Artificial Intelligence (First-Order Logic)

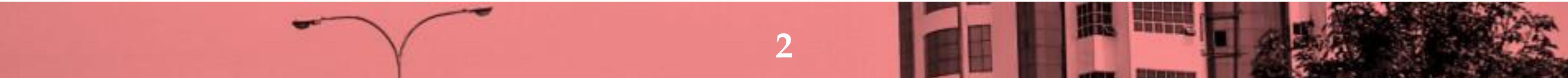
Syntax and Semantics of First-Order Logic, Using First Order Logic

Module 5 Lecture 1

The CSC415 Team 2018/2019

LESSON OUTLINE

- More on Representation
- Syntax and Semantics of First-Order Logic
- Using First Order Logic
- Knowledge Engineering in First-Order Logic



First-Order Logic

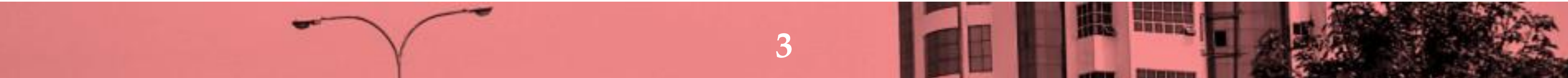
- Propositional logic assumes that the world contains facts.

- First-order logic (like natural language) assumes the world contains

  – Objects: people, houses, numbers, colors, baseball games, wars, …

  – Relations: red, round, prime, brother of, bigger than, part of, comes between, …

  – Functions: father of, best friend, one more than, plus, …



Logics in General

Language	**Ontological Commitment**	**Epistemological Commitment**
Propositional Logic	Facts	True / False / Unknown
First-Order Logic	Fact, objects, relations	True / False / Unknown
Temporal Logic	Facts, objects, relations, times	True / False / Unknown
Probability Theory	Facts	Degree of belief $\in [0,1]$
Fuzzy Logic	Degree of truth $\in [0,1]$	Known interval value

Syntax of FOL: Basic elements

- Constant Symbols:
  - Stand for objects
  - e.g., KingJohn, 2, UCI,...

- Predicate Symbols
  - Stand for relations
  - E.g., Brother(Richard, John), greater_than(3,2)...

- Function Symbols
  - Stand for functions
  - E.g., Sqrt(3), LeftLegOf(John),...

Syntax of FOL: Basic elements

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

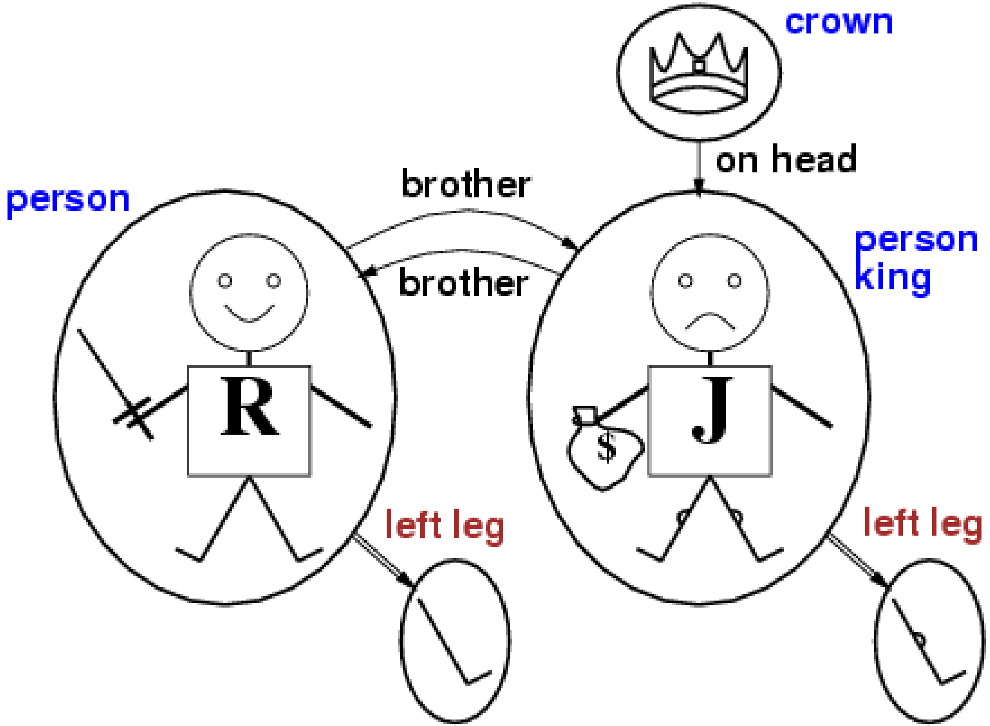
Relations

- Some relations are **properties**:
  - they state some fact about a single object: Round(ball), Prime(7).

- **n-ary** relations state facts about two or more objects:
  - Married(John,Mary), LargerThan(3,2).

- Some relations are **functions**:
  - their value is another object:
    - Plus(2,3), Father(Dan).

Models for FOL: Graphical Example



Tabular Representation

- A FOL model is basically equivalent to a relational database instance.
- Historically, the relational data model comes from FOL.

Student

s-id	Intelligence	Ranking
Jack	3	1
Kim	2	1
Paul	1	2

Course

c-id	Rating	Difficulty
101	3	1
102	2	2

Professor

p-id	Popularity	Teaching-a
Oliver	3	1
Jim	2	1

RA

s-id	p-id	Salary	Capability
Jack	Oliver	High	3
Kim	Oliver	Low	1
Paul	Jim	Med	2

Registration

s-id	c.id	Grade	Satisfaction
Jack	101	A	1
Jack	102	B	2
Kim	102	A	1
Paul	101	B	1

Terms

- Term = logical expression that refers to an object.

- There are 2 kinds of terms:
  - constant symbols: Table, Computer
  - function symbols: LeftLeg(Pete), Sqrt(3), Plus(2,3) etc
- Functions can be nested:
  - Pat_Grandfather(x) = father(father(x))
- Terms can contain variables.
- No variables = **ground term**.

Atomic Sentences

- Atomic sentences state facts using terms and predicate symbols
  - P(x,y) interpreted as “x is P of y”
- Examples:
  - LargerThan(2,3) is false.
  - Brother_of(Mary,Pete) is false.
  - Married(Father(Richard), Mother(John)) could be true or false
- Note: Functions do not state facts and form no sentence:
  - Brother(Pete) refers to John (his brother) and is neither true nor false.
- Brother_of(Pete,Brother(Pete)) is True.

Binary relation Function

Complex Sentences

- We make complex sentences with connectives (just like in propositional logic).

property

$$\neg Brother(LeftLeg(Richard), John) \vee (Democrat(Bush))$$

binary relation

function

objects

connectives

More Examples

- Brother(Richard, John) $\wedge$ Brother(John, Richard)
- King(Richard) $\vee$ King(John)
- King(John) => $\neg$ King(Richard)
- LessThan(Plus(1,2) ,4) $\wedge$ GreaterThan(1,2)

- (Semantics are the same as in propositional logic)

Variables

- Person(John) is true or false because we give it a single argument 'John'

- We can be much more flexible if we allow variables which can take on values in a domain. e.g., all persons x, all integers i,   etc.
  - E.g., can state rules like Person(x) => HasHead(x)

  - or Integer(i) => Integer(plus(i,1))

Universal Quantification $\forall$

- $\forall$ means "for all"
- Allows us to make statements about all objects that have certain properties
- Can now state general rules:
  - $\forall$ x King(x) => Person(x)
  - $\forall$ x Person(x) => HasHead(x)
  - $\forall$ i Integer(i) => Integer(plus(i,1))
- Note that
  - $\forall$ x King(x) $\wedge$ Person(x) is not correct!
  - This would imply that all objects x are Kings and are People
  - $\forall$ x King(x) => Person(x) is the correct way to say

Existential Quantification ∃

- ∃ x means "there exists an x such that…." (at least one object x)
- Allows us to make statements about some object without naming it
- Examples:
  - ∃ x King(x)
  - ∃ x Lives_in(John, Castle(x))
  - ∃ i Integer(i) ∧ GreaterThan(i,0)
- Note that ∧ is the natural connective to use with ∃
- (And => is the natural connective to use with ∀)

Combining Quantifiers

- $\forall$ x $\exists$ y Loves(x,y)
  - For everyone ("all x") there is someone ("y") that they love.
- y $\forall$ x Loves(x,y)
  - there is someone ("y") who is loved by everyone
- Clearer with parentheses:
  - $\exists$ y ($\forall$ x Loves(x,y))

Exercise

- Represent the following sentences in first-order logic using a consistent vocabulary you must define.
  - a) Not all student take both History and Biology
  - b) Only one student failed History
  - c) Only one student failed both History and Biology
  - d) The best score in History was better than the best score in Biology
  - e) Every person who dislikes all vegetarian is smart