CS 1571 Introduction to AI
Lecture 10

First-order logic

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

Solving logical inference problem

In the following:

How to design the procedure that answers:

$$KB \models \alpha \ ?$$

Three approaches:

- **Truth-table approach**
- **Inference rules**
- **Conversion to the inverse SAT problem**
  - **Resolution-refutation**

KBs in the Horn form

Horn clause:

a special type of clause with at most one positive literal

$$(A \vee \neg B) \wedge (\neg A \vee \neg C \vee D)$$

Can be written also as: $(B \Rightarrow A) \wedge ((A \wedge C) \Rightarrow D)$

KB with statements in the Horn form:

- Two types of propositional statements:
  - Implications: called **rules** $(B \Rightarrow A)$
  - Propositional symbols: **facts** B

Modus ponens:

- is the "universal "(complete) rule for the KB with sentences in the Horn form

$$\frac{A \Rightarrow B, \quad A}{B} \qquad \frac{A_1 \wedge A_2 \wedge \ldots \wedge A_k \Rightarrow B, A_1, A_2, \ldots A_k}{B}$$

Forward and backward chaining

Two inference procedures based on **modus ponens** for **Horn KBs**:

- **Forward chaining**

  Idea: Whenever the premises of a rule are satisfied, infer the conclusion. Continue with rules that became satisfied.

- **Backward chaining (goal reduction)**

  Idea: To prove the fact that appears in the conclusion of a rule prove the premises of the rule. Continue recursively.

Both procedures are **complete for KBs in the Horn form** !!!

Forward chaining example

- **Forward chaining**

  Idea: Whenever the premises of a rule are satisfied, infer the conclusion. Continue with rules that became satisfied.

Assume the KB with the following rules and facts:

KB: R1: $A \wedge B \Rightarrow C$

R2: $C \wedge D \Rightarrow E$

R3: $C \wedge F \Rightarrow G$

F1: A
F2: B
F3: D

Theorem: E ?

Forward chaining example

Theorem: E

KB: R1: $A \wedge B \Rightarrow C$

R2: $C \wedge D \Rightarrow E$

R3: $C \wedge F \Rightarrow G$

F1: A
F2: B
F3: D

Rule R1 is satisfied.

F4: C

Forward chaining example

Theorem: E

KB: R1: $A \wedge B \Rightarrow C$

R2: $C \wedge D \Rightarrow E$

R3: $C \wedge F \Rightarrow G$

F1: A

F2: B

F3: D

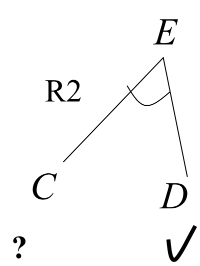
Rule R1 is satisfied.

F4: C

Rule R2 is satisfied.

F5: E ✓

Backward chaining example



KB: R1: $A \wedge B \Rightarrow C$

R2: $C \wedge D \Rightarrow E$

R3: $C \wedge F \Rightarrow G$

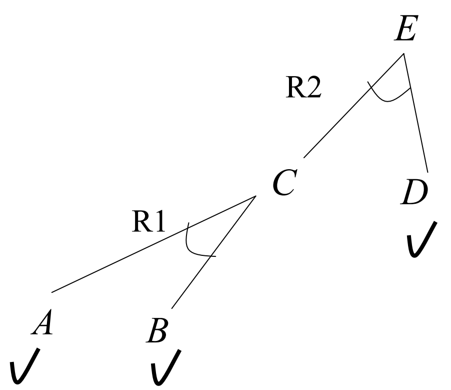
F1: A

F2: B

F3: D

- Backward chaining is more focused:
  - tries to prove the theorem only

Backward chaining example



KB: R1: $A \wedge B \Rightarrow C$

R2: $C \wedge D \Rightarrow E$

R3: $C \wedge F \Rightarrow G$

F1: A

F2: B

F3: D

- Backward chaining is more focused:
  - tries to prove the theorem only

KB agents based on propositional logic

- Propositional logic allows us to build **knowledge-based agents** capable of answering queries about the world by infering new facts from the known ones
- **Example:** an agent for diagnosis of a bacterial disease

Facts: The stain of the organism is gram-positive
The growth conformation of the organism is chains

Rules: **(If)** The stain of the organism is gram-positive $\wedge$
The morphology of the organism is coccus $\wedge$
The growth conformation of the organism is chains
(Then) $\Rightarrow$ The identity of the organism is streptococcus

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

Assume we have: *John is older than Mary*
Mary is older than Paul

To derive *John is older than Paul* we need:

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

Assume we add another fact: *Jane is older than Mary*

To derive *Jane is older than Paul* we need:

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

Problem: KB grows large

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: ??**

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 $\wedge$
Mary likes vacation $\wedge$
Ann likes vacation $\wedge$
…

- **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
- **Predicate logic:** first-order logic without the quantification fix

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 $\exists, \forall$

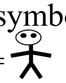
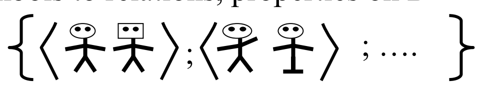
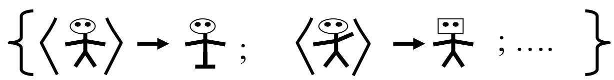
- stand for the **existential** and the **universal** quantifier

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:

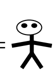
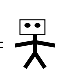
- Constant symbols to objects in D
  I(*John*) = 
- Predicate symbols to relations, properties on D
  I(*brother*) = 
- Function symbols to functional relations on D
  I(*father-of*) = 

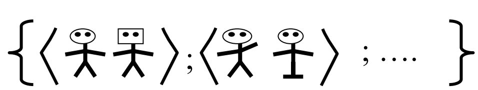
Semantics of sentences.

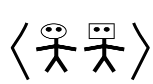
Meaning (evaluation) function:

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

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

- **Quantifications**

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

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

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 implication

- **Existential quantification**

  Someone at CMU is smart

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

Typically the existential quantifier connects with 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

$$\forall x, y \ \ parent\ (x, y) \Rightarrow child\ (y, \mathrm{x})$$

$$\forall y, x \ \ parent\ (x, y) \Rightarrow child\ (y, \mathrm{x})$$

- **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

$$\forall x, y \ \ parent\ (x, y) \Rightarrow child\ (y, \mathrm{x})$$

$$\forall y, x \ \ parent\ (x, y) \Rightarrow child\ (y, \mathrm{x})$$

- **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

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

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

- **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

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

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

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

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

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

- 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 ….