First Order Logic - Substitution and Normal Forms

K. Subramani[1]

[1]Lane Department of Computer Science and Electrical Engineering
West Virginia University

13 February, 15 February 2013

Outline

Outline

Substitution

Substitution

Main issue

The purpose of substitution in FOL is the same as in propositional logic, i.e., to facilitate theorem proving.

Substitution

Main issue

The purpose of substitution in FOL is the same as in propositional logic, i.e., to facilitate theorem proving. However, unlike propositional logic, substitution in FOL is complex and requires a lot of care.

Substitution

Main issue

The purpose of substitution in FOL is the same as in propositional logic, i.e., to facilitate theorem proving. However, unlike propositional logic, substitution in FOL is complex and requires a lot of care.

Definition

Renaming the variable x to a fresh variable x' in the fomula $Qx\ F[x]$ produces the equivalent formula $Qx'\ F[x']$, where x' is any variable that does not occur in F.

Substitution

Main issue

The purpose of substitution in FOL is the same as in propositional logic, i.e., to facilitate theorem proving. However, unlike propositional logic, substitution in FOL is complex and requires a lot of care.

Definition

Renaming the variable x to a fresh variable x' in the fomula $Qx\ F[x]$ produces the equivalent formula $Qx'\ F[x']$, where x' is any variable that does not occur in F.

Example

Rename the bound variable x to x' in: $F\ :\ P(x) \wedge (\forall x)\ Q(x, y)$.

Fundamentals

Fundamentals

Definition

Fundamentals

Definition

A substitution is a map from FOL formulae to FOL formulae of the form:

Fundamentals

Definition

A substitution is a map from FOL formulae to FOL formulae of the form:

$$\sigma \;:\; \{F_1 \mapsto G_1, F_2 \mapsto G_2, \ldots F_n \mapsto G_n\}.$$

Fundamentals

Definition

A substitution is a map from FOL formulae to FOL formulae of the form:

$$\sigma \ : \ \{F_1 \mapsto G_1, F_2 \mapsto G_2, \ldots F_n \mapsto G_n\}.$$

The replacements are simultaneous.

Fundamentals

Definition

A substitution is a map from FOL formulae to FOL formulae of the form:

$$\sigma \ : \ \{F_1 \mapsto G_1, F_2 \mapsto G_2, \ldots F_n \mapsto G_n\}.$$

The replacements are simultaneous. If both F_i and F_j are in in the domain of σ and F_i is a strict subformula of F_j, then replace F_j by G_j.

Fundamentals

Definition

A substitution is a map from FOL formulae to FOL formulae of the form:

$$\sigma \ : \ \{F_1 \mapsto G_1, F_2 \mapsto G_2, \ldots F_n \mapsto G_n\}.$$

The replacements are simultaneous. If both F_i and F_j are in in the domain of σ and F_i is a strict subformula of F_j, then replace F_j by G_j.

Example

Fundamentals

Definition

A substitution is a map from FOL formulae to FOL formulae of the form:

$$\sigma \ : \ \{F_1 \mapsto G_1, F_2 \mapsto G_2, \ldots F_n \mapsto G_n\}.$$

The replacements are simultaneous. If both F_i and F_j are in in the domain of σ and F_i is a strict subformula of F_j, then replace F_j by G_j.

Example

Perform the substitution $\sigma \ : \ \{x \mapsto G(x), y \mapsto F(x), Q((F(y), x) \mapsto (\exists x)\ H(x, y)\}$, over the formula $F \ : \ (\forall x)\ P(x, y) \rightarrow Q(F(y), x)$.

Outline

Safe Substitution

Safe Substitution

Issue

There exists a restricted form of substitution called Safe Substitution that preserves formula equivalence.

Safe Substitution

Issue

There exists a restricted form of substitution called Safe Substitution that preserves formula equivalence.

Technique

Safe Substitution

Issue

There exists a restricted form of substitution called Safe Substitution that preserves formula equivalence.

Technique

Consider a substitution σ of the form: $\{F_1 \mapsto G_1, F_2 \mapsto G_2, \ldots F_n \mapsto G_n\}$.

Safe Substitution

Issue

There exists a restricted form of substitution called Safe Substitution that preserves formula equivalence.

Technique

Consider a substitution σ of the form: $\{F_1 \mapsto G_1, F_2 \mapsto G_2, \ldots F_n \mapsto G_n\}$. Let $V_\sigma = \cup_i((\text{free}(F_i) \cup \text{free}(G_i))$.

Safe Substitution

Issue

There exists a restricted form of substitution called Safe Substitution that preserves formula equivalence.

Technique

Consider a substitution σ of the form: $\{F_1 \mapsto G_1, F_2 \mapsto G_2, \ldots F_n \mapsto G_n\}$. Let $V_\sigma = \cup_i((\text{free}(F_i) \cup \text{free}(G_i))$.

Safe Substitution

Issue

There exists a restricted form of substitution called Safe Substitution that preserves formula equivalence.

Technique

Consider a substitution σ of the form: $\{F_1 \mapsto G_1, F_2 \mapsto G_2, \ldots F_n \mapsto G_n\}$. Let $V_\sigma = \cup_i((\text{free}(F_i) \cup \text{free}(G_i))$.

(i) For each quantified variable x in F, such that $x \in V_\sigma$, rename x to a fresh variable x' to produce F'.

Safe Substitution

Issue

There exists a restricted form of substitution called Safe Substitution that preserves formula equivalence.

Technique

Consider a substitution σ of the form: $\{F_1 \mapsto G_1, F_2 \mapsto G_2, \ldots F_n \mapsto G_n\}$. Let $V_\sigma = \cup_i((\mathrm{free}(F_i) \cup \mathrm{free}(G_i))$.

(i) For each quantified variable x in F, such that $x \in V_\sigma$, rename x to a fresh variable x' to produce F'.

(ii) Compute $F'\sigma$.

Safe Substitution

Issue

There exists a restricted form of substitution called Safe Substitution that preserves formula equivalence.

Technique

Consider a substitution σ of the form: $\{F_1 \mapsto G_1, F_2 \mapsto G_2, \ldots F_n \mapsto G_n\}$. Let $V_\sigma = \cup_i((\text{free}(F_i) \cup \text{free}(G_i))$.

(i) For each quantified variable x in F, such that $x \in V_\sigma$, rename x to a fresh variable x' to produce F'.

(ii) Compute $F'\sigma$.

Example

Given $F \; : \; (\forall x)\ P(x, y) \rightarrow Q(F(y), x)$ and $\sigma \; : \; \{x \mapsto G(x), y \mapsto F(x), Q(F(y), x) \mapsto (\exists x)\ H(x, y)\}$, perform the safe substitution $F\sigma$.

Property of Safe Substitutions

Property of Safe Substitutions

Proposition

Consider a substitution $\sigma : \{F_1 \mapsto G_1, F_2 \mapsto G_2, \ldots F_n \mapsto G_n\}$*, such that for each* i*,* $F_i \Leftrightarrow G_i$.

Property of Safe Substitutions

Proposition

Consider a substitution $\sigma : \{F_1 \mapsto G_1, F_2 \mapsto G_2, \ldots F_n \mapsto G_n\}$*, such that for each* i*,* $F_i \Leftrightarrow G_i$*. Then,* $F \Leftrightarrow F\sigma$*, when* $F\sigma$ *is computed in a safe substitution.*

Outline

Schema Substitution

Schema Substitution

Main Idea

Schema Substitution

Main Idea

Rather than painstakingly prove the validity of individual FOL formulae, prove the validity of a general "schema" and obtain the validity of a particular formula, using substitution!

Schema Substitution

Main Idea

Rather than painstakingly prove the validity of individual FOL formulae, prove the validity of a general "schema" and obtain the validity of a particular formula, using substitution! Formula schema and formula substitutions provide the desired generality.

Schema Substitution

Main Idea

Rather than painstakingly prove the validity of individual FOL formulae, prove the validity of a general "schema" and obtain the validity of a particular formula, using substitution! Formula schema and formula substitutions provide the desired generality. Typically, a schema has side conditions, which must be respected during a substitution.

Schema Substitution

Main Idea

Rather than painstakingly prove the validity of individual FOL formulae, prove the validity of a general “schema” and obtain the validity of a particular formula, using substitution! Formula schema and formula substitutions provide the desired generality. Typically, a schema has side conditions, which must be respected during a substitution. For instance, suppose you want to prove the validity of
$G\ :\ (\forall x)(\exists y)Q(x,y) \rightarrow (\neg(\exists x)\neg(\exists y)Q(x,y))$.

Schema Substitution

Main Idea

Rather than painstakingly prove the validity of individual FOL formulae, prove the validity of a general "schema" and obtain the validity of a particular formula, using substitution! Formula schema and formula substitutions provide the desired generality. Typically, a schema has side conditions, which must be respected during a substitution. For instance, suppose you want to prove the validity of $G : (\forall x)(\exists y)Q(x,y) \rightarrow (\neg(\exists x)\neg(\exists y)Q(x,y))$. You first prove the validity of the schema:

Schema Substitution

Main Idea

Rather than painstakingly prove the validity of individual FOL formulae, prove the validity of a general "schema" and obtain the validity of a particular formula, using substitution! Formula schema and formula substitutions provide the desired generality. Typically, a schema has side conditions, which must be respected during a substitution. For instance, suppose you want to prove the validity of $G\ :\ (\forall x)(\exists y)Q(x,y) \rightarrow (\neg(\exists x)\neg(\exists y)Q(x,y))$. You first prove the validity of the schema: $H\ :\ (\forall x)\ F \rightarrow (\neg(\exists x)\ \neg F)$.

Schema Substitution

Main Idea

Rather than painstakingly prove the validity of individual FOL formulae, prove the validity of a general "schema" and obtain the validity of a particular formula, using substitution! Formula schema and formula substitutions provide the desired generality. Typically, a schema has side conditions, which must be respected during a substitution. For instance, suppose you want to prove the validity of $G : (\forall x)(\exists y)Q(x,y) \rightarrow (\neg(\exists x)\neg(\exists y)Q(x,y))$. You first prove the validity of the schema: $H : (\forall x)\ F \rightarrow (\neg(\exists x)\ \neg F)$. Then use the substitution $\sigma : \{F \mapsto (\exists y)Q(x,y)\}$.

Schema Substitution

Main Idea

Rather than painstakingly prove the validity of individual FOL formulae, prove the validity of a general "schema" and obtain the validity of a particular formula, using substitution! Formula schema and formula substitutions provide the desired generality. Typically, a schema has side conditions, which must be respected during a substitution. For instance, suppose you want to prove the validity of $G \ : \ (\forall x)(\exists y)Q(x,y) \rightarrow (\neg(\exists x)\neg(\exists y)Q(x,y))$. You first prove the validity of the schema: $H \ : \ (\forall x)\ F \rightarrow (\neg(\exists x)\ \neg F)$. Then use the substitution $\sigma \ : \ \{F \mapsto (\exists y)Q(x,y)\}$.

A formula schema H is said to be **valid**, if $H\sigma$ is valid, for every schema substitution σ that respects the side conditions of H.

Schema Substitution

Main Idea

Rather than painstakingly prove the validity of individual FOL formulae, prove the validity of a general "schema" and obtain the validity of a particular formula, using substitution! Formula schema and formula substitutions provide the desired generality. Typically, a schema has side conditions, which must be respected during a substitution. For instance, suppose you want to prove the validity of $G\ :\ (\forall x)(\exists y)Q(x,y) \rightarrow (\neg(\exists x)\neg(\exists y)Q(x,y))$. You first prove the validity of the schema: $H\ :\ (\forall x)\ F \rightarrow (\neg(\exists x)\ \neg F)$. Then use the substitution $\sigma\ :\ \{F \mapsto (\exists y)Q(x,y)\}$.

A formula schema H is said to be **valid**, if $H\sigma$ is valid, for every schema substitution σ that respects the side conditions of H. The key point is that the validity of the schema itself can be proved by the semantic argument method.

Example of proving schema validity

Example of proving schema validity

Example

Prove the validity of

Example of proving schema validity

Example

Prove the validity of

$$H \ : \ (\forall x)\ (F_1 \wedge F_2) \rightarrow (\forall x)\ F_1 \wedge F_2, \ \text{provided } x \notin \text{free}(F_2)$$

Application of schema substitution

Application of schema substitution

Proposition

If H is a valid formula schema and σ is a substitution that respects H's side conditions, then $H\sigma$ is also valid.

Normal Forms

Normal Forms

Types

Normal Forms

Types

Normal Forms

Types

(i) NNF.

Normal Forms

Types

(i) NNF.
(ii) PNF.

Normal Forms

Types

(i) NNF.
(ii) PNF.
(iii) CNF.

Normal Forms

Types

(i) NNF.
(ii) PNF.
(iii) CNF.
(iv) DNF.

NNF

NNF

Technique

NNF

Technique

NNF

Technique

(i) $\neg(\forall x)\ F[x] \Leftrightarrow (\exists x)\ \neg F[x]$.

NNF

Technique

(i) $\neg(\forall x)\ F[x] \Leftrightarrow (\exists x)\ \neg F[x]$.

(ii) $\neg(\exists x)\ F[x] \Leftrightarrow (\forall x)\ \neg F[x]$.

NNF

Technique

(i) $\neg(\forall x)\ F[x] \Leftrightarrow (\exists x)\ \neg F[x]$.

(ii) $\neg(\exists x)\ F[x] \Leftrightarrow (\forall x)\ \neg F[x]$.

Example

Let $G\ :\ (\forall x)((\exists y)\ P(x,y) \wedge P(x,z)) \rightarrow (\exists w)\ P(x,w)$.

NNF

Technique

(i) $\neg(\forall x)\ F[x] \Leftrightarrow (\exists x)\ \neg F[x]$.

(ii) $\neg(\exists x)\ F[x] \Leftrightarrow (\forall x)\ \neg F[x]$.

Example

Let $G\ :\ (\forall x)((\exists y)\ P(x,y) \wedge P(x,z)) \rightarrow (\exists w)\ P(x,w)$. Put G in NNF.

PNF

PNF

Definition

A formula is in Prenex Normal Form (PNF), if all of its quantifiers appear at the beginning of the formula

PNF

Definition

A formula is in Prenex Normal Form (PNF), if all of its quantifiers appear at the beginning of the formula (e.g. $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \; F[x_1, x_2, \ldots, x_n]$).

PNF

Definition

A formula is in Prenex Normal Form (PNF), if all of its quantifiers appear at the beginning of the formula (e.g. $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \; F[x_1, x_2, \ldots, x_n]$).

Conversion technique

PNF

Definition

A formula is in Prenex Normal Form (PNF), if all of its quantifiers appear at the beginning of the formula (e.g. $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \ F[x_1, x_2, \ldots, x_n]$).

Conversion technique

(i) Convert F into NNF formula F_1.

PNF

Definition

A formula is in Prenex Normal Form (PNF), if all of its quantifiers appear at the beginning of the formula (e.g. $Q_1x_1Q_2x_2 \ldots Q_nx_n\ F[x_1, x_2, \ldots, x_n]$).

Conversion technique

(i) Convert F into NNF formula F_1.

(ii) Use renaming with fresh variables to ensure that each quantified variable has a different name.

PNF

Definition

A formula is in Prenex Normal Form (PNF), if all of its quantifiers appear at the beginning of the formula (e.g. $Q_1x_1Q_2x_2 \ldots Q_nx_n \; F[x_1, x_2, \ldots, x_n]$).

Conversion technique

(i) Convert F into NNF formula F_1.

(ii) Use renaming with fresh variables to ensure that each quantified variable has a different name. Call the resultant formula F_2.

PNF

Definition

A formula is in Prenex Normal Form (PNF), if all of its quantifiers appear at the beginning of the formula (e.g. $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \; F[x_1, x_2, \ldots, x_n]$).

Conversion technique

(i) Convert F into NNF formula F_1.

(ii) Use renaming with fresh variables to ensure that each quantified variable has a different name. Call the resultant formula F_2.

(iii) Remove all quantifiers from F_2 to produce a quantifier-free formula F_3.

PNF

Definition

A formula is in Prenex Normal Form (PNF), if all of its quantifiers appear at the beginning of the formula (e.g. $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \; F[x_1, x_2, \ldots, x_n]$).

Conversion technique

(i) Convert F into NNF formula F_1.

(ii) Use renaming with fresh variables to ensure that each quantified variable has a different name. Call the resultant formula F_2.

(iii) Remove all quantifiers from F_2 to produce a quantifier-free formula F_3.

(iv) Add the quantifiers before F_3 to get a formula F_4, respecting scope rules.

PNF

Definition

A formula is in Prenex Normal Form (PNF), if all of its quantifiers appear at the beginning of the formula (e.g. $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \; F[x_1, x_2, \ldots, x_n]$).

Conversion technique

(i) Convert F into NNF formula F_1.

(ii) Use renaming with fresh variables to ensure that each quantified variable has a different name. Call the resultant formula F_2.

(iii) Remove all quantifiers from F_2 to produce a quantifier-free formula F_3.

(iv) Add the quantifiers before F_3 to get a formula F_4, respecting scope rules. Let F_4 be $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n F_3$. If Q_j is in the scope of Q_i in F_1, then we must have $i < j$.

PNF

Definition

A formula is in Prenex Normal Form (PNF), if all of its quantifiers appear at the beginning of the formula (e.g. $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \; F[x_1, x_2, \ldots, x_n]$).

Conversion technique

(i) Convert F into NNF formula F_1.

(ii) Use renaming with fresh variables to ensure that each quantified variable has a different name. Call the resultant formula F_2.

(iii) Remove all quantifiers from F_2 to produce a quantifier-free formula F_3.

(iv) Add the quantifiers before F_3 to get a formula F_4, respecting scope rules. Let F_4 be $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n F_3$. If Q_j is in the scope of Q_i in F_1, then we must have $i < j$.

Example

Convert the following formula to PNF:

$$F \; : \; (\forall x) \; [\neg((\exists y) \; (P(x, y) \wedge P(x, z))) \vee (\exists y) \; P(x, y)].$$

CNF and DNF

CNF and DNF

Definition

CNF and DNF

Definition

A formula is said to be in CNF (respectively DNF), if it is in PNF, and its quantifier-free subformula is in CNF (respectively DNF).