CS 1571 Introduction to AI
Lecture 14

Inference in first-order logic

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

Administration announcements

Midterm:

- **Thursday, October 25, 2012**
- **In-class**
- **Closed book**

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

- Is the Truth-table approach a viable approach for the FOL?

  ?

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?

  ?

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 by inference rules 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

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

- 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

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

- 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

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

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

  – 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)**

$$\frac{\phi}{\forall x\ \phi}$$ x – is not free in ϕ

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

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

- **Existential instantiation (introduction)**

$$\frac{\phi(a)}{\exists x \phi(x)}$$ a – is a ground term in ϕ
x – is not free in ϕ

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

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

Unification

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

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

- **Solution: make only 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: Generalized Modus Ponens

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

$$SUBST\ (\sigma, A_i) = SUBST\ (\sigma, A_i')$$ **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: $$\frac{P(x) \vee Q(x), \quad \neg Q(John) \vee S(y)}{P(John) \vee S(y)}$$

Inference with resolution rule

- **Proof by refutation:**
  - Prove that $KB, \neg \alpha$ is **unsatisfiable**
  - resolution is **refutation-complete**

- **Main procedure (steps):**
  1. Convert $KB, \neg \alpha$ to CNF with ground terms and universal variables only
  2. Apply repeatedly the resolution rule while keeping track and consistency of substitutions
  3. Stop when empty set (contradiction) is derived or no more new resolvents (conclusions) follow

Conversion to CNF

1. Eliminate implications, equivalences

$$(p \Rightarrow q) \to (\neg p \vee q)$$

2. Move negations inside (DeMorgan's Laws, double negation)

$$\neg(p \wedge q) \to \neg p \vee \neg q$$
$$\neg(p \vee q) \to \neg p \wedge \neg q$$

$$\neg \forall x\ p \to \exists x\ \neg p$$
$$\neg \exists x\ p \to \forall x\ \neg p$$
$$\neg\neg\ p \to p$$

3. Standardize variables (rename duplicate variables)

$$(\forall x\ P(x)) \vee (\exists x\ Q(x)) \to (\forall x\ P(x)) \vee (\exists y\ Q(y))$$

4. Move all quantifiers left (no invalid capture possible)

$$(\forall x\ P(x)) \vee (\exists y\ Q(y)) \to \forall x\ \exists y\ P(x) \vee Q(y)$$

Conversion to CNF

5. Skolemization (removal of existential quantifiers through elimination)

- If no universal quantifier occurs before the **existential quantifier**, replace the **variable with a new constant symbol** **also called Skolem constant**

$$\exists y\ P(A) \vee Q(y) \to P(A) \vee Q(B)$$

- If a universal quantifier precedes the existential quantifier replace the variable with a function of the "universal" variable

$$\forall x\ \exists y\ P(x) \vee Q(y) \to \forall x\ P(x) \vee Q(F(x))$$

$F(x)$ - **a special function**
- **called Skolem function**

Conversion to CNF

6. Drop universal quantifiers (all variables are universally quantified)

$$\forall x \;\; P(x) \vee Q(F(x)) \rightarrow P(x) \vee Q(F(x))$$

7. Convert to CNF using the distributive laws

$$p \vee (q \wedge r) \rightarrow (p \vee q) \wedge (p \vee r)$$

The result is a CNF with variables, constants, functions

Resolution example

KB $\neg\alpha$

$$\underbrace{\neg P(w) \vee Q(w), \neg Q(y) \vee S(y), \;\; P(x) \vee R(x), \neg R(z) \vee S(z)}_{KB}, \;\; \neg S(A)$$

Resolution example

KB $\neg\alpha$

$\neg P(w) \vee Q(w), \neg Q(y) \vee S(y),\ P(x) \vee R(x), \neg R(z) \vee S(z),\ \neg S(A)$

$\{y / w\}$

$\neg P(w) \vee S(w)$

CS 1571 Intro to AI

M. Hauskrecht

Resolution example

KB $\neg\alpha$

$\neg P(w) \vee Q(w), \neg Q(y) \vee S(y),\ P(x) \vee R(x), \neg R(z) \vee S(z),\ \neg S(A)$

$\{y / w\}$

$\neg P(w) \vee S(w)$

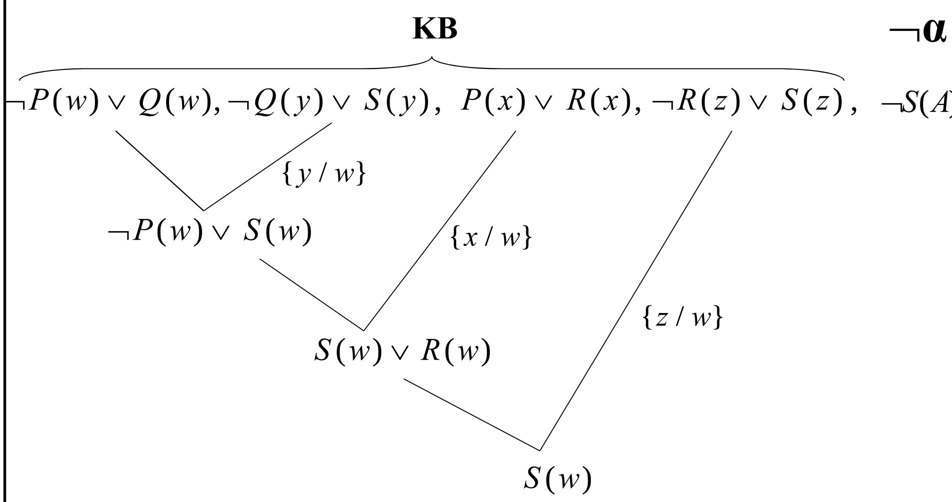
$\{x / w\}$

$S(w) \vee R(w)$

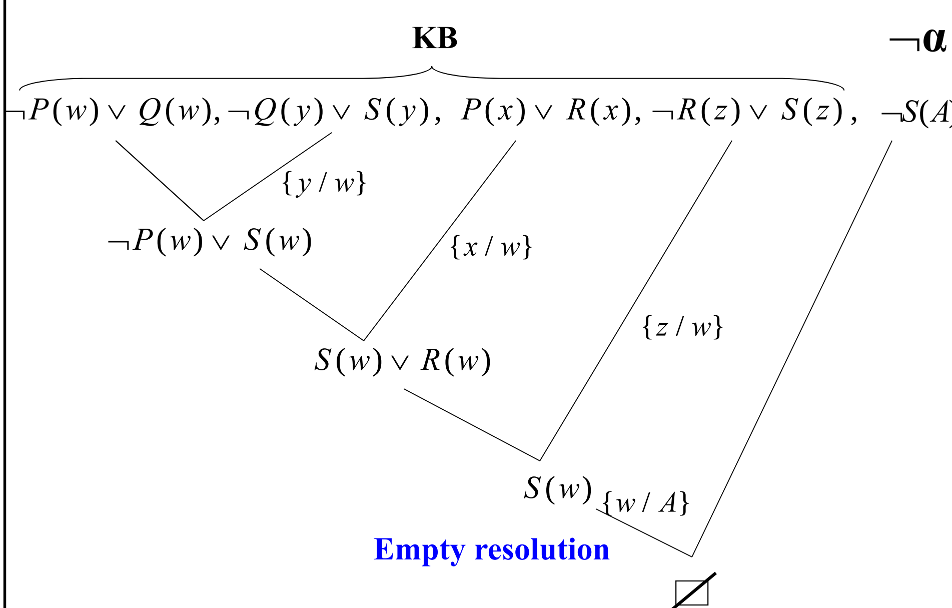
CS 1571 Intro to AI

M. Hauskrecht

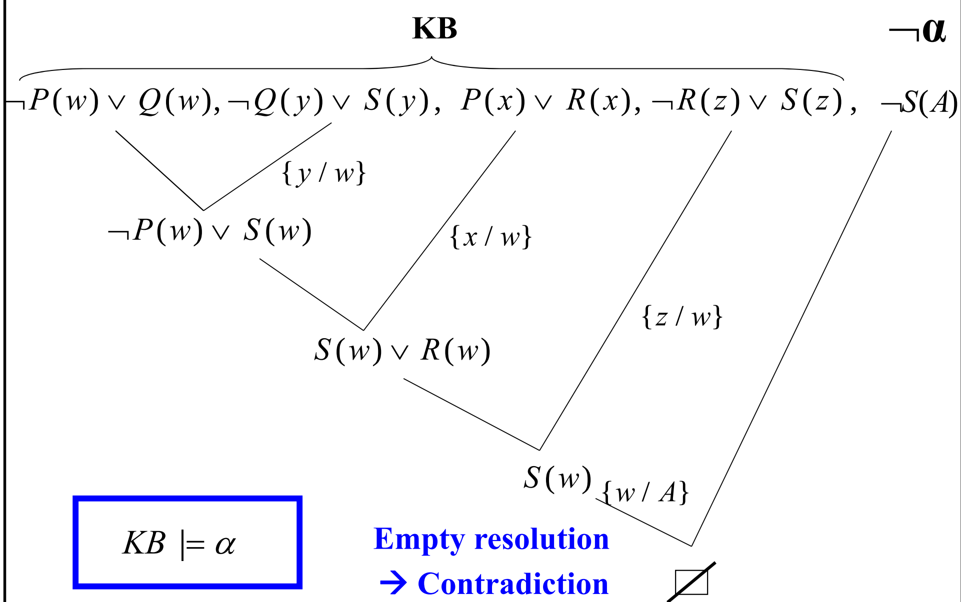
Resolution example



Resolution example



Resolution example



Dealing with equality

- Resolution works for first-order logic without equalities
- To incorporate equalities we need an additional inference rule
- **Demodulation rule**

$\sigma = UNIFY(z_i, t_1) \neq fail$ where z_i occurs in ϕ_i

$$\frac{\phi_1 \vee \phi_2 \ldots \vee \phi_k, \quad t_1 = t_2}{SUB(SUBST(\sigma, t_1), SUBST(\sigma, t_2), \phi_1 \vee \phi_2 \ldots \vee \phi_k)}$$

- **Example:** $$\frac{P(f(a)), f(x) = x}{P(a)}$$
- **Paramodulation rule: more powerful**
- **Resolution+paramodulation give a refutation-complete proof theory for FOL**