CS 1571 Introduction to AI
Lecture 15

Inference in first-order logic

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

Administrations

- **Midterm 1 - Thursday, October 29, 2009**
  - **In-class** ?
  - **Closed book**

- **Review at the end of the lecture today**

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 ….

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.

Logical inference problem in the Propositional logic

Computational procedures that answer:

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

Three approaches:

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

Inference in FOL: Truth table approach

- Is the Truth-table approach a viable approach for the FOL?
  ?
- NO!
- Why?
- It would require us to enumerate and list all possible interpretations **I**
- I = (assignments of symbols to objects, predicates to relations and functions to relational mappings)
- Simply there are too many interpretations

Inference in FOL: Inference rules

- Is the Inference rule approach a viable approach for the FOL?
  ?
- Yes.
- The inference rules represent sound inference patterns one can apply to sentences in the KB
- What is derived follows from the KB
- Caveat: we need to add rules for handling quantifiers

Inference rules

- **Inference rules from the propositional logic:**
  - Modus ponens

$$\frac{A \Rightarrow B, \quad A}{B}$$

  - Resolution

$$\frac{A \vee B, \quad \neg B \vee C}{A \vee C}$$

  - 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.

$$\exists x \ P(y) \wedge Q(x) \qquad y \text{ is free}$$

Examples:

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

- Bound or free?

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.

    $\exists x \; P(y) \wedge Q(x)$ $\quad y$ is free

Examples:

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

- Bound

$\forall x \, \left(Likes(x, y) \wedge \exists y \; Likes(y, Raymond) \right)$

- Bound or free?

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.

    $\exists x \; P(y) \wedge Q(x)$ $\quad y$ is free

Examples:

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

- Bound

$\forall x \, \left(Likes(x, y) \wedge \exists y \; Likes(y, Raymond) \right)$

- Free

Sentences with variables

First-order logic sentences can include variables.

- **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

    $$\exists x \;\; P(y) \wedge Q(x) \qquad y \text{ is free}$$

  - **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

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

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

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

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

- **Existential elimination.**

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

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

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

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:**

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

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

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

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

Generalized inference rules.

- **Use substitutions that let us make inferences**

Example: Modus Ponens

- **If there exists a substitution σ such that**

$$SUBST(\sigma, A_i) = SUBST(\sigma, A_i') \quad \textbf{for all i=1,2, n}$$

$$\frac{A_1 \wedge A_2 \wedge \ldots A_n \Rightarrow B, \quad A_1', A_2', \ldots A_n'}{SUBST(\sigma, B)}$$

- Substitution that satisfies the generalized inference rule can be build via unification process
- Advantage of the generalized rules: they are focused
  - only substitutions that allow the inferences to proceed

Resolution inference rule

- **Recall:** Resolution inference rule is sound and complete (refutation-complete) for the **propositional logic** and CNF

$$\frac{A \vee B, \quad \neg A \vee C}{B \vee C}$$

- **Generalized resolution rule is sound and refutation complete** for the first-order logic and CNF w/o equalities (if unsatisfiable the resolution will find the contradiction)

$$\sigma = UNIFY(\phi_i, \neg \psi_j) \neq fail$$

$$\frac{\phi_1 \vee \phi_2 \ldots \vee \phi_k, \quad \psi_1 \vee \psi_2 \vee \ldots \psi_n}{SUBST(\sigma, \phi_1 \vee \ldots \vee \phi_{i-1} \vee \phi_{i+1} \ldots \vee \phi_k \vee \psi_1 \vee \ldots \vee \psi_{j-1} \vee \psi_{j+1} \ldots \psi_n)}$$

Example: $\frac{P(x) \vee Q(x), \quad \neg Q(John) \vee S(y)}{?}$

Resolution inference rule

- **Recall:** Resolution inference rule is sound and complete (refutation-complete) for the **propositional logic** and CNF

$$\frac{A \vee B, \quad \neg A \vee C}{B \vee C}$$

- **Generalized resolution rule is sound and refutation complete** for the first-order logic and CNF w/o equalities (if unsatisfiable the resolution will find the contradiction)

$$\sigma = UNIFY\ (\phi_i, \neg \psi_j) \neq fail$$

$$\frac{\phi_1 \vee \phi_2 \ldots \vee \phi_k, \quad \psi_1 \vee \psi_2 \vee \ldots \psi_n}{SUBST(\sigma, \phi_1 \vee \ldots \vee \phi_{i-1} \vee \phi_{i+1} \ldots \vee \phi_k \vee \psi_1 \vee \ldots \vee \psi_{j-1} \vee \psi_{j+1} \ldots \psi_n)}$$

Example: $$\frac{P(x) \vee Q(x), \quad \neg Q(John) \vee S(y)}{P(John) \vee S(y)}$$