CS 1571 Introduction to AI
Lecture 12

First order logic

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Administration

- **PS-4:**
  - **Due today**
- **PS-5**
  - **Out today**
  - **Propositional and First-order Logic**

Limitations of propositional logic

World we want to represent and reason about consists of a number of objects with variety of properties and relations among them

Propositional logic:

- Represents statements about the world without reflecting this structure and without modeling these entities explicitly

Consequence:

- some knowledge is hard or impossible to encode in the propositional logic.
- Two cases that are hard to represent:
  - **Statements about similar objects, relations**
  - **Statements referring to groups of objects**.

Limitations of propositional logic

- **Statements about similar objects and relations needs to be enumerated**
- **Example:** Seniority of people domain

  For inferences we need:

  John is older than Mary $\wedge$ *Mary is older than Paul*
  $\Rightarrow$ *John is older than Paul*

  Jane is older than Mary $\wedge$ *Mary is older than Paul*
  $\Rightarrow$ *Jane is older than Paul*

- **Problem:** if we have many people and facts about their seniority we need represent many rules like this to allow inferences
- **Possible solution: introduce variables**

  PersA *is older than* ***PersB*** $\wedge$ ***PersB*** *is older than* ***PersC***
  $\Rightarrow$ ***PersA*** *is older than* ***PersC***

Limitations of propositional logic

- **Statements referring to groups of objects require exhaustive enumeration of objects**
- **Example:**

Assume we want to express *Every student likes vacation*

Doing this in propositional logic would require to include statements about every student

John likes vacation ∧
Mary likes vacation ∧
Ann likes vacation ∧
…

- **Solution:** Allow quantification in statements

First-order logic (FOL)

- More expressive than **propositional logic**

- **Eliminates deficiencies of PL by:**
  - Representing objects, their properties, relations and statements about them;
  - Introducing variables that refer to an arbitrary objects and can be substituted by a specific object
  - Introducing quantifiers allowing quantification statements over objects without the need to represent each of them separately

Logic

Logic is defined by**:**

- **A set of sentences**
  - A sentence is constructed from a set of primitives according to syntax rules.
- **A set of interpretations**
  - An interpretation gives a semantic to primitives. It associates primitives with objects, values in the real world.
- **The valuation (meaning) function** V
  - Assigns a truth value to a given sentence under some interpretation

$$V : \text{sentence} \times \text{interpretation} \rightarrow \{True, False\}$$

First-order logic. Syntax.

Term - syntactic entity for representing objects

Terms in FOL:

- **Constant symbols:**
  - E.g. *John, France, car89*
- **Variables:**
  - E.g. *x,y,z*
- **Functions** applied to one or more terms
  - E.g. *father-of (John)*
    father-of(father-of(John))

First order logic. Syntax.

Sentences in FOL:

- **Atomic sentences:**
  - **A predicate symbol** applied to 0 or more terms

    Examples:

    Red(car12),

    Sister(Amy, Jane);

    Manager(father-of(John));

  - t1 = t2 **equivalence** of terms

    Example:

    John = father-of(Peter)

First order logic. Syntax.

Sentences in FOL:

- **Complex sentences:**
- Assume ϕ, ψ are sentences. Then:
  - $(\phi \wedge \psi)$ $(\phi \vee \psi)$ $(\phi \Rightarrow \psi)$ $(\phi \Leftrightarrow \psi)$ $\neg \psi$

    and
  - $\forall x\ \phi$ $\exists y\ \phi$

  are sentences

Symbols $\exists, \forall$

- stand for the **existential** and the **universal** quantifier

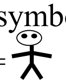
Semantics. Interpretation.

An interpretation ***I*** is defined by a **domain** and a **mapping**

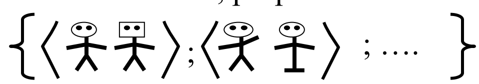
- **domain D:** a set of objects in the world we represent; domain of discourse;

An interpretation *I* maps:

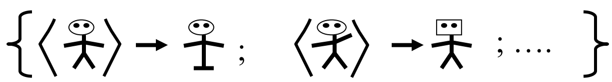
- Constant symbols to objects in D

  I(*John*) = 

- Predicate symbols to relations, properties on D

  I(*brother*) = 

- Function symbols to functional relations on D

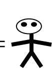
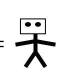
  I(*father-of*) = 

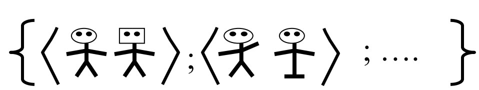
Semantics of sentences.

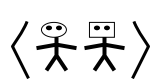
Meaning (evaluation) function:

V : sentence × interpretation → {*True* , *False* }

A **predicate** *predicate(term-1, term-2, term-3, term-n)* is true for the interpretation *I* , iff the objects referred to by *term-1, term-2, term-3, term-n* are in the relation referred to by *predicate*

I(*John*) =  I(*Paul*) = 

I(*brother*) = 

brother(John, Paul) =  in I(*brother*)

V(*brother(John, Paul), I*) = ***True***

Semantics of sentences.

- **Equality** $V(term\text{-}1 = term\text{-}2, I) = \textbf{True}$
  Iff $I(term\text{-}1) = I(term\text{-}2)$
- **Boolean expressions: standard**

  E.g. $V(sentence\text{-}1 \lor sentence\text{-}2, I) = \textbf{True}$
  Iff $V(sentence\text{-}1, I) = \textbf{True}$ or $V(sentence\text{-}2, I) = \textbf{True}$

- **Quantifications**

  $V(\forall x\ \phi, I) = \textbf{True}$
  Iff for all $d \in D$ $V(\phi, \underbrace{I[x/d]}_{\text{substitution of } x \text{ with } d}) = \textbf{True}$

  $V(\exists x\ \phi, I) = \textbf{True}$
  Iff there is a $d \in D$, s.t. $V(\phi, I[x/d]) = \textbf{True}$

Examples of sentences with quantifiers

- **Universal quantification**

  All Upitt students are smart

  $\forall x\ student(x) \land at(x, Upitt) \Rightarrow smart(x)$

Typically the universal quantifier connects with an implication

- **Existential quantification**

  Someone at CMU is smart

  $\exists x\ at(x, CMU) \land smart(x)$

Typically the existential quantifier connects with a conjunction

Order of quantifiers

- **Order of quantifiers of the same type does not matter**

For all x and y, if x is a parent of y then y is a child of x

$$\forall x, y \; parent(x, y) \Rightarrow child(y, x)$$
$$\forall y, x \; parent(x, y) \Rightarrow child(y, x)$$

- **Order of different quantifiers changes the meaning**

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

Order of quantifiers

- **Order of quantifiers of the same type does not matter**

For all x and y, if x is a parent of y then y is a child of x

$$\forall x, y \; parent(x, y) \Rightarrow child(y, x)$$
$$\forall y, x \; parent(x, y) \Rightarrow child(y, x)$$

- **Order of different quantifiers changes the meaning**

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

Everybody loves somebody

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

Order of quantifiers

- **Order of quantifiers of the same type does not matter**

  For all x and y, if x is a parent of y then y is a child of x

  $$\forall x, y \ \ parent(x, y) \Rightarrow child(y, x)$$

  $$\forall y, x \ \ parent(x, y) \Rightarrow child(y, x)$$

- **Order of different quantifiers changes the meaning**

  $$\forall x \exists y \ loves(x, y)$$

  Everybody loves somebody

  $$\exists y \forall x \ loves(x, y)$$

  There is someone who is loved by everyone

Connections between quantifiers

Everyone likes ice cream

?

Connections between quantifiers

Everyone likes ice cream

$$\forall x \; likes(x, IceCream)$$

Connections between quantifiers

Everyone likes ice cream

$$\forall x \; likes(x, IceCream)$$

Is it possible to convey the same meaning using an existential quantifier ?

Connections between quantifiers

Everyone likes ice cream

$\forall x \; likes(x, IceCream)$

Is it possible to convey the same meaning using an existential quantifier ?

There is no one who does not like ice cream

$\neg \exists x \; \neg likes(x, IceCream)$

A universal quantifier in the sentence can be expressed using an existential quantifier !!!

Connections between quantifiers

Someone likes ice cream

?

Connections between quantifiers

Someone likes ice cream

$$\exists x \; likes(x, IceCream)$$

Is it possible to convey the same meaning using a universal quantifier ?

Connections between quantifiers

Someone likes ice cream

$$\exists x \; likes(x, IceCream)$$

Is it possible to convey the same meaning using a universal quantifier ?

Not everyone does not like ice cream

$$\neg \forall \, x \; \neg likes(x, IceCream)$$

An existential quantifier in the sentence can be expressed using a universal quantifier !!!

Representing knowledge in FOL

Example:

Kinship domain

- **Objects:** people

  $John\ ,\ Mary\ ,\ Jane\ , \ldots$
- **Properties:** gender

  $Male\ (x), Female\ \ (x)$
- **Relations:** parenthood, brotherhood, marriage

  $Parent\ \ (x, y), Brother\ \ (x, y), Spouse\ \ (x, y)$
- **Functions:** mother-of (one for each person x)

  $MotherOf\ \ \ (x)$

Kinship domain in FOL

Relations between predicates and functions: write down what we know about them; how relate to each other.

- Male and female are disjoint categories
- Parent and child relations are inverse
- A grandparent is a parent of parent
- A sibling is another child of one's parents
- And so on ….

Kinship domain in FOL

Relations between predicates and functions: write down what we know about them; how relate to each other.

- Male and female are disjoint categories

  $$\forall x \; Male(x) \Leftrightarrow \neg Female(x)$$

- Parent and child relations are inverse
- A grandparent is a parent of parent
- A sibling is another child of one's parents
- And so on ….

Kinship domain in FOL

Relations between predicates and functions: write down what we know about them; how relate to each other.

- Male and female are disjoint categories

  $$\forall x \; Male(x) \Leftrightarrow \neg Female(x)$$

- Parent and child relations are inverse

  $$\forall x, y \; Parent(x, y) \Leftrightarrow Child(y, x)$$

- A grandparent is a parent of parent
- A sibling is another child of one's parents
- And so on ….

Kinship domain in FOL

Relations between predicates and functions: write down what we know about them; how relate to each other.

- Male and female are disjoint categories

  $$\forall x\ Male\ (x) \Leftrightarrow \neg Female\ (x)$$

- Parent and child relations are inverse

  $$\forall x, y\ Parent\ (x, y) \Leftrightarrow Child\ (y, x)$$

- A grandparent is a parent of parent

  $$\forall g, c\ Grandparent(g, c) \Leftrightarrow \exists p\ Parent(g, p) \wedge Parent(p, c)$$

- A sibling is another child of one's parents
- And so on ….

Kinship domain in FOL

Relations between predicates and functions: write down what we know about them; how relate to each other.

- Male and female are disjoint categories

  $$\forall x\ Male\ (x) \Leftrightarrow \neg Female\ (x)$$

- Parent and child relations are inverse

  $$\forall x, y\ Parent\ (x, y) \Leftrightarrow Child\ (y, x)$$

- A grandparent is a parent of parent

  $$\forall g, c\ Grandparent(g, c) \Leftrightarrow \exists p\ Parent(g, p) \wedge Parent(p, c)$$

- A sibling is another child of one's parents

  $$\forall x, y\ Sibling\ (x, y) \Leftrightarrow (x \neq y) \wedge \exists p\ Parent\ (p, x) \wedge Parent\ (p, y)$$

- And so on ….

Logical inference in FOL

Logical inference problem:

- Given a knowledge base KB (a set of sentences) and a sentence α , does the KB semantically entail α ?

$$KB \models \alpha \quad ?$$

In other words: In all interpretations in which sentences in the KB are true, is also α true?

Logical inference problem in the first-order logic is undecidable !!!. No procedure that can decide the entailment for all possible input sentences in a finite number of steps.

Truth table approach

- **Is it possible to modify the truth table approach also to the first-order logic (FOL)?**
- **Truth table approach:**
  - Generate all interpretations
  - Find the ones for which the KB evaluates to true
  - Check whether the theorem evaluates to true for all KB consistent interpretations

Inference rules approach

Advantage: Does not generate all possible interpretations

- **Inference rules from the propositional logic:**
  - **Modus ponens** $\dfrac{A \Rightarrow B, \quad A}{B}$
  - **Resolution** $\dfrac{A \vee B, \quad \neg B \vee C}{A \vee C}$
  - and others: And-introduction, And-elimination, Or-introduction, Negation elimination
- **Additional inference rules** are needed for sentences with quantifiers and variables
  - Must involve variable substitutions

Sentences with variables

First-order logic sentences can include variables.

- **Variable** is:
  - **Bound** – if it is in the scope of some quantifier
    $\forall x \; P(x)$
  - **Free** – if it is not bound.
    $\exists x \; P(y) \wedge Q(x)$ y is free
- **Sentence** (formula) is:
  - **Closed** – if it has no free variables
    $\forall y \exists x \; P(y) \Rightarrow Q(x)$
  - **Open** – if it is not closed
  - **Ground** – if it does not have any variables
    $Likes(John, Jane)$

Variable substitutions

- Variables in the sentences can be substituted with terms. (**terms = constants, variables, functions**)
- **Substitution:**
  - Is represented by a mapping from variables to terms

$$\{x_1 / t_1, x_2 / t_2, \ldots\}$$

  - Application of the substitution to sentences

$$SUBST(\{x / Sam, y / Pam\}, Likes(x, y)) = Likes(Sam, Pam)$$

$$SUBST(\{x / z, y / fatherof(John)\}, Likes(x, y)) = Likes(z, fatherof(John))$$

Inference rules for quantifiers

- **Universal elimination**

$$\frac{\forall x\ \phi(x)}{\phi(a)}$$ a - is a constant symbol

  - substitutes a variable with a constant symbol

$$\forall x\ Likes(x, IceCream) \qquad Likes(Ben, IceCream)$$

- **Existential elimination.**

$$\frac{\exists x\ \phi(x)}{\phi(a)}$$

  - Substitutes a variable with a constant symbol that does not appear elsewhere in the KB

$$\exists x\ Kill(x, Victim) \qquad Kill(Murderer, Victim)$$

Inference rules for quantifiers

- **Universal instantiation (introduction)**

$$\frac{\phi}{\forall x\ \phi} \qquad x - \text{is not free in } \phi$$

  - Introduces a universal variable which does not affect ϕ or its assumptions

$$Sister(Amy, Jane) \qquad \forall x\ Sister(Amy, Jane)$$

- **Existential instantiation (introduction)**

$$\frac{\phi(a)}{\exists x \phi(x)} \qquad \begin{array}{l} a - \text{is a ground term in } \phi \\ x - \text{is not free in } \phi \end{array}$$

  - Substitutes a ground term in the sentence with a variable and an existential statement

$$Likes(Ben, IceCream) \qquad \exists x\ Likes(x, IceCream)$$

Unification

- **Problem in inference:** Universal elimination gives many opportunities for substituting variables with ground terms

$$\frac{\forall x\ \phi(x)}{\phi(a)} \qquad a \text{ - is a constant symbol}$$

- **Solution:** Try substitutions that may help
  - Use substitutions of "similar" sentences in KB

- **Unification** – takes two similar sentences and computes the substitution that makes them look the same, if it exists

$$UNIFY\ (p, q) = \sigma \ \text{ s.t. } SUBST(\sigma, p) = SUBST\ (\sigma, q)$$

Unification. Examples.

- **Unification:**

  $UNIFY(p,q) = \sigma$ s.t. $SUBST(\sigma, p) = SUBST(\sigma, q)$

- **Examples:**

$$UNIFY(Knows(John, x), Knows(John, Jane)) = \{x / Jane\}$$

$$UNIFY(Knows(John, x), Knows(y, Ann)) = ?$$

Unification. Examples.

- **Unification:**

  $UNIFY(p,q) = \sigma$ s.t. $SUBST(\sigma, p) = SUBST(\sigma, q)$

- **Examples:**

$$UNIFY(Knows(John, x), Knows(John, Jane)) = \{x / Jane\}$$

$$UNIFY(Knows(John, x), Knows(y, Ann)) = \{x / Ann, y / John\}$$

$$UNIFY(Knows(John, x), Knows(y, MotherOf(y))) \\ = ?$$

Unification. Examples.

- **Unification:**

  $UNIFY(p, q) = \sigma$ s.t. $SUBST(\sigma, p) = SUBST(\sigma, q)$

- **Examples:**

$$UNIFY(Knows(John, x), Knows(John, Jane)) = \{x / Jane\}$$

$$UNIFY(Knows(John, x), Knows(y, Ann)) = \{x / Ann, y / John\}$$

$$UNIFY(Knows(John, x), Knows(y, MotherOf(y))) = \{x / MotherOf(John), y / John\}$$

$$UNIFY(Knows(John, x), Knows(x, Elizabeth)) = ?$$

Unification. Examples.

- **Unification:**

  $UNIFY(p, q) = \sigma$ s.t. $SUBST(\sigma, p) = SUBST(\sigma, q)$

- **Examples:**

$$UNIFY(Knows(John, x), Knows(John, Jane)) = \{x / Jane\}$$

$$UNIFY(Knows(John, x), Knows(y, Ann)) = \{x / Ann, y / John\}$$

$$UNIFY(Knows(John, x), Knows(y, MotherOf(y))) = \{x / MotherOf(John), y / John\}$$

$$UNIFY(Knows(John, x), Knows(x, Elizabeth)) = fail$$