Title: First-Order Logic

AIMA: Chapter 8 (Sections 8.1 and 8.2)

Section 8.3, discussed briefly, is also required reading

Introduction to Artificial Intelligence

CSCE 476-876, Spring 2008

URL: www.cse.unl.edu/~choueiry/S08-476-876

Berthe Y. Choueiry (Shu-we-ri)

choueiry@cse.unl.edu, (402)472-5444

Outline

- First-order logic:
  - basic elements
  - syntax
  - semantics
- Examples

Pros and cons of propositional logic

- 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 $B_{1,1} \wedge P_{1,2}$ 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)
- but...
  Propositional logic has very limited expressive power
  E.g., cannot say "pits cause breezes in adjacent squares"
  except by writing one sentence for each square

Propositional Logic

- is simple
- illustrates important points:
  model, inference, validity, satisfiability, ..
- is restrictive: world is a set of facts
- lacks expressiveness:
  $\rightarrow$ In PL, world contains facts

First-Order Logic

- more symbols (objects, properties, relations)
- more connectives (quantifier)

First Order Logic

→ FOL provides more "primitives" to express knowledge:
  — objects (identity & properties)
  — relations among objects (including functions)

Objects: people, houses, numbers, Einstein, Huskers, event, ..
Properties: smart, nice, large, intelligent, loved, occurred, ..
Relations: brother-of, bigger-than, part-of, occurred-after, ..
Functions: father-of, best-friend, double-of, ..

Examples: (objects? function? relation? property?)
— one plus two equals four [sic]
— squares neighboring the wumpus are smelly

Logic

Attracts: mathematicians, philosophers and AI people

Advantages:
— allows to represent the world and reason about it
— expresses anything that can be programmed

Non-committal to:
— symbols could be objects or relations
  (*e.g.*, King(Gustave), King(Sweden, Gustave), Merciless(King))
— classes, categories, time, events, uncertainty

.. but amenable to extensions: OO FOL, temporal logics, situation/event calculus, modal logic, etc.

⟶ Some people think FOL *is* the language of AI
  true/false? donno :—(but it will remain around for some time..

Types of logic

Logics are characterized by what they commit to as "primitives"

Ontological commitment :
what exists—facts? objects? time? beliefs?

Epistemological commitment :
what states of knowledge?

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief 0…1
Fuzzy logic	degree of truth	degree of belief 0…1

Higher-Order Logic: views relations and functions of FOL as objects

Syntax of FOL: words and grammar

The words: symbols

- Constant symbols stand for objects: QueenMary, 2, UNL, etc.
- Variable symbols stand for objects: x, y, etc.
- Predicate symbols stand for relations: Odd, Even, Brother, Sibling, etc.
- Function symbols stand for functions (viz. relation) Father-of, Square-root, LeftLeg, etc.
- Quantifiyers $\forall$, $\exists$
- Connectives: $\wedge$, $\vee$, $\neg$, $\Rightarrow$, $\Leftrightarrow$,
- (Sometimes) equality $=$

Predicates and functions can have any arity (number of arguments)

Basic elements in FOL (i.e., the grammar)

In propositional logic, every expression is a sentence

In FOL,

- Terms
- Sentences:
  - atomic sentences
  - complex sentences
- Quantifiers:
  - Universal quantifier
  - Existential quantifier

Term

logical expression that refers to an object

— built with: constant symbols, variables, function symbols

$$\begin{aligned} \text{Term} &= function(term_1, \ldots, term_n) \\ & \quad \text{or constant or variable} \end{aligned}$$

— **ground term**: term with no variable

Atomic sentences

state facts

built with terms and predicate symbols

$$\text{Atomic sentence} = \mathit{predicate}(\mathit{term}_1, \ldots, \mathit{term}_n)$$
$$\text{or } \mathit{term}_1 = \mathit{term}_2$$

Examples:

Brother (Richard, John)

Married (FatherOf(Richard), MotherOf(John))

Complex Sentences

built with atomic sentences and logical connectives

$\neg S$

$S_1 \wedge S_2$

$S_1 \vee S_2$

$S_1 \Rightarrow S_2$

$S_1 \Leftrightarrow S_2$

Examples:

Sibling(KingJohn,Richard) $\Rightarrow$ Sibling(Richard,KingJohn)

$>(1,2) \vee \leq(1,2)$

$>(1,2) \wedge \neg{>}(1,2)$

Truth in first-order logic: Semantic

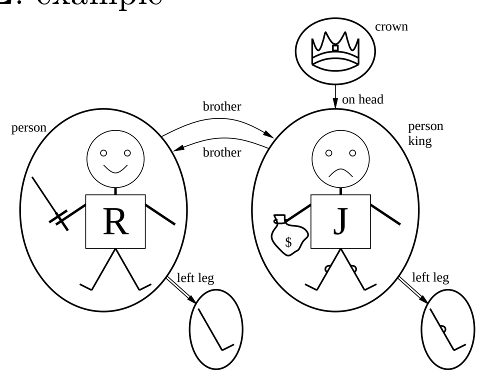
Sentences are true with respect to a model and an interpretation

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*

Model in FOL: example



The domain of a model is the set of objects it contains:
five objects

Intended interpretation: Richard refers Richard the Lion Heart, John refers to Evil King John, Brother refers to brotherhood relation, etc.

Models for FOL: Lots!

We can enumerate the models for a given KB vocabulary:

For each number of domain elements n from 1 to ∞
  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 models is not going to be easy!

There are many possible interpretations, also some model domain are not bounded
$\longrightarrow$ Checking entailment by enumerating is not an option

Quantifiers

allow to make statements about entire collections of objects

- universal quantifier: make statements about <u>everything</u>
- existential quantifier: make statements about <u>some things</u>

Universal quantification

$\forall \langle variables \rangle \ \langle sentence \rangle$

Example: all dogs like bones $\forall\, x Dog(x) \Rightarrow LikeBones(x)$

x = Indy is a dog   x = Indiana Jones is a person

$\forall\, x\ P$ is equivalent to the <u>conjunction</u> of <u>instantiations</u> of P

$$\begin{array}{ll} & Dog(Indy) \Rightarrow LikeBones(Indy) \\ \wedge & Dog(Rebel) \Rightarrow LikeBones(Rebel) \\ \wedge & Dog(KingJohn) \Rightarrow LikeBones(KingJohn) \\ \wedge & \ldots \end{array}$$

Typically: $\Rightarrow$ is the main connective with $\forall$

Common mistake: using $\wedge$ as the main connective with $\forall$

Example: $\forall\, x\ \ Dog(x)\ \wedge\ LikeBones(x)$

all objects in the world are dogs, and all like bones

Existential quantification

$\exists \langle variables \rangle \ \langle sentence \rangle$

Example: some student will talk at the TechFair

$\exists\, x Student(x) \wedge TalksAtTechFair(x)$

Pat, Leslie, Chris are students

$\exists\, x\ P$ is equivalent to the <u>disjunction</u> of <u>instantiations</u> of P

$$\begin{array}{ll} & Student(Pat)\ \wedge TalksAtTechFair(Pat) \\ \vee & Student(Leslie) \wedge TalksAtTechFair(Leslie) \\ \vee & Student(Chris) \wedge TalksAtTechFair(Chris) \\ \vee & \ldots \end{array}$$

Typically: $\wedge$ is the main connective with $\exists$

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

$\exists\, x\ Student(x)\ \Rightarrow\ TalksAtTechFair(x)$

is true if there is anyone who is not Student

Properties of quantifiers (I)

$\forall x\, \forall y$ is the same as $\forall y\, \forall x$

$\exists x\, \exists y$ is the same as $\exists y\, \exists x$

$\exists x\, \forall y$ is not the same as $\forall y\, \exists x$

$\exists x\, \forall y\; Loves(x, y)$
"There is a person who loves everyone in the world"

$\forall y\, \exists x Loves(x, y)$
"Everyone in the world is loved by at least one person"

Quantifier duality: each can be expressed using the other

$\forall x\; Likes(x, IceCream)$ $\neg\, \exists x\, \neg Likes(x, IceCream)$

$\exists x\; Likes(x, Broccoli)$ $\neg\, \forall x\, \neg Likes(x, Broccoli)$

Parsimony principal: $\forall$, $\neg$, and $\Rightarrow$ are sufficient

Properties of quantifiers (II)

Nested quantifier:
$\forall\, x(\exists\, y(P(x, y))$:
every object in the world has a particular property, which is the property to be related to some object by the relation P

$\exists\;\; x\; (\forall\, y(P(x, y))$:
there is some object in the world that has a particular property, which is the property to be related to every object by the relation P

Lexical scoping: $\forall\, x[Cat(x) \vee \exists\, xBrother(Richard, x)]$

Well-formed formulas (WFF): (kind of correct spelling)
every variable must be introduced by a quantifier
$\forall\, xP(y)$ is not a WFF

Examples

Brothers are siblings

.

"Sibling" is symmetric

.

One's mother is one's female parent

.

A first cousin is a child of a parent's sibling

Examples

.

$\forall x, y\ Brother(x, y) \Rightarrow Sibling(x, y)$

.

$\forall x, y\ Sibling(x, y) \Rightarrow Sibling(y, x)$

.

$\forall x, y\ Mother(x, y) \Rightarrow (Female(x) \wedge Parent(x, y))$

.

$\forall x, y\ FirstCousin(x, y) \Leftrightarrow$
$\exists a, b\ Parent(a, x) \wedge Sibling(a, b) \wedge Parent(b, y)$

Tricky example

Someone is loved by everyone

$\exists x \, \forall y \; Loves(y, x)$

Someone with red-hair is loved by everyone

$\exists x \, \forall y \; Redhair(x) \wedge Loves(y, x)$

Alternatively:

$\exists x \; Person(x) \wedge Redhair(x) \wedge (\forall y \; Person(y) \Rightarrow Loves(y, x))$

Equality

$term_1 = term_2$ is true under a given interpretation
if and only if $term_1$ and $term_2$ refer to the same object

Examples

- Father(John)=Henry
- $1 = 2$ is satisfiable
- $2 = 2$ is valid
- Useful to distinguish two objects:
  – Definition of (full) *Sibling* in terms of *Parent*:
  $\forall x, y \; Sibling(x, y) \Leftrightarrow [\neg(x = y) \wedge \exists m, f \neg(m = f) \wedge Parent(m, x) \wedge Parent(f, x) \wedge Parent(m, y) \wedge Parent(f, y)]$
  – Spot has at least two sisters: ...

AIMA, Exercise 8.4 & 8.7

Knowledge representation (KR)

Domain: a section of the world about which we wish to express some knowledge

Example: Family relations (kinship):
– Objects: people
– Properties: gender, married, divorced, single, widowed
– Relations: parenthood, brotherhood, marriage..

Unary predicates: Male, Female
Binary relations: Parent, Sibling, Brother, Child, etc.
Functions: Mother, Father

$\forall\, m, c, Mother(c) = m \;\Leftrightarrow\; Female(m) \land Parent(m, c)$

In Logic (informally)

- Basic facts: **<u>axioms</u>** (definitions)
- Derived facts: **<u>theorems</u>**

Independent axiom
an axiom that cannot be derived from the rest
$\longrightarrow$ Goal of mathematicians: find the minimal set of independent axioms

In AI

- Assertions: sentences added to a KB using TELL
- Queries or goals: sentences asked to KB using ASK

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 $t = 5$:

$Tell(KB, Percept([Smell, Breeze, None], 5))$
$Ask(KB, \exists a Action(a, 5))$

I.e., does the KB entail any particular actions at $t = 5$?

Answer: Yes, $\{a/Shoot\}$ $\leftarrow$ substitution (binding list)

Given a sentence S and a substitution σ,
$S\sigma$ denotes the result of plugging σ into S; e.g.,
$S = Smarter(x, y)$
$\sigma = \{x/Hillary, y/Bill\}$
$S\sigma = Smarter(Hillary, Bill)$

$Ask(KB, S)$ returns some/all σ such that $KB \models S\sigma$

Prepare for next lecture: AIMA, Exercise 8.6, page 268

Takes(x, c, s): student x takes course c in semester s

Passes(x, c, s): student x passes course c in semester s

Score(x, c, s): the score obtained by student x in course c in semester s

$x > y$: x is greater that y

F and G: specific French and Greek courses

Buys(x, y, z): x buys y from z

Sells(x, y, z): x sells y from z

Shaves(x, y): person x shaves person y

Born(x, c): person x is born in country c

Parent(x, y): person x is parent of person y

Citizen(x, c, r): person x is citizen of country c for reason r

Resident(x, c): person x is resident of country c of person y

Fools(x, y, t): person x fools person y at time t

Student (x), Person(x), Man(x), Barber(x), Expensive(x), Agent(x),
Insured(x), Smart(x), Politician(x),

AI Limerick

If your thesis is utterly vacuous
Use first-order predicate calculus
With sufficient formality
The sheerest banality
Will be hailed by the critics: "Miraculous!"

Henry Kautz

In Canadian Artificial Intelligence, September 1986

(then: University of Rochester

then: head of AI at AT&T Labs-Research

and Program co-chair of AAAI-2000

Now: Associate Professor at University of Washington, Seattle)