, 5)) \)

I.e., does \( KB \) entail any particular actions at \( t = 5 \)?

Answer: Yes, \( \{a/\text{Shoot}\} \) ← substitution (binding list)

Given a sentence \( S \) and a substitution \( \sigma \),

\( S\sigma \) denotes the result of plugging \( \sigma \) into \( S \); e.g.,

\( S = \text{Smarter}(x, y) \)
\( \sigma = \{x/\text{Hillary}, y/\text{Bill}\} \)
\( S\sigma = \text{Smarter}(\text{Hillary}, \text{Bill}) \)

\( \text{Ask}(KB, S) \) returns some/all \( \sigma \) such that \( KB \models S\sigma \)
**Knowledge Base for Wumpus World**

"Perception"
\[ \forall b, g, t \text{ Percept}([\text{Smell}, b, g], t) \Rightarrow \text{Smell}(t) \]
\[ \forall s, b, t \text{ Percept}([s, b, \text{Glitter}], t) \Rightarrow \text{AtGold}(t) \]

Reflex: \[ \forall t \text{ AtGold}(t) \Rightarrow \text{Action(Grab, t)} \]

Reflex with internal state: do we have the gold already?
\[ \forall t \text{ AtGold}(t) \land \neg \text{Holding(Gold, t)} \Rightarrow \text{Action(Grab, t)} \]

\textit{Holding(Gold, t)} cannot be observed
\[ \Rightarrow \text{keeping track of change is essential} \]

---

**Deducing Hidden Properties**

Properties of locations:
\[ \forall x, t \text{ At(Agent, x, t) \land Smell(t) \Rightarrow Smelly(x)} \]
\[ \forall x, t \text{ At(Agent, x, t) \land Breeze(t) \Rightarrow Breezy(x)} \]

Squares are breezy near a pit:

Diagnostic rule—infer cause from effect
\[ \forall y \text{ Breezy(y) \Rightarrow } \exists x \text{ Pit(x) \land Adjacent(x, y)} \]

Causal rule—infer effect from cause
\[ \forall x, y \text{ Pit(x) \land Adjacent(x, y) \Rightarrow Breezy(y)} \]

Neither of these is complete—e.g., the causal rule doesn’t say whether squares far away from pits can be breezy.
**TERMINOLOGY**

- **First-Order Logic (FOL)** aka **First-Order Predicate Calculus (FOPC)**
  - **Components**
    - Semantics (meaning, denotation): objects, functions, relations
    - Syntax: constants, variables, terms, predicates
  - **Properties of sentences (and sets of sentences, aka knowledge bases)**
    - entailment
    - provability/derivability
    - validity: truth in all models (aka tautological truth)
    - satisfiability: truth in some models
  - **Properties of proof rules**
    - soundness: $\text{KB} \vdash \alpha \Rightarrow \text{KB} \models \alpha$ (can prove only true sentences)
    - completeness: $\text{KB} \models \alpha \Rightarrow \text{KB} \vdash \alpha$ (can prove all true sentences)

- **Conjunctive Normal Form (CNF)**
- **Universal Quantification** ("For All")
- **Existential Quantification** ("Exists")

**SUMMARY POINTS**

- **Last Class: Overview of Knowledge Representation (KR) and Logic**
  - Representations covered in this course, by ontology and epistemology
  - Propositional calculus (aka propositional logic)
    - Syntax and semantics
    - Relationship to Boolean algebra
    - Properties

- **Propositional Resolution**
- **Elements of Logics – Ontology, Epistemology**
- **Today: First-Order Logic (FOL) aka FOPC**
  - Components: syntax, semantics
  - Sentences: entailment vs. provability/derivability, validity vs. satisfiability
  - Soundness and completeness
  - Properties of proof rules
    - soundness: $\text{KB} \vdash \alpha \Rightarrow \text{KB} \models \alpha$ (can prove only true sentences)
    - completeness: $\text{KB} \models \alpha \Rightarrow \text{KB} \vdash \alpha$ (can prove all true sentences)

- **Next: First-Order Resolution**