



Lecture 11 of 42

Resolution, Forward/Backward Chaining Discussion: Expert Systems

Monday, 17 September 2007

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-2007/CIS730>

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

Reading for Next Class:

Section 7.5 – 7.7, p. 211 - 232, Russell & Norvig 2nd edition



Normal Forms: CNF, DNF, Horn

Other approaches to inference use syntactic operations on sentences, often expressed in standardized forms

Conjunctive Normal Form (CNF—universal)
conjunction of disjunctions of literals
clauses

E.g., $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

Disjunctive Normal Form (DNF—universal)
disjunction of conjunctions of literals
terms

E.g., $(A \wedge B) \vee (A \wedge \neg C) \vee (A \wedge \neg D) \vee (\neg B \wedge \neg C) \vee (\neg B \wedge \neg D)$

Horn Form (restricted)

conjunction of Horn clauses (clauses with ≤ 1 positive literal)

E.g., $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

Often written as set of implications:

$B \Rightarrow A$ and $(C \wedge D) \Rightarrow B$

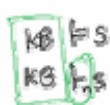
Precedent antecedent consequent

$B \Rightarrow A$

$A \vdash B$

$A_1 \vdash B, A_2 \vdash B, \dots, A_n \vdash B$

$A_1 \vee A_2 \vee \dots \vee A_n \vdash B$



entailment
probability/derivability
Property of KB + sentences
KB w/ pos. prec.

Soundness
Completeness

Satisfiability
Validity

Proof prec.
Sentences (KB) + semantics

$[\alpha]$ det poly

$L \in NP$ -complete

$\rightarrow L \in P \wedge \forall L' \in P L \leq P L'$

all-time

Adapted from slides by S. Russell, UC Berkeley



Ex
False
Quodlibet

Validity and Satisfiability

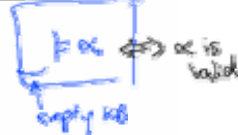
Q | ant
E | rat
D | consequentia

A sentence is **valid** if it is true in **all** models

e.g., $A \vee \neg A$, $\boxed{A \Rightarrow A} \Rightarrow (A \wedge (A \Rightarrow B)) \Rightarrow B$

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., $A \vee B$, C
 $\exists w \in [A]$

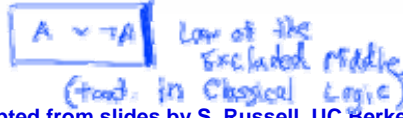
A sentence is **unsatisfiable** if it is true in **no** models

e.g., $A \wedge \neg A$
 $\neg \exists w \in [A] \Leftrightarrow \forall w. w \notin [A]$

Satisfiability is connected to inference via the following:

$KB \models \alpha$ if and only if $(KB \wedge \neg \alpha)$ is unsatisfiable

i.e., prove α by *reductio ad absurdum*

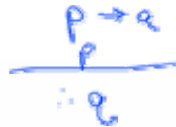


Adapted from slides by S. Russell, UC Berkeley



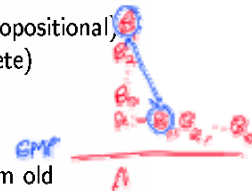
Proof Methods

Proof methods divide into (roughly) two kinds:



Model checking

- truth table enumeration (sound and complete for propositional)
- heuristic search in model space (sound but incomplete)
e.g., the GSAT algorithm (Ex. 6.15)



Application of inference rules

Legitimate (sound) generation of new sentences from old

Proof = a sequence of inference rule applications

Can use inference rules as operators in a standard search alg.



Adapted from slides by S. Russell, UC Berkeley



Inference (Sequent) Rules for Propositional Logic

$$\frac{A \vee B, \neg(A \vee B) \vee B}{\neg(A \vee B) \vee B} \quad \frac{\neg A \Rightarrow B}{\neg A \Rightarrow B}$$

Resolution (for CNF): complete for propositional logic

$$\frac{\alpha \vee \beta, \neg\beta \vee \gamma}{\alpha \vee \gamma}$$

$$\frac{\neg A \Rightarrow B \quad B \Rightarrow C}{\neg A \Rightarrow C}$$

$$\frac{A \Rightarrow B \quad B \Rightarrow C}{A \Rightarrow C}$$

Modus Ponens (for Horn Form): complete for Horn KBs

$$\frac{\alpha_1, \dots, \alpha_n \quad \alpha_1 \wedge \dots \wedge \alpha_n \Rightarrow \beta}{\beta}$$

Can be used with forward chaining or backward chaining

Adapted from slides by S. Russell, UC Berkeley



Logical Agents: Taking Stock

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
- soundness: derivations produce only entailed sentences
- completeness: derivations can produce all entailed sentences

KB
Proof rules

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



The Road Ahead: Predicate Logic and FOL

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

Tim is Awake Prop.
Awake (Tim). Pred.

$P(x_1, x_2, \dots, x_n)$

Adapted from slides by S. Russell, UC Berkeley



Syntax of FOL: Basic Elements

- ✓ Constants $[KingJohn], 2, UCB, \dots$
- Predicates $[Brother], >, \dots$
- Functions $[Sqr], LeftLegOf, \dots$
- ✓ Variables x, y, a, b, \dots
- ✓ Connectives $\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$
- Equality $=$
- Quantifiers \forall, \exists

Objects
Relation: a set of tuples for which the pred holds

$[f] = \{x \mid f(x) = y\}$

$\forall x. P(x)$
 $\Leftrightarrow P(x_1) \wedge P(x_2) \wedge \dots$
 $\exists x. P(x)$
 $\Leftrightarrow P(x_1) \vee P(x_2) \vee \dots$

Adapted from slides by S. Russell, UC Berkeley



FOL: Atomic Sentences (Atomic Well-Formed Formulae)

Atomic sentence = $predicate(term_1, \dots, term_n)$
or $term_1 = term_2$] 0-ary

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

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



Adapted from slides by S. Russell, UC Berkeley



Summary Points

- **Logical Agents Overview (Last Time)**
 - * Knowledge Bases (KB) and KB agents
 - * Motivating example: Wumpus World
 - * Logic in general
 - * Syntax of propositional calculus
- **Propositional and First-Order Calculi (Today)**
 - * Propositional calculus (concluded)
 - ⇒ Normal forms
 - ⇒ Inference (aka sequent) rules
 - * Production systems
 - * Predicate logic without quantifiers
 - * Introduction to First-Order Logic (FOL)
 - ⇒ Examples
 - ⇒ Inference rules (sketch)
- **Next Week: FOL Review, Intro to Resolution**



Fun with Sentences: Family Feud

- Brothers are Siblings
 - * $\forall x, y . \text{Brother}(x, y) \Leftrightarrow \text{Sibling}(x, y)$
- Siblings (i.e., Sibling Relationships) are Reflexive
 - * $\forall x, y . \text{Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x)$
- One's Mother is One's Female Parent
 - * $\forall x, y . \text{Mother}(x, y) \Leftrightarrow \text{Female}(x) \wedge \text{Parent}(x, y)$
- A First Cousin Is A Child of A Parent's Sibling
 - * $\forall x, y . \text{First-Cousin}(x, y) \Leftrightarrow$
 $\exists p, ps . \text{Parent}(p, x) \wedge \text{Sibling}(p, ps) \wedge \text{Parent}(ps, y)$

Adapted from slides by S. Russell, UC Berkeley



Jigsaw Exercise [1]: First-Order Logic Sentences

- "Every Dog Chases Its Own Tail"
 - * $\forall d . \text{Chases}(d, \text{tail-of}(d))$
 - * Alternative Statement: $\forall d . \exists t . \text{Tail-Of}(t, d) \wedge \text{Chases}(d, t)$
 - * Prefigures concept of Skolemization (Skolem vars / 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"
 - * $\forall x . \text{Flees}(x) \wedge [\neg \exists y \text{Pursues}(y, x)] \Rightarrow \text{Wicked}(x)$
 - * Alternative : $\forall x . [\exists y . \text{Flees}(x, y)] \wedge [\neg \exists z . \text{Pursues}(z, x)] \Rightarrow \text{Wicked}(x)$
- Offline Exercise: What Is An *n*th Cousin, *m* Times Removed?





Jigsaw Exercise [2]: First-Order Logic Sentences



Validity and Satisfiability

A sentence is **valid** if it is true in **all** models

e.g., $A \vee \neg A$, $A \Rightarrow A$, $(A \wedge (A \Rightarrow B)) \Rightarrow B$

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., $A \vee B$, C

A sentence is **unsatisfiable** if it is true in **no** models

e.g., $A \wedge \neg A$

Satisfiability is connected to inference via the following:

$KB \models \alpha$ if and only if $(KB \wedge \neg \alpha)$ is unsatisfiable

i.e., prove α by *reductio ad absurdum*





Proof Methods

Proof methods divide into (roughly) two kinds:

Model checking

truth table enumeration (sound and complete for propositional)
heuristic search in model space (sound but incomplete)
e.g., the GSAT algorithm (Ex. 6.15)

Application of inference rules

Legitimate (sound) generation of new sentences from old

Proof = a sequence of inference rule applications

Can use inference rules as operators in a standard search alg.

Adapted from slides by S. Russell, UC Berkeley



Logical Agents: Taking Stock

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
- soundness: 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



FOL: Atomic Sentences (Atomic Well-Formed Formulae)

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

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

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

Adapted from slides by S. Russell, UC Berkeley

CIS 530 / 730: Artificial Intelligence

Monday, 17 Sep 2007

Computing & Information Sciences
Kansas State University



FOL: Complex Sentences (Well-Formed Formulae)

Complex sentences are made from atomic sentences using connectives

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

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

CIS 530 / 730: Artificial Intelligence

Monday, 17 Sep 2007

Computing & Information Sciences
Kansas State University



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

Adapted from slides by S. Russell, UC Berkeley



Terminology

● Logical Frameworks

- * Knowledge Bases (KB)
- * Logic in general: representation languages, syntax, semantics
- * Propositional logic
- * First-order logic (FOL, FOPL)
- * 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)

