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:
  
  \[
  \begin{align*}
  John & & is & & older & & than & & Mary & & \wedge & & Mary & & is & & older & & than & & Paul \\
  \Rightarrow & & John & & is & & older & & than & & Paul \\
  Jane & & is & & older & & than & & Mary & & \wedge & & Mary & & is & & older & & than & & Paul \\
  \Rightarrow & & Jane & & is & & older & & than & & Paul 
  \end{align*}
  \]

- 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

\[
\begin{align*}
PersA & & is & & older & & than & & PersB & & \wedge & & PersB & & is & & older & & than & & PersC \\
\Rightarrow & & PersA & & is & & older & & than & & PersC
\end{align*}
\]
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

  \[ \text{John likes vacation} \land \text{Mary likes vacation} \land \text{Ann likes vacation} \land \ldots \]

- **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

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

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)
    
    $$\text{father-of(father-of(John))}$$
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)

• Complex sentences:
  – Assume $\phi, \psi$ are sentences. Then:
    \[
    (\phi \land \psi) \quad (\phi \lor \psi) \quad (\phi \Rightarrow \psi) \quad (\phi \Leftrightarrow \psi) \quad \neg \psi
    \]
    and
    \[
    \forall x \phi \quad \exists y \phi
    \]
    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 \( D \)

  \[
  I(John) = \overline{\text{John}}
  \]

- Predicate symbols to relations, properties on \( D \)

  \[
  I(\text{brother}) = \begin{Bmatrix}
  \langle \overline{\text{John}} \overline{\text{Paul}} \rangle; \langle \overline{\text{Paul}} \overline{\text{John}} \rangle; \ldots
  \end{Bmatrix}
  \]

- Function symbols to functional relations on \( D \)

  \[
  I(\text{father-of}) = \begin{Bmatrix}
  \langle \overline{\text{John}} \rangle \rightarrow \overline{\text{Paul}}; \langle \overline{\text{Paul}} \rangle \rightarrow \overline{\text{John}}; \ldots
  \end{Bmatrix}
  \]

---

Semantics of sentences.

**Meaning (evaluation) function:**

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

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

\[
I(John) = \overline{\text{John}} \quad I(Paul) = \overline{\text{Paul}}
\]

\[
I(\text{brother}) = \begin{Bmatrix}
\langle \overline{\text{John}} \overline{\text{Paul}} \rangle; \langle \overline{\text{Paul}} \overline{\text{John}} \rangle; \ldots
\end{Bmatrix}
\]

\[
\text{brother}(John, Paul) = \langle \overline{\text{John}} \overline{\text{Paul}} \rangle \quad \text{in } I(\text{brother})
\]

\[
V(\text{brother}(John, Paul), I) = \text{True}
\]
Semantics of sentences.

• **Equality**
  \[ V(\text{term-1} = \text{term-2}, I) = \text{True} \]
  Iff \[ I(\text{term-1}) = I(\text{term-2}) \]

• **Boolean expressions:** standard
  
  E.g. \[ V(\text{sentence-1} \lor \text{sentence-2}, I) = \text{True} \]
  Iff \[ V(\text{sentence-1}, I) = \text{True} \text{ or } V(\text{sentence-2}, I) = \text{True} \]

• **Quantifications**
  
  \[ V(\forall x \phi, I) = \text{True} \]
  Iff for all \( d \in D \) \[ V(\phi, I[x/d]) = \text{True} \]

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

Examples of sentences with quantifiers

• **Universal quantification**
  
  *All Upitt students are smart*
  \[ \forall x \text{ student}(x) \land \text{at}(x, \text{Upitt}) \Rightarrow \text{smart}(x) \]

  Typically the universal quantifier connects with an implication

• **Existential quantification**
  
  *Someone at CMU is smart*
  \[ \exists x \text{ at}(x, \text{CMU}) \land \text{smart}(x) \]

  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 $x$
  
  \[
  \forall x, y \text{ parent } (x, y) \Rightarrow \text{child } (y, x)
  \]

  \[
  \forall y, x \text{ parent } (x, y) \Rightarrow \text{child } (y, x)
  \]

- **Order of different quantifiers changes the meaning**
  
  \[
  \forall x \exists y \text{ loves } (x, y)
  \]

---

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 $x$
  
  \[
  \forall x, y \text{ parent } (x, y) \Rightarrow \text{child } (y, x)
  \]

  \[
  \forall y, x \text{ parent } (x, y) \Rightarrow \text{child } (y, x)
  \]

- **Order of different quantifiers changes the meaning**
  
  Everybody loves somebody
  
  \[
  \forall x \exists y \text{ loves } (x, y)
  \]

  \[
  \exists y \forall x \text{ loves } (x, y)
  \]
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

\[ \forall x \text{ likes } (x, \text{IceCream }) \]

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

*Everyone likes ice cream*

$$\forall x \text{ likes (} x, \text{ 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 \text{ likes (} x, \text{ 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

\( \exists x \text{ likes } (x, \text{IceCream} ) \)

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

Not everyone does not like ice cream

\( \neg \forall x \neg \text{ likes } (x, \text{IceCream} ) \)

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, …
- **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
  \( \forall x \ Male (x) \Leftrightarrow \lnot Female (x) \)

• 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
  \( \forall x \ Male (x) \iff \neg Female (x) \)

• Parent and child relations are inverse
  \( \forall x, y \ Parent (x, y) \iff Child (y, x) \)

• A grandparent is a parent of parent
  \( \forall g, c \ Grandparent(g, c) \iff \exists p \ Parent(g, p) \land Parent(p, c) \)

• A sibling is another child of one’s parents
  \( \forall x, y \ Sibling (x, y) \iff (x \neq y) \land \exists p \ Parent (p, x) \land Parent (p, y) \)

• And so on ….
Logical inference in FOL

**Logical inference problem:**
- Given a knowledge base KB (a set of sentences) and a sentence $\alpha$, does the KB semantically entail $\alpha$?

$$KB \models \alpha ?$$

In other words: In all interpretations in which sentences in the KB are true, is also $\alpha$ 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 \[ \frac{A \rightarrow B, \ A}{B} \]
  - Resolution \[ \frac{A \lor B, \ \neg B \lor C}{A \lor C} \]

  - 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.
    \[ \exists x \ P(y) \land Q(x) \quad y \text{ is free} \]

- **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
    \[ \text{Likes}(John, Jane) \]
Variable substitutions

- Variables in the sentences can be substituted with terms.
  \( \text{terms} = \text{constants, variables, functions} \)

- **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
    \[ \text{SUBST} \{ x / Sam, y / Pam \}, \text{Likes}(x, y) \} = \text{Likes}(Sam, Pam) \]
    \[ \text{SUBST} \{ x / z, y / \text{fatherof}(John) \}, \text{Likes}(x, y) \} = \text{Likes}(z, \text{fatherof}(John)) \]

Inference rules for quantifiers

- **Universal elimination**
  \[ \frac{\forall x \, \phi(x)}{\phi(a)} \quad a - \text{is a constant symbol} \]
  - Substitutes a variable with a constant symbol
    \[ \forall x \, \text{Likes}(x, \text{IceCream}) \quad \text{Likes}(Ben, \text{IceCream}) \]

- **Existential elimination.**
  \[ \frac{\exists x \, \phi(x)}{\phi(a)} \]
  - Substitutes a variable with a constant symbol that does not appear elsewhere in the KB
    \[ \exists x \, \text{Kill}(x, \text{Victim}) \quad \text{Kill}(\text{Murderer}, \text{Victim}) \]
Inference rules for quantifiers

- **Universal instantiation (introduction)**
  \[
  \frac{\phi}{\forall x \phi}
  \]
  \(x\) – is not free in \(\phi\)
  - Introduces a universal variable which does not affect \(\phi\) or its assumptions
  \[
  \text{Sister}(Amy, Jane) \quad \forall x \text{Sister}(Amy, Jane)
  \]

- **Existential instantiation (introduction)**
  \[
  \frac{\phi(a)}{\exists x \phi(x)}
  \]
  \(a\) – is a ground term in \(\phi\)
  \(x\) – is not free in \(\phi\)
  - Substitutes a ground term in the sentence with a variable and an existential statement
  \[
  \text{Likes}(Ben, IceCream) \quad \exists x \text{Likes}(x, IceCream)
  \]

Unification

- **Problem in inference:** Universal elimination gives many opportunities for substituting variables with ground terms
  \[
  \frac{\forall x \phi(x)}{\phi(a)}
  \]
  \(a\) - 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
  \[
  \text{UNIFY} \ (p, q) = \sigma \quad \text{s.t.} \ \text{SUBST} (\sigma, p) = \text{SUBST} (\sigma, q)
  \]
Unification. Examples.

- **Unification:**
  \[ \text{UNIFY} (p, q) = \sigma \quad \text{s.t.} \quad \text{SUBST}(\sigma, p) = \text{SUBST}(\sigma, q) \]

- **Examples:**
  \[ \text{UNIFY}(\text{Knows}(\text{John}, x), \text{Knows}(\text{John}, \text{Jane})) = \{x / \text{Jane}\} \]
  \[ \text{UNIFY}(\text{Knows}(\text{John}, x), \text{Knows}(y, \text{Ann})) = ? \]
Unification. Examples.

• **Unification:**
  
  \[ UNIFY(p, q) = \sigma \text{ s.t. \textit{SUBST}(\sigma, p) = \textit{SUBST}(\sigma, q)} \]

• **Examples:**

  \[ UNIFY(\text{Knows}(John, x), \text{Knows}(John, Jane)) = \{x / Jane\} \]

  \[ UNIFY(\text{Knows}(John, x), \text{Knows}(y, Ann)) = \{x / Ann, y / John\} \]

  \[ UNIFY(\text{Knows}(John, x), \text{Knows}(y, \text{MotherOf}(y))) \]
  \[ = \{x / \text{MotherOf}(John), y / John\} \]

  \[ UNIFY(\text{Knows}(John, x), \text{Knows}(x, Elizabeth)) = \text{fail} \]