Logic 1

First-Order Logic

Mădălina Erașcu   Tudor Jebelean

Research Institute for Symbolic Computation,
Johannes Kepler University, Linz, Austria

{merascu,tjebelea}@risc.jku.at

November 21, 2013

Outline

Syntax

Semantics

(Un)Satisfiability & (In)Validity

Equivalences of Formulas

Normal Forms

Formula Clausification

Substitution

Outline

Syntax

The language of FOL consists in terms and formulas.

Terms are defined recursively as follows:

1. A constant is a term.
2. A variable is a term.
3. If f is an n-place function symbol, and t_1, ..., t_n are terms then $f[t_1, ..., t_n]$ is a term.
4. All terms are generated by applying the above rules.

If P is an n-place predicate symbol and t_1, ..., t_n are terms then $P[t_1, ..., t_n]$ is an atom.

An atom is $\mathbb{T}$, $\mathbb{F}$, or an n-ary predicate applied to n terms.

A literal is an atom or its negation.

Syntax

The language of FOL consists in terms and formulas.

Terms are defined recursively as follows:

1. A constant is a term.
2. A variable is a term.
3. If f is an n-place function symbol, and t_1, ..., t_n are terms then $f[t_1, ..., t_n]$ is a term.
4. All terms are generated by applying the above rules.

If P is an n-place predicate symbol and t_1, ..., t_n are terms then $P[t_1, ..., t_n]$ is an atom.

An atom is $\mathbb{T}$, $\mathbb{F}$, or an n-ary predicate applied to n terms.

A literal is an atom or its negation.

Syntax

The language of FOL consists in terms and formulas.

Terms are defined recursively as follows:

1. A constant is a term.
2. A variable is a term.
3. If f is an n-place function symbol, and t_1, ..., t_n are terms then $f[t_1, ..., t_n]$ is a term.
4. All terms are generated by applying the above rules.

If P is an n-place predicate symbol and t_1, ..., t_n are terms then $P[t_1, ..., t_n]$ is an atom.

An atom is $\mathbb{T}$, $\mathbb{F}$, or an n-ary predicate applied to n terms.

A literal is an atom or its negation.

Syntax

The language of FOL consists in terms and formulas.

Terms are defined recursively as follows:

1. A constant is a term.
2. A variable is a term.
3. If f is an n-place function symbol, and t_1, ..., t_n are terms then $f[t_1, ..., t_n]$ is a term.
4. All terms are generated by applying the above rules.

If P is an n-place predicate symbol and t_1, ..., t_n are terms then $P[t_1, ..., t_n]$ is an atom.

An atom is $\mathbb{T}$, $\mathbb{F}$, or an n-ary predicate applied to n terms.

A literal is an atom or its negation.

Syntax

The language of FOL consists in terms and formulas.

Terms are defined recursively as follows:

1. A constant is a term.
2. A variable is a term.
3. If f is an n-place function symbol, and t_1, ..., t_n are terms then $f[t_1, ..., t_n]$ is a term.
4. All terms are generated by applying the above rules.

If P is an n-place predicate symbol and t_1, ..., t_n are terms then $P[t_1, ..., t_n]$ is an atom.

An atom is $\mathbb{T}$, $\mathbb{F}$, or an n-ary predicate applied to n terms.

A literal is an atom or its negation.

Syntax

The language of FOL consists in terms and formulas.

Terms are defined recursively as follows:

1. A constant is a term.
2. A variable is a term.
3. If f is an n-place function symbol, and t_1, ..., t_n are terms then $f[t_1, ..., t_n]$ is a term.
4. All terms are generated by applying the above rules.

If P is an n-place predicate symbol and t_1, ..., t_n are terms then $P[t_1, ..., t_n]$ is an atom.

An atom is $\mathbb{T}$, $\mathbb{F}$, or an n-ary predicate applied to n terms.

A literal is an atom or its negation.

Syntax

The language of FOL consists in terms and formulas.

Terms are defined recursively as follows:

1. A constant is a term.
2. A variable is a term.
3. If f is an n-place function symbol, and t_1, ..., t_n are terms then $f[t_1, ..., t_n]$ is a term.
4. All terms are generated by applying the above rules.

If P is an n-place predicate symbol and t_1, ..., t_n are terms then $P[t_1, ..., t_n]$ is an atom.

An atom is $\mathbb{T}$, $\mathbb{F}$, or an n-ary predicate applied to n terms.

A literal is an atom or its negation.

Syntax

The language of FOL consists in terms and formulas.

Terms are defined recursively as follows:

1. A constant is a term.
2. A variable is a term.
3. If f is an n-place function symbol, and t_1, ..., t_n are terms then $f[t_1, ..., t_n]$ is a term.
4. All terms are generated by applying the above rules.

If P is an n-place predicate symbol and t_1, ..., t_n are terms then $P[t_1, ..., t_n]$ is an atom.

An atom is $\mathbb{T}$, $\mathbb{F}$, or an n-ary predicate applied to n terms.

A literal is an atom or its negation.

Syntax

The language of FOL consists in terms and formulas.

Terms are defined recursively as follows:

1. A constant is a term.
2. A variable is a term.
3. If f is an n-place function symbol, and t_1, ..., t_n are terms then $f[t_1, ..., t_n]$ is a term.
4. All terms are generated by applying the above rules.

If P is an n-place predicate symbol and t_1, ..., t_n are terms then $P[t_1, ..., t_n]$ is an atom.

An atom is $\mathbb{T}$, $\mathbb{F}$, or an n-ary predicate applied to n terms.

A literal is an atom or its negation.

Syntax (cont'd)

Formulas are defined as follows:

1. An atom is a formula.
2. If F and G are formulas then $\neg F$, $F \vee G$, $F \wedge G$, $F \Longrightarrow G$, and $F \iff G$ are formulas.
3. If F is a formula and x is a variable, then $\underset{x}{\forall} F$ and $\underset{x}{\exists} F$ are formulas.
4. Formulas are generated only by a finite number of applications of the above rules.

A variable x is bound in the formula F if there is an occurrence of x in the scope of a binding quantifier $\underset{x}{\forall}$ or $\underset{x}{\exists}$.

A variable x is free in the formula F if there is an occurrence of x that is not bound by any quantifier.

Examples: Identify constants, variables (free, bound), quantifiers, functions, predicates, atoms, terms, formulas from the bellow

1. $\underset{x}{\forall}\, x + 1 \geq x$
2. $\neg \left(\underset{x}{\exists}\, E[0, f[x]] \right)$
3. $\underset{x}{\forall} \underset{y}{\exists} \Big(E[y, f[x]] \;\wedge\; \underset{z}{\forall} \big(E[z, f[x]] \Rightarrow E[y, z] \big) \Big)$

Syntax (cont'd)

Formulas are defined as follows:

1. An atom is a formula.
2. If F and G are formulas then $\neg F$, $F \vee G$, $F \wedge G$, $F \Longrightarrow G$, and $F \iff G$ are formulas.
3. If F is a formula and x is a variable, then $\underset{x}{\forall} F$ and $\underset{x}{\exists} F$ are formulas.
4. Formulas are generated only by a finite number of applications of the above rules.

A variable x is bound in the formula F if there is an occurrence of x in the scope of a binding quantifier $\underset{x}{\forall}$ or $\underset{x}{\exists}$.

A variable x is free in the formula F if there is an occurrence of x that is not bound by any quantifier.

Examples: Identify constants, variables (free, bound), quantifiers, functions, predicates, atoms, terms, formulas from the bellow

1. $\underset{x}{\forall}\, x + 1 \geq x$
2. $\neg \left(\underset{x}{\exists}\, E[0, f[x]] \right)$
3. $\underset{x}{\forall} \underset{y}{\exists} \left(E[y, f[x]] \;\wedge\; \underset{z}{\forall} (E[z, f[x]] \Rightarrow E[y, z]) \right)$

Syntax (cont'd)

Formulas are defined as follows:

1. An atom is a formula.
2. If F and G are formulas then $\neg F$, $F \vee G$, $F \wedge G$, $F \implies G$, and $F \iff G$ are formulas.
3. If F is a formula and x is a variable, then $\underset{x}{\forall} F$ and $\underset{x}{\exists} F$ are formulas.
4. Formulas are generated only by a finite number of applications of the above rules.

A variable x is bound in the formula F if there is an occurrence of x in the scope of a binding quantifier $\underset{x}{\forall}$ or $\underset{x}{\exists}$.

A variable x is free in the formula F if there is an occurrence of x that is not bound by any quantifier.

Examples: Identify constants, variables (free, bound), quantifiers, functions, predicates, atoms, terms, formulas from the bellow

1. $\underset{x}{\forall}\ x + 1 \geq x$
2. $\neg\left(\underset{x}{\exists}\ E[0, f[x]]\right)$
3. $\underset{x}{\forall}\underset{y}{\exists}\left(E[y, f[x]] \ \wedge\ \underset{z}{\forall}\left(E[z, f[x]] \Rightarrow E[y, z]\right)\right)$

Syntax (cont'd)

Formulas are defined as follows:

1. An atom is a formula.
2. If F and G are formulas then $\neg F$, $F \vee G$, $F \wedge G$, $F \implies G$, and $F \iff G$ are formulas.
3. If F is a formula and x is a variable, then $\underset{x}{\forall} F$ and $\underset{x}{\exists} F$ are formulas.
4. Formulas are generated only by a finite number of applications of the above rules.

A variable x is bound in the formula F if there is an occurrence of x in the scope of a binding quantifier $\underset{x}{\forall}$ or $\underset{x}{\exists}$.

A variable x is free in the formula F if there is an occurrence of x that is not bound by any quantifier.

Examples: Identify constants, variables (free, bound), quantifiers, functions, predicates, atoms, terms, formulas from the bellow

1. $\underset{x}{\forall}\, x+1 \geq x$
2. $\neg\left(\underset{x}{\exists}\, E[0, f[x]]\right)$
3. $\underset{x}{\forall}\,\underset{y}{\exists}\left(E[y, f[x]] \;\wedge\; \underset{z}{\forall}\left(E[z, f[x]] \Rightarrow E[y, z]\right)\right)$

Syntax (cont'd)

Formulas are defined as follows:

1. An atom is a formula.
2. If F and G are formulas then $\neg F$, $F \vee G$, $F \wedge G$, $F \implies G$, and $F \iff G$ are formulas.
3. If F is a formula and x is a variable, then $\underset{x}{\forall} F$ and $\underset{x}{\exists} F$ are formulas.
4. Formulas are generated only by a finite number of applications of the above rules.

A variable x is bound in the formula F if there is an occurrence of x in the scope of a binding quantifier $\underset{x}{\forall}$ or $\underset{x}{\exists}$.

A variable x is free in the formula F if there is an occurrence of x that is not bound by any quantifier.

Examples: Identify constants, variables (free, bound), quantifiers, functions, predicates, atoms, terms, formulas from the bellow

1. $\underset{x}{\forall}\ x + 1 \geq x$
2. $\neg \left(\underset{x}{\exists}\ E[0, f[x]] \right)$
3. $\underset{x}{\forall} \underset{y}{\exists} \Big(E[y, f[x]] \ \wedge\ \underset{z}{\forall} \big(E[z, f[x]] \Rightarrow E[y, z] \big) \Big)$

Syntax (cont'd)

Formulas are defined as follows:

1. An atom is a formula.
2. If F and G are formulas then $\neg F$, $F \vee G$, $F \wedge G$, $F \Longrightarrow G$, and $F \iff G$ are formulas.
3. If F is a formula and x is a variable, then $\underset{x}{\forall} F$ and $\underset{x}{\exists} F$ are formulas.
4. Formulas are generated only by a finite number of applications of the above rules.

A variable x is bound in the formula F if there is an occurrence of x in the scope of a binding quantifier $\underset{x}{\forall}$ or $\underset{x}{\exists}$.

A variable x is free in the formula F if there is an occurrence of x that is not bound by any quantifier.

Examples: Identify constants, variables (free, bound), quantifiers, functions, predicates, atoms, terms, formulas from the bellow

1. $\underset{x}{\forall}\ x + 1 \geq x$
2. $\neg\left(\underset{x}{\exists}\ E[0, f[x]]\right)$
3. $\underset{x}{\forall}\underset{y}{\exists}\left(E[y, f[x]]\ \wedge\ \underset{z}{\forall}(E[z, f[x]] \Rightarrow E[y, z])\right)$

Syntax (cont'd)

Formulas are defined as follows:

1. An atom is a formula.
2. If F and G are formulas then $\neg F$, $F \vee G$, $F \wedge G$, $F \Longrightarrow G$, and $F \iff G$ are formulas.
3. If F is a formula and x is a variable, then $\underset{x}{\forall} F$ and $\underset{x}{\exists} F$ are formulas.
4. Formulas are generated only by a finite number of applications of the above rules.

A variable x is bound in the formula F if there is an occurrence of x in the scope of a binding quantifier $\underset{x}{\forall}$ or $\underset{x}{\exists}$.

A variable x is free in the formula F if there is an occurrence of x that is not bound by any quantifier.

Examples: Identify constants, variables (free, bound), quantifiers, functions, predicates, atoms, terms, formulas from the bellow

1. $\underset{x}{\forall}\ x + 1 \geq x$
2. $\neg \left(\underset{x}{\exists}\ E[0, f[x]] \right)$
3. $\underset{x}{\forall} \underset{y}{\exists} \left(E[y, f[x]] \ \wedge \ \underset{z}{\forall} \left(E[z, f[x]] \Rightarrow E[y, z] \right) \right)$

Syntax (cont'd)

Formulas are defined as follows:

1. An atom is a formula.
2. If F and G are formulas then $\neg F$, $F \vee G$, $F \wedge G$, $F \implies G$, and $F \iff G$ are formulas.
3. If F is a formula and x is a variable, then $\underset{x}{\forall} F$ and $\underset{x}{\exists} F$ are formulas.
4. Formulas are generated only by a finite number of applications of the above rules.

A variable x is bound in the formula F if there is an occurrence of x in the scope of a binding quantifier $\underset{x}{\forall}$ or $\underset{x}{\exists}$.

A variable x is free in the formula F if there is an occurrence of x that is not bound by any quantifier.

Examples: Identify constants, variables (free, bound), quantifiers, functions, predicates, atoms, terms, formulas from the bellow

1. $\underset{x}{\forall}\, x + 1 \geq x$
2. $\neg \left(\underset{x}{\exists}\, E[0, f[x]] \right)$
3. $\underset{x}{\forall}\, \underset{y}{\exists} \left(E[y, f[x]] \;\wedge\; \underset{z}{\forall} (E[z, f[x]] \Rightarrow E[y, z]) \right)$

Syntax (cont'd)

Formulas are defined as follows:

1. An atom is a formula.
2. If F and G are formulas then $\neg F$, $F \vee G$, $F \wedge G$, $F \implies G$, and $F \iff G$ are formulas.
3. If F is a formula and x is a variable, then $\underset{x}{\forall} F$ and $\underset{x}{\exists} F$ are formulas.
4. Formulas are generated only by a finite number of applications of the above rules.

A variable x is bound in the formula F if there is an occurrence of x in the scope of a binding quantifier $\underset{x}{\forall}$ or $\underset{x}{\exists}$.

A variable x is free in the formula F if there is an occurrence of x that is not bound by any quantifier.

Examples: Identify constants, variables (free, bound), quantifiers, functions, predicates, atoms, terms, formulas from the bellow

1. $\underset{x}{\forall}\; x + 1 \geq x$
2. $\neg \left(\underset{x}{\exists}\; E[0, f[x]] \right)$
3. $\underset{x}{\forall} \underset{y}{\exists} \left(E[y, f[x]] \;\wedge\; \underset{z}{\forall} (E[z, f[x]] \Rightarrow E[y, z]) \right)$

Syntax (cont'd)

Formulas are defined as follows:

1. An atom is a formula.
2. If F and G are formulas then $\neg F$, $F \vee G$, $F \wedge G$, $F \Longrightarrow G$, and $F \iff G$ are formulas.
3. If F is a formula and x is a variable, then $\underset{x}{\forall} F$ and $\underset{x}{\exists} F$ are formulas.
4. Formulas are generated only by a finite number of applications of the above rules.

A variable x is bound in the formula F if there is an occurrence of x in the scope of a binding quantifier $\underset{x}{\forall}$ or $\underset{x}{\exists}$.

A variable x is free in the formula F if there is an occurrence of x that is not bound by any quantifier.

Examples: Identify constants, variables (free, bound), quantifiers, functions, predicates, atoms, terms, formulas from the bellow

1. $\underset{x}{\forall}\ x + 1 \geq x$
2. $\neg \left(\underset{x}{\exists}\ E[0, f[x]] \right)$
3. $\underset{x}{\forall} \underset{y}{\exists} \left(E[y, f[x]] \ \wedge \ \underset{z}{\forall} (E[z, f[x]] \Rightarrow E[y, z]) \right)$

Syntax (cont'd)

Formulas are defined as follows:

1. An atom is a formula.
2. If F and G are formulas then $\neg F$, $F \vee G$, $F \wedge G$, $F \implies G$, and $F \iff G$ are formulas.
3. If F is a formula and x is a variable, then $\underset{x}{\forall} F$ and $\underset{x}{\exists} F$ are formulas.
4. Formulas are generated only by a finite number of applications of the above rules.

A variable x is bound in the formula F if there is an occurrence of x in the scope of a binding quantifier $\underset{x}{\forall}$ or $\underset{x}{\exists}$.

A variable x is free in the formula F if there is an occurrence of x that is not bound by any quantifier.

Examples: Identify constants, variables (free, bound), quantifiers, functions, predicates, atoms, terms, formulas from the bellow

1. $\underset{x}{\forall}\; x + 1 \geq x$
2. $\neg \left(\underset{x}{\exists}\; E[0, f[x]] \right)$
3. $\underset{x}{\forall}\, \underset{y}{\exists} \Big(E[y, f[x]] \;\wedge\; \underset{z}{\forall} (E[z, f[x]] \Rightarrow E[y, z]) \Big)$

Outline

- Syntax
- **Semantics**
- (Un)Satisfiability & (In)Validity
- Equivalences of Formulas
- Normal Forms
- Formula Clausification
- Substitution

Semantics

An interpretation I of a formula F in FOL consists of a nonempty domain D and an assignment of values to each constant, function, symbol and predicate symbol occurring in F as follows:

- to each constant we assign an element in D
- to each function symbol we assign a mapping from D^n to D
- to each predicate symbol we assign a mapping from D^n to $\{\mathbb{T}, \mathbb{F}\}$.

Then the semantics of the formula F is a function $f : \mathcal{I} \to \{\mathbb{T}, \mathbb{F}\}$, where $I \in \mathcal{I}$ and $\mathcal{I}$ is the set of all interpretations of the formula F.

Semantics

An interpretation I of a formula F in FOL consists of a nonempty domain D and an assignment of values to each constant, function, symbol and predicate symbol occurring in F as follows:

- to each constant we assign an element in D
- to each function symbol we assign a mapping from D^n to D
- to each predicate symbol we assign a mapping from D^n to $\{\mathbb{T}, \mathbb{F}\}$.

Then the semantics of the formula F is a function $f : \mathcal{I} \to \{\mathbb{T}, \mathbb{F}\}$, where $I \in \mathcal{I}$ and $\mathcal{I}$ is the set of all interpretations of the formula F.

Semantics

An interpretation I of a formula F in FOL consists of a nonempty domain D and an assignment of values to each constant, function, symbol and predicate symbol occurring in F as follows:

- to each constant we assign an element in D
- to each function symbol we assign a mapping from D^n to D
- to each predicate symbol we assign a mapping from D^n to $\{\mathbb{T}, \mathbb{F}\}$.

Then the semantics of the formula F is a function $f : \mathcal{I} \to \{\mathbb{T}, \mathbb{F}\}$, where $I \in \mathcal{I}$ and $\mathcal{I}$ is the set of all interpretations of the formula F.

Semantics

An interpretation I of a formula F in FOL consists of a nonempty domain D and an assignment of values to each constant, function, symbol and predicate symbol occurring in F as follows:

- to each constant we assign an element in D
- to each function symbol we assign a mapping from D^n to D
- to each predicate symbol we assign a mapping from D^n to $\{\mathbb{T}, \mathbb{F}\}$.

Then the semantics of the formula F is a function $f : \mathcal{I} \to \{\mathbb{T}, \mathbb{F}\}$, where $I \in \mathcal{I}$ and $\mathcal{I}$ is the set of all interpretations of the formula F.

Semantics

An interpretation I of a formula F in FOL consists of a nonempty domain D and an assignment of values to each constant, function, symbol and predicate symbol occurring in F as follows:

- to each constant we assign an element in D
- to each function symbol we assign a mapping from D^n to D
- to each predicate symbol we assign a mapping from D^n to $\{\mathbb{T}, \mathbb{F}\}$.

Then the semantics of the formula F is a function $f : \mathcal{I} \to \{\mathbb{T}, \mathbb{F}\}$, where $I \in \mathcal{I}$ and $\mathcal{I}$ is the set of all interpretations of the formula F.

Semantics (cont'd)

Example: Find the truth value of the formulas:

- $F_1 :\iff \underset{x}{\forall}\,\underset{y}{\forall}\, x \leq y,$ where $I : \begin{cases} D = \{0, 1\} \\ \leq_I \;\rightarrow\; \leq_{\mathbb{Z}} \end{cases}$
- $F_2 :\iff \underset{x}{\forall}\,\underset{y}{\exists}\, x + y > c,$ where $I : \begin{cases} D = \{0, 1\} \\ c_I = 0 \\ +_I \;\rightarrow\; +_{\mathbb{Z}} \\ >_I \;\rightarrow\; >_{\mathbb{Z}} \end{cases}$
- $F_3 :\iff \underset{x}{\forall}\, (P[x] \Longrightarrow Q[f[x], a])$, where

$$I : \begin{cases} D = \{1, 2\} \\ a_I = 1 \\ f_I : D \rightarrow D & \begin{cases} f_I[1] = 1 \\ f_I[2] = 1 \end{cases} \\ P_I : D \rightarrow \{\mathbb{T}, \mathbb{F}\} & \begin{cases} P_I[1] = \mathbb{T} \\ P_I[2] = \mathbb{F} \end{cases} \\ Q_I : D^2 \rightarrow \{\mathbb{T}, \mathbb{F}\} & \begin{cases} Q_I[1,1] = \mathbb{T} & Q_I[1,2] = \mathbb{F} \\ Q_I[2,1] = \mathbb{F} & Q_I[2,2] = \mathbb{T} \end{cases} \end{cases}$$

Outline

(Un)Satisfiability & (In)Validity

A formula F is satisfiable (consistent) iff there exists an interpretation I such that F is evaluated to $\mathbb{T}$ in I.

A formula F is unsatisfiable (inconsistent) iff for all interpretations I, F is evaluated to $\mathbb{F}$ in I.

A formula F is valid iff for all interpretations I, F is evaluated to $\mathbb{T}$ in I.

A formula F is invalid iff there exists an interpretation I, such that F is evaluated to $\mathbb{F}$ in I.

A formula G is a logical consequence of formulas F_1, F_2, ..., F_n iff for every interpretation I, if $F_1 \wedge F_2 \wedge \dots \wedge F_n$ is true in I, G is also true in I.

Note that validity and satisfiability applies to closed formulas.

Examples: Prove that

- $\underset{x}{\forall} P[x] \wedge \underset{y}{\exists} \neg P[y]$ is inconsistent.

(Un)Satisfiability & (In)Validity

A formula F is satisfiable (consistent) iff there exists an interpretation I such that F is evaluated to $\mathbb{T}$ in I.

A formula F is unsatisfiable (inconsistent) iff for all interpretations I, F is evaluated to $\mathbb{F}$ in I.

A formula F is valid iff for all interpretations I, F is evaluated to $\mathbb{T}$ in I.

A formula F is invalid iff there exists an interpretation I, such that F is evaluated to $\mathbb{F}$ in I.

A formula G is a logical consequence of formulas F_1, F_2, ..., F_n iff for every interpretation I, if $F_1 \wedge F_2 \wedge \ldots \wedge F_n$ is true in I, G is also true in I.

Note that validity and satisfiability applies to closed formulas.

Examples: Prove that

- $\underset{x}{\forall} P[x] \wedge \underset{y}{\exists} \neg P[y]$ is inconsistent.

(Un)Satisfiability & (In)Validity

A formula F is satisfiable (consistent) iff there exists an interpretation I such that F is evaluated to $\mathbb{T}$ in I.

A formula F is unsatisfiable (inconsistent) iff for all interpretations I, F is evaluated to $\mathbb{F}$ in I.

A formula F is valid iff for all interpretations I, F is evaluated to $\mathbb{T}$ in I.

A formula F is invalid iff there exists an interpretation I, such that F is evaluated to $\mathbb{F}$ in I.

A formula G is a logical consequence of formulas F_1, F_2, ..., F_n iff for every interpretation I, if $F_1 \wedge F_2 \wedge \ldots \wedge F_n$ is true in I, G is also true in I.

Note that validity and satisfiability applies to closed formulas.

Examples: Prove that

- $\underset{x}{\forall} P[x] \wedge \underset{y}{\exists} \neg P[y]$ is inconsistent.

(Un)Satisfiability & (In)Validity

A formula F is satisfiable (consistent) iff there exists an interpretation I such that F is evaluated to $\mathbb{T}$ in I.

A formula F is unsatisfiable (inconsistent) iff for all interpretations I, F is evaluated to $\mathbb{F}$ in I.

A formula F is valid iff for all interpretations I, F is evaluated to $\mathbb{T}$ in I.

A formula F is invalid iff there exists an interpretation I, such that F is evaluated to $\mathbb{F}$ in I.

A formula G is a logical consequence of formulas F_1, F_2, ..., F_n iff for every interpretation I, if $F_1 \land F_2 \land \ldots \land F_n$ is true in I, G is also true in I.

Note that validity and satisfiability applies to closed formulas.

Examples: Prove that

- $\underset{x}{\forall} P[x] \land \underset{y}{\exists} \neg P[y]$ is inconsistent.

(Un)Satisfiability & (In)Validity

A formula F is satisfiable (consistent) iff there exists an interpretation I such that F is evaluated to $\mathbb{T}$ in I.

A formula F is unsatisfiable (inconsistent) iff for all interpretations I, F is evaluated to $\mathbb{F}$ in I.

A formula F is valid iff for all interpretations I, F is evaluated to $\mathbb{T}$ in I.

A formula F is invalid iff there exists an interpretation I, such that F is evaluated to $\mathbb{F}$ in I.

A formula G is a logical consequence of formulas F_1, F_2, ..., F_n iff for every interpretation I, if $F_1 \wedge F_2 \wedge \ldots \wedge F_n$ is true in I, G is also true in I.

Note that validity and satisfiability applies to closed formulas.

Examples: Prove that

- $\underset{x}{\forall} P[x] \wedge \underset{y}{\exists} \neg P[y]$ is inconsistent.

(Un)Satisfiability & (In)Validity

A formula F is satisfiable (consistent) iff there exists an interpretation I such that F is evaluated to $\mathbb{T}$ in I.

A formula F is unsatisfiable (inconsistent) iff for all interpretations I, F is evaluated to $\mathbb{F}$ in I.

A formula F is valid iff for all interpretations I, F is evaluated to $\mathbb{T}$ in I.

A formula F is invalid iff there exists an interpretation I, such that F is evaluated to $\mathbb{F}$ in I.

A formula G is a logical consequence of formulas F_1, F_2, ..., F_n iff for every interpretation I, if $F_1 \ \wedge \ F_2 \ \wedge \ \ldots \ \wedge \ F_n$ is true in I, G is also true in I.

Note that validity and satisfiability applies to closed formulas.

Examples: Prove that

- $\underset{x}{\forall} P[x] \ \wedge \ \underset{y}{\exists} \neg P[y]$ is inconsistent.

(Un)Satisfiability & (In)Validity

A formula F is satisfiable (consistent) iff there exists an interpretation I such that F is evaluated to $\mathbb{T}$ in I.

A formula F is unsatisfiable (inconsistent) iff for all interpretations I, F is evaluated to $\mathbb{F}$ in I.

A formula F is valid iff for all interpretations I, F is evaluated to $\mathbb{T}$ in I.

A formula F is invalid iff there exists an interpretation I, such that F is evaluated to $\mathbb{F}$ in I.

A formula G is a logical consequence of formulas F_1, F_2, ..., F_n iff for every interpretation I, if $F_1 \land F_2 \land \ldots \land F_n$ is true in I, G is also true in I.

Note that validity and satisfiability applies to closed formulas.

Examples: Prove that

- $\underset{x}{\forall} P[x] \land \underset{y}{\exists} \neg P[y]$ is inconsistent.

Outline

Syntax

Semantics

(Un)Satisfiability & (In)Validity

Equivalences of Formulas

Normal Forms

Formula Clausification

Substitution

Equivalences of Formulas

Two formulas F and G are equivalent iff the truth values of F and G are the same under any interpretation.

$F \iff G \equiv (F \Rightarrow G) \wedge (G \Rightarrow F)$	
$F \Rightarrow G \equiv \neg F \vee G$	
$F \vee G \equiv G \vee F$	$F \wedge G \equiv G \wedge F$
$(F \vee G) \vee H \equiv F \vee (G \vee H)$	$(F \wedge G) \wedge H \equiv F \wedge (G \wedge H)$
$F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$	$F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$
$F \vee \mathbb{T} \equiv \mathbb{T}$	$F \wedge \mathbb{T} \equiv F$
$F \vee \mathbb{F} \equiv F$	$F \wedge \mathbb{F} \equiv \mathbb{F}$
$F \vee \neg F \equiv \mathbb{T}$	$F \wedge \neg F \equiv \mathbb{F}$
$\neg(\neg F) \equiv F$	
$\neg(F \vee G) \equiv \neg F \wedge \neg G$	$\neg(F \wedge G) \equiv \neg F \vee \neg G$
$(Qx)F[x] \vee G \equiv (Qx)(F[x] \vee G)$	$(Qx)F[x] \wedge G \equiv (Qx)(F[x] \wedge G)$
$\neg \underset{x}{\forall} F[x] \equiv \underset{x}{\exists} \neg F[x]$	$\neg(\underset{x}{\exists} x)F[x] \equiv \underset{x}{\forall} \neg F[x]$
$\underset{x}{\forall} F[x] \vee \underset{x}{\forall} G[x] \not\equiv \underset{x}{\forall}(F[x] \vee G[x])$	$\underset{x}{\forall} F[x] \wedge \underset{x}{\forall} G[x] \equiv \underset{x}{\forall}(F[x] \wedge G[x])$
$\underset{x}{\exists} F[x] \vee \underset{x}{\exists} G[x] \equiv \underset{x}{\exists}(F[x] \vee G[x])$	$\underset{x}{\exists} F[x] \wedge \underset{x}{\exists} G[x] \not\equiv \underset{x}{\exists}(F[x] \wedge G[x])$

Which implications do not hold in the $\not\equiv$ above?

$\Longrightarrow$	$\not\Longrightarrow$
$\not\Longleftarrow$	$\Longleftarrow$

Equivalences of Formulas

$$
\begin{aligned}
F \iff G &\equiv (F \Rightarrow G) \land (G \Rightarrow F) \\
F \Rightarrow G &\equiv \neg F \lor G \\
F \lor G &\equiv G \lor F \\
(F \lor G) \lor H &\equiv F \lor (G \lor H) \\
F \lor (G \land H) &\equiv (F \lor G) \land (F \lor H) \\
F \lor \mathbb{T} &\equiv \mathbb{T} \\
F \lor \mathbb{F} &\equiv F \\
F \lor \neg F &\equiv \mathbb{T} \\
\neg(\neg F) &\equiv F \\
\neg(F \lor G) &\equiv \neg F \land \neg G \\
(Qx)F[x] \lor G &\equiv (Qx)(F[x] \lor G) \\
\neg \forall_x F[x] &\equiv \exists_x \neg F[x] \\
\forall_x F[x] \lor \forall_x G[x] &\not\equiv \forall_x (F[x] \lor G[x]) \\
\exists_x F[x] \lor \exists_x G[x] &\equiv \exists_x (F[x] \lor G[x])
\end{aligned}
$$

$$
\begin{aligned}
F \land G &\equiv G \land F \\
(F \land G) \land H &\equiv F \land (G \land H) \\
F \land (G \lor H) &\equiv (F \land G) \lor (F \land H) \\
F \land \mathbb{T} &\equiv F \\
F \land \mathbb{F} &\equiv \mathbb{F} \\
F \land \neg F &\equiv \mathbb{F} \\
\neg(F \land G) &\equiv \neg F \lor \neg G \\
(Qx)F[x] \land G &\equiv (Qx)(F[x] \land G) \\
\neg(\exists_x x)F[x] &\equiv \forall_x \neg F[x] \\
\forall_x F[x] \land \forall_x G[x] &\equiv \forall_x (F[x] \land G[x]) \\
\exists_x F[x] \land \exists_x G[x] &\not\equiv \exists_x (F[x] \land G[x])
\end{aligned}
$$

Which implications do not hold in the $\not\equiv$ above?

Equivalences of Formulas

$$
\begin{aligned}
F &\iff G \equiv (F \Rightarrow G) \land (G \Rightarrow F)\\
F &\Rightarrow G \equiv \neg F \lor G\\
F \lor G &\equiv G \lor F\\
(F \lor G) \lor H &\equiv F \lor (G \lor H)\\
F \lor (G \land H) &\equiv (F \lor G) \land (F \lor H)\\
F \lor \mathbb{T} &\equiv \mathbb{T}\\
F \lor \mathbb{F} &\equiv F\\
F \lor \neg F &\equiv \mathbb{T}\\
\neg(\neg F) &\equiv F\\
\neg(F \lor G) &\equiv \neg F \land \neg G\\
(Qx)F[x] \lor G &\equiv (Qx)(F[x] \lor G)\\
\neg \underset{x}{\forall} F[x] &\equiv \underset{x}{\exists} \neg F[x]\\
\underset{x}{\forall} F[x] \lor \underset{x}{\forall} G[x] &\not\equiv \underset{x}{\forall}(F[x] \lor G[x])\\
\underset{x}{\exists} F[x] \lor \underset{x}{\exists} G[x] &\equiv \underset{x}{\exists}(F[x] \lor G[x])
\end{aligned}
$$

$$
\begin{aligned}
F \land G &\equiv G \land F\\
(F \land G) \land H &\equiv F \land (G \land H)\\
F \land (G \lor H) &\equiv (F \land G) \lor (F \land H)\\
F \land \mathbb{T} &\equiv F\\
F \land \mathbb{F} &\equiv \mathbb{F}\\
F \land \neg F &\equiv \mathbb{F}\\
\neg(F \land G) &\equiv \neg F \lor \neg G\\
(Qx)F[x] \land G &\equiv (Qx)(F[x] \land G)\\
\neg(\underset{x}{\exists} x)F[x] &\equiv \underset{x}{\forall} \neg F[x]\\
\underset{x}{\forall} F[x] \land \underset{x}{\forall} G[x] &\equiv \underset{x}{\forall}(F[x] \land G[x])\\
\underset{x}{\exists} F[x] \land \underset{x}{\exists} G[x] &\not\equiv \underset{x}{\exists}(F[x] \land G[x])
\end{aligned}
$$

Which implications do not hold in the $\not\equiv$ above?

$\Longrightarrow$ $\quad\quad$ $\not\Longrightarrow$

$\not\Longleftarrow$ $\quad\quad$ $\Longleftarrow$

Equivalences of Formulas

$F \iff G \;\equiv\; (F \Rightarrow G) \land (G \Rightarrow F)$	
$F \Rightarrow G \;\equiv\; \neg F \lor G$	
$F \lor G \;\equiv\; G \lor F$	$F \land G \;\equiv\; G \land F$
$(F \lor G) \lor H \;\equiv\; F \lor (G \lor H)$	$(F \land G) \land H \;\equiv\; F \land (G \land H)$
$F \lor (G \land H) \;\equiv\; (F \lor G) \land (F \lor H)$	$F \land (G \lor H) \;\equiv\; (F \land G) \lor (F \land H)$
$F \lor \mathbb{T} \;\equiv\; \mathbb{T}$	$F \land \mathbb{T} \;\equiv\; F$
$F \lor \mathbb{F} \;\equiv\; F$	$F \land \mathbb{F} \;\equiv\; \mathbb{F}$
$F \lor \neg F \;\equiv\; \mathbb{T}$	$F \land \neg F \;\equiv\; \mathbb{F}$
$\neg(\neg F) \;\equiv\; F$	
$\neg(F \lor G) \;\equiv\; \neg F \land \neg G$	$\neg(F \land G) \;\equiv\; \neg F \lor \neg G$
$(Qx)F[x] \lor G \;\equiv\; (Qx)(F[x] \lor G)$	$(Qx)F[x] \land G \;\equiv\; (Qx)(F[x] \land G)$
$\neg \underset{x}{\forall} F[x] \;\equiv\; \underset{x}{\exists} \neg F[x]$	$\neg(\underset{x}{\exists})F[x] \;\equiv\; \underset{x}{\forall} \neg F[x]$
$\underset{x}{\forall} F[x] \lor \underset{x}{\forall} G[x] \;\not\equiv\; \underset{x}{\forall}(F[x] \lor G[x])$	$\underset{x}{\forall} F[x] \land \underset{x}{\forall} G[x] \;\equiv\; \underset{x}{\forall}(F[x] \land G[x])$
$\underset{x}{\exists} F[x] \lor \underset{x}{\exists} G[x] \;\equiv\; \underset{x}{\exists}(F[x] \lor G[x])$	$\underset{x}{\exists} F[x] \land \underset{x}{\exists} G[x] \;\not\equiv\; \underset{x}{\exists}(F[x] \land G[x])$

Which implications do not hold in the $\not\equiv$ above?

$\Longrightarrow$	$\not\Longrightarrow$
$\not\Longleftarrow$	$\Longleftarrow$

Equivalences of Formulas (cont'd)

Note that

$$\forall_x F[x] \vee \forall_x G[x] \quad \equiv \quad \forall_x F[x] \vee \forall_y G[y] \quad \equiv \quad \forall_{x,y} F[x] \vee G[y]$$

$$\exists_x F[x] \ \wedge \ \exists_x G[x] \quad \equiv \quad \exists_x F[x] \wedge \exists_y G[y] \quad \equiv \quad \exists_{x,y} F[x] \wedge G[y]$$

Outline

Normal Forms

Normal forms:

1. CNF
2. DNF
3. negation normal form (NNF)
4. prenex normal form (PNF)
5. Skolem standard form

Negation normal form (NNF) requires that $\neg$, $\wedge$, and $\vee$ to be the only logical connectives and that negations appear only in literals.

A formula F in FOL is said to be in prenex normal form (PNF) iff the formula is in the form $(Q_1 x_1) ... (Q_n x_n)\ M$, where $Q_i \in \{\forall, \exists\}$ and M is quantifier-free.

A FOL formula is in Skolem standard form if it is of the form $\underset{x_1,\ldots,x_n}{\forall} M$, where M is a quantifier-free formula in CNF.

Normal Forms

Normal forms:

1. CNF
2. DNF
3. negation normal form (NNF)
4. prenex normal form (PNF)
5. Skolem standard form

Negation normal form (NNF) requires that $\neg$, $\wedge$, and $\vee$ to be the only logical connectives and that negations appear only in literals.

A formula F in FOL is said to be in prenex normal form (PNF) iff the formula is in the form $(Q_1 x_1)...(Q_n x_n)\ M$, where $Q_i \in \{\forall, \exists\}$ and M is quantifier-free.

A FOL formula is in Skolem standard form if it is of the form $\underset{x_1,\ldots,x_n}{\forall} M$, where M is a quantifier-free formula in CNF.

Normal Forms

Normal forms:

1. CNF
2. DNF
3. negation normal form (NNF)
4. prenex normal form (PNF)
5. Skolem standard form

Negation normal form (NNF) requires that $\neg$, $\wedge$, and $\vee$ to be the only logical connectives and that negations appear only in literals.

A formula F in FOL is said to be in prenex normal form (PNF) iff the formula is in the form $(Q_1 x_1)...(Q_n x_n)\ M$, where $Q_i \in \{\forall, \exists\}$ and M is quantifier-free.

A FOL formula is in Skolem standard form if it is of the form $\underset{x_1,\ldots,x_n}{\forall} M$, where M is a quantifier-free formula in CNF.

Normal Forms

Normal forms:

1. CNF
2. DNF
3. negation normal form (NNF)
4. prenex normal form (PNF)
5. Skolem standard form

Negation normal form (NNF) requires that $\neg$, $\wedge$, and $\vee$ to be the only logical connectives and that negations appear only in literals.

A formula F in FOL is said to be in prenex normal form (PNF) iff the formula is in the form $(Q_1 x_1) \ldots (Q_n x_n)\ M$, where $Q_i \in \{\forall, \exists\}$ and M is quantifier-free.

A FOL formula is in Skolem standard form if it is of the form $\underset{x_1, \ldots, x_n}{\forall} M$, where M is a quantifier-free formula in CNF.

Normal Forms

Normal forms:

1. CNF
2. DNF
3. negation normal form (NNF)
4. prenex normal form (PNF)
5. Skolem standard form

Negation normal form (NNF) requires that $\neg$, $\wedge$, and $\vee$ to be the only logical connectives and that negations appear only in literals.

A formula F in FOL is said to be in prenex normal form (PNF) iff the formula is in the form $(Q_1 x_1)\ldots(Q_n x_n)\ M$, where $Q_i \in \{\forall, \exists\}$ and M is quantifier-free.

A FOL formula is in Skolem standard form if it is of the form $\underset{x_1,\ldots,x_n}{\forall} M$, where M is a quantifier-free formula in CNF.

Normal Forms

Normal forms:

1. CNF
2. DNF
3. negation normal form (NNF)
4. prenex normal form (PNF)
5. Skolem standard form

Negation normal form (NNF) requires that $\neg$, $\wedge$, and $\vee$ to be the only logical connectives and that negations appear only in literals.

A formula F in FOL is said to be in prenex normal form (PNF) iff the formula is in the form $(Q_1 x_1) \ldots (Q_n x_n)\ M$, where $Q_i \in \{\forall, \exists\}$ and M is quantifier-free.

A FOL formula is in Skolem standard form if it is of the form $\underset{x_1, \ldots, x_n}{\forall} M$, where M is a quantifier-free formula in CNF.

Normal Forms

Normal forms:

1. CNF
2. DNF
3. negation normal form (NNF)
4. prenex normal form (PNF)
5. Skolem standard form

Negation normal form (NNF) requires that $\neg$, $\wedge$, and $\vee$ to be the only logical connectives and that negations appear only in literals.

A formula F in FOL is said to be in prenex normal form (PNF) iff the formula is in the form $(Q_1 x_1) \ldots (Q_n x_n)\ M$, where $Q_i \in \{\forall, \exists\}$ and M is quantifier-free.

A FOL formula is in Skolem standard form if it is of the form $\underset{x_1,\ldots,x_n}{\forall} M$, where M is a quantifier-free formula in CNF.

Normal Forms

Normal forms:

1. CNF
2. DNF
3. negation normal form (NNF)
4. prenex normal form (PNF)
5. Skolem standard form

Negation normal form (NNF) requires that $\neg$, $\wedge$, and $\vee$ to be the only logical connectives and that negations appear only in literals.

A formula F in FOL is said to be in prenex normal form (PNF) iff the formula is in the form $(Q_1 x_1)...(Q_n x_n)\ M$, where $Q_i \in \{\forall, \exists\}$ and M is quantifier-free.

A FOL formula is in Skolem standard form if it is of the form $\underset{x_1,\ldots,x_n}{\forall} M$, where M is a quantifier-free formula in CNF.

Normal Forms

Normal forms:

1. CNF
2. DNF
3. negation normal form (NNF)
4. prenex normal form (PNF)
5. Skolem standard form

Negation normal form (NNF) requires that $\neg$, $\wedge$, and $\vee$ to be the only logical connectives and that negations appear only in literals.

A formula F in FOL is said to be in prenex normal form (PNF) iff the formula is in the form $(Q_1 x_1) \ldots (Q_n x_n)\ M$, where $Q_i \in \{\forall, \exists\}$ and M is quantifier-free.

A FOL formula is in Skolem standard form if it is of the form $\underset{x_1,\ldots,x_n}{\forall} M$, where M is a quantifier-free formula in CNF.

Normal Forms (cont'd)

Examples:

1. Prove the following by bringing the formulas into conjunctive normal form
$$\left(\underset{x}{\forall} P[x]\right) \Rightarrow Q \quad \equiv \quad \underset{x}{\exists} (P[x] \Rightarrow Q)\,.$$

2. Bring the following formulas into Skolem standard form

   - $$\underset{x}{\forall}\ \underset{y,z}{\exists} ((\neg P[x,y] \ \wedge\ Q[x,z]) \ \vee\ R[x,y,z])$$

   - $$\underset{x,y}{\forall} \left(\underset{z}{\exists} P[x,z] \wedge P[y,z]\right) \ \Rightarrow\ \underset{u}{\exists} Q[x,y,u]$$

Normal Forms (cont'd)

Examples:

1. Prove the following by bringing the formulas into conjunctive normal form
$$\left(\underset{x}{\forall} P[x]\right) \Rightarrow Q \;\equiv\; \underset{x}{\exists} (P[x] \Rightarrow Q)\,.$$
2. Bring the following formulas into Skolem standard form
   - $$\underset{x}{\forall}\ \underset{y,z}{\exists} ((\neg P[x,y] \;\wedge\; Q[x,z]) \;\vee\; R[x,y,z])$$
   - $$\underset{x,y}{\forall} \left(\underset{z}{\exists} P[x,z] \wedge P[y,z]\right) \;\Rightarrow\; \underset{u}{\exists} Q[x,y,u]$$

Normal Forms (cont'd)

Examples:

1. Prove the following by bringing the formulas into conjunctive normal form
$$\left(\underset{x}{\forall} P[x]\right) \Rightarrow Q \;\equiv\; \underset{x}{\exists}\,(P[x] \Rightarrow Q)\,.$$
2. Bring the following formulas into Skolem standard form
   - $$\underset{x}{\forall}\ \underset{y,z}{\exists}\,((\neg P[x,y] \;\wedge\; Q[x,z]) \;\vee\; R[x,y,z])$$
   - $$\underset{x,y}{\forall}\left(\underset{z}{\exists} P[x,z] \wedge P[y,z]\right) \;\Rightarrow\; \underset{u}{\exists} Q[x,y,u]$$

Normal Forms (cont'd)

Examples:

1. Prove the following by bringing the formulas into conjunctive normal form
$$\left(\underset{x}{\forall} P[x]\right) \Rightarrow Q \quad \equiv \quad \underset{x}{\exists} (P[x] \Rightarrow Q)\,.$$

2. Bring the following formulas into Skolem standard form
   - $$\underset{x}{\forall}\ \underset{y,z}{\exists} ((\neg P[x,y] \ \wedge\ Q[x,z]) \ \vee\ R[x,y,z])$$
   - $$\underset{x,y}{\forall} \left(\underset{z}{\exists} P[x,z] \wedge P[y,z]\right) \ \Rightarrow\ \underset{u}{\exists} Q[x,y,u]$$

Normal Forms (cont'd)

Examples:

1. Prove the following by bringing the formulas into conjunctive normal form
$$\left(\underset{x}{\forall} P[x]\right) \Rightarrow Q \quad \equiv \quad \underset{x}{\exists}\,(P[x] \Rightarrow Q)\,.$$
2. Bring the following formulas into Skolem standard form
   - $$\underset{x}{\forall}\ \underset{y,z}{\exists}((\neg P[x,y]\ \wedge\ Q[x,z])\ \vee\ R[x,y,z])$$
   - $$\underset{x,y}{\forall}\left(\underset{z}{\exists} P[x,z] \wedge P[y,z]\right) \ \Rightarrow\ \underset{u}{\exists} Q[x,y,u]$$

Outline

Formula Clausification

A clause is a disjunction of literals.

Examples: $\neg P[x] \vee Q[y, f[x]]$, $P[x]$

A set of clauses S is regarded as a conjunction of all clauses in S, where every variable in S is considered governed by a universal quantifier.

Example: Let

$$\underset{x}{\forall} \underset{y,z}{\exists} ((\neg P[x, y] \ \wedge \ Q[x, z]) \ \vee \ R[x, y, z])$$

The standard form of the formula above, that is

$$\underset{x}{\forall} ((\neg P[x, f[x]] \ \vee \ R[x, f[x], g[x]]) \ \wedge \ (Q(x, g[x]) \ \vee \ R[x, f[x], g[x]]))$$

can be represented by the following set of clauses

$$\{\neg P[x, f[x]] \ \vee \ R[x, f[x], g[x]], Q(x, g[x]) \ \vee \ R[x, f[x], g[x]]\}$$

Note that, if S is a set of clauses that represents a standard form of a formula F, then F is inconsistent iff S is inconsistent.

Formula Clausification

A clause is a disjunction of literals.

Examples: $\neg P[x] \vee Q[y, f[x]]$, $P[x]$

A set of clauses S is regarded as a conjunction of all clauses in S, where every variable in S is considered governed by a universal quantifier.

Example: Let

$$\underset{x}{\forall}\ \underset{y,z}{\exists}((\neg P[x,y]\ \wedge\ Q[x,z])\ \vee\ R[x,y,z])$$

The standard form of the formula above, that is

$$\underset{x}{\forall}((\neg P[x,f[x]]\ \vee\ R[x,f[x],g[x]])\ \wedge\ (Q(x,g[x])\ \vee\ R[x,f[x],g[x]]))$$

can be represented by the following set of clauses

$$\{\neg P[x,f[x]]\ \vee\ R[x,f[x],g[x]], Q(x,g[x])\ \vee\ R[x,f[x],g[x]]\}$$

Note that, if S is a set of clauses that represents a standard form of a formula F, then F is inconsistent iff S is inconsistent.

Formula Clausification

A clause is a disjunction of literals.

Examples: $\neg P[x] \vee Q[y, f[x]]$, $P[x]$

A set of clauses S is regarded as a conjunction of all clauses in S, where every variable in S is considered governed by a universal quantifier.

Example: Let

$$\underset{x}{\forall} \underset{y,z}{\exists} ((\neg P[x, y] \ \wedge \ Q[x, z]) \ \vee \ R[x, y, z])$$

The standard form of the formula above, that is

$$\underset{x}{\forall} ((\neg P[x, f[x]] \ \vee \ R[x, f[x], g[x]]) \ \wedge \ (Q(x, g[x]) \ \vee \ R[x, f[x], g[x]]))$$

can be represented by the following set of clauses

$$\{\neg P[x, f[x]] \ \vee \ R[x, f[x], g[x]], Q(x, g[x]) \ \vee \ R[x, f[x], g[x]]\}$$

Note that, if S is a set of clauses that represents a standard form of a formula F, then F is inconsistent iff S is inconsistent.

Formula Clausification

A clause is a disjunction of literals.

Examples: $\neg P[x] \vee Q[y, f[x]]$, $P[x]$

A set of clauses S is regarded as a conjunction of all clauses in S, where every variable in S is considered governed by a universal quantifier.

Example: Let

$$\underset{x}{\forall} \underset{y,z}{\exists} ((\neg P[x, y] \ \wedge \ Q[x, z]) \ \vee \ R[x, y, z])$$

The standard form of the formula above, that is

$$\underset{x}{\forall} ((\neg P[x, f[x]] \ \vee \ R[x, f[x], g[x]]) \ \wedge \ (Q(x, g[x]) \ \vee \ R[x, f[x], g[x]]))$$

can be represented by the following set of clauses

$$\{\neg P[x, f[x]] \ \vee \ R[x, f[x], g[x]], Q(x, g[x]) \ \vee \ R[x, f[x], g[x]]\}$$

Note that, if S is a set of clauses that represents a standard form of a formula F, then F is inconsistent iff S is inconsistent.

Formula Clausification

A clause is a disjunction of literals.

Examples: $\neg P[x] \vee Q[y, f[x]]$, $P[x]$

A set of clauses S is regarded as a conjunction of all clauses in S, where every variable in S is considered governed by a universal quantifier.

Example: Let

$$\underset{x}{\forall} \underset{y,z}{\exists} ((\neg P[x, y] \ \wedge \ Q[x, z]) \ \vee \ R[x, y, z])$$

The standard form of the formula above, that is

$$\underset{x}{\forall} ((\neg P[x, f[x]] \ \vee \ R[x, f[x], g[x]]) \ \wedge \ (Q(x, g[x]) \ \vee \ R[x, f[x], g[x]]))$$

can be represented by the following set of clauses

$$\{\neg P[x, f[x]] \ \vee \ R[x, f[x], g[x]], Q(x, g[x]) \ \vee \ R[x, f[x], g[x]]\}$$

Note that, if S is a set of clauses that represents a standard form of a formula F, then F is inconsistent iff S is inconsistent.

Formulas Clausification (cont'd)

Example :

Transform the formulas F_1, F_2,F_3,F_4, and $\neg G$ into a set of clauses, where

$$F_1: \quad \underset{x,y}{\forall}\ \underset{z}{\exists} P[x,y,z]$$

$$F_2: \quad \begin{array}{l} \underset{x,y,z,u,v,w}{\forall} ((P[x,y,u] \wedge P[y,z,v] \wedge P[u,z,w]) \Rightarrow P[x,v,w]) \\ \wedge \\ \underset{x,y,z,u,v,w}{\forall} (P[x,y,u] \wedge (P[y,z,v] \wedge P[x,v,w]) \Rightarrow P[u,z,w]) \end{array}$$

$$F_3: \quad \underset{x}{\forall} P[x,e,x] \wedge \underset{x}{\forall} P[e,x,x]$$

$$F_4: \quad \underset{x}{\forall} P[x,i[x],e] \wedge \underset{x}{\forall} P[i[x],x,e]$$

$$G: \quad \left(\underset{x}{\forall} P[x,x,e]\right) \Rightarrow \underset{u,v,w}{\forall} (P[u,v,w] \Rightarrow P[v,u,w])$$

Outline

Substitution

Example: Let

$$\begin{array}{ll} C_1 : & P[x] \vee Q[x] \\ C_2 : & \neg P[f[x]] \vee R[x] \end{array}$$

Substitution

Example: Let

$$\begin{array}{ll} C_1 : & P[x] \lor Q[x] \\ C_2 : & \neg P[f[x]] \lor R[x] \end{array}$$

Substitution

Example: Let

$$\begin{array}{ll} C_1 : & P[x] \lor Q[x] \\ C_2 : & \neg P[f[x]] \lor R[x] \end{array}$$

Let $x \to f[a]$ in C_1, $x \to a$ in C_2.

Substitution

Example: Let

$$\begin{array}{ll} C_1: & P[x] \vee Q[x] \\ C_2: & \neg P[f[x]] \vee R[x] \end{array}$$

Let $x \to f[a]$ in C_1, $x \to a$ in C_2.

We have

$$\begin{array}{ll} C_1': & P[f[a]] \vee Q[f[a]] \\ C_2': & \neg P[f[a]] \vee R[a] \end{array}$$

Substitution

Example: Let

$$\begin{array}{ll} C_1: & P[x] \vee Q[x] \\ C_2: & \neg P[f[x]] \vee R[x] \end{array}$$

Let $x \rightarrow f[a]$ in C_1, $x \rightarrow a$ in C_2.

We have

$$\begin{array}{ll} C_1': & P[f[a]] \vee Q[f[a]] \\ C_2': & \neg P[f[a]] \vee R[a] \end{array}$$

C_1' and C_2' are ground instances.

Substitution

Example: Let

$$C_1 : \qquad P[x] \lor Q[x]$$
$$C_2 : \qquad \neg P[f[x]] \lor R[x]$$

Let $x \to f[a]$ in C_1, $x \to a$ in C_2.

We have

$$C_1' : \qquad P[f[a]] \lor Q[f[a]]$$
$$C_2' : \qquad \neg P[f[a]] \lor R[a]$$

C_1' and C_2' are ground instances.

A resolvent of C_1' and C_2' is

$$C_3' : \qquad Q[f[a]] \lor R[a]$$

Substitution

Example: Let

$$\begin{array}{ll} C_1: & P[x] \vee Q[x] \\ C_2: & \neg P[f[x]] \vee R[x] \end{array}$$

Let $x \to f[x]$ in C_1. We have

$$C_1^*: \quad P[f[x]] \vee Q[f[x]]$$

C_1^* is an instance of C_1.

A resolvent of

$$\begin{array}{ll} C_2: & \neg P[f[x]] \vee R[x] \\ C_1^*: & P[f[x]] \vee Q[f[x]] \end{array}$$

is

$$C_3: \quad Q[f[x]] \vee R[x]$$

C_3' is an instance of C_3. C_3 is the most general clause.

Substitution

Example: Let

$$\begin{array}{ll} C_1: & P[x] \vee Q[x] \\ C_2: & \neg P[f[x]] \vee R[x] \end{array}$$

Let $x \to f[x]$ in C_1. We have

$$C_1^*: \quad P[f[x]] \vee Q[f[x]]$$

C_1^* is an instance of C_1.

A resolvent of

$$\begin{array}{ll} C_2: & \neg P[f[x]] \vee R[x] \\ C_1^*: & P[f[x]] \vee Q[f[x]] \end{array}$$

is

$$C_3: \quad Q[f[x]] \vee R[x]$$

C_3' is an instance of C_3. C_3 is the most general clause.

Substitution

Example: Let

$$C_1 : \qquad P[x] \vee Q[x]$$
$$C_2 : \qquad \neg P[f[x]] \vee R[x]$$

Let $x \to f[x]$ in C_1. We have

$$C_1^* : \qquad P[f[x]] \vee Q[f[x]]$$

C_1^* is an instance of C_1.

A resolvent of

$$C_2 : \qquad \neg P[f[x]] \vee R[x]$$
$$C_1^* : \qquad P[f[x]] \vee Q[f[x]]$$

is

$$C_3 : \qquad Q[f[x]] \vee R[x]$$

C_3' is an instance of C_3. C_3 is the most general clause.

Substitution

Example: Let

$$\begin{array}{ll} C_1 : & P[x] \vee Q[x] \\ C_2 : & \neg P[f[x]] \vee R[x] \end{array}$$

Let $x \to f[x]$ in C_1. We have

$$C_1^* : \quad P[f[x]] \vee Q[f[x]]$$

C_1^* is an instance of C_1.

A resolvent of

$$\begin{array}{ll} C_2 : & \neg P[f[x]] \vee R[x] \\ C_1^* : & P[f[x]] \vee Q[f[x]] \end{array}$$

is

$$C_3 : \quad Q[f[x]] \vee R[x]$$

C_3' is an instance of C_3. C_3 is the most general clause.

Substitution

Example: Let

$$\begin{array}{ll} C_1: & P[x] \vee Q[x] \\ C_2: & \neg P[f[x]] \vee R[x] \end{array}$$

Let $x \to f[x]$ in C_1. We have

$$C_1^*: \quad P[f[x]] \vee Q[f[x]]$$

C_1^* is an instance of C_1.

A resolvent of

$$\begin{array}{ll} C_2: & \neg P[f[x]] \vee R[x] \\ C_1^*: & P[f[x]] \vee Q[f[x]] \end{array}$$

is

$$C_3: \quad Q[f[x]] \vee R[x]$$

C_3' is an instance of C_3. C_3 is the most general clause.

Substitution (cont'd)

A substitution σ is a finite set of the form $\{v_1 \to t_1, ..., v_n \to t_n\}$ where every t_i is a term different from v_i and no two elements in the set have the same variable v_i.

Let σ be defined as above and E be an expression. Then $E\sigma$ is an expression obtained from E by replacing simultaneously each occurrence of v_i in E by the term t_i

Example: Let $\sigma = \{x \to z, z \to h[a, y]\}$ and $E = f[z, a, g[x], y]$. Then $E\sigma = f[h[a, y], a, g[z], y]$.

Substitution (cont'd)

A substitution σ is a finite set of the form $\{v_1 \to t_1, ..., v_n \to t_n\}$ where every t_i is a term different from v_i and no two elements in the set have the same variable v_i.

Let σ be defined as above and E be an expression. Then $E\sigma$ is an expression obtained from E by replacing simultaneously each occurrence of v_i in E by the term t_i

Example: Let $\sigma = \{x \to z, z \to h[a, y]\}$ and $E = f[z, a, g[x], y]$. Then $E\sigma = f[h[a, y], a, g[z], y]$.

Substitution (cont'd)

A substitution σ is a finite set of the form $\{v_1 \to t_1, ..., v_n \to t_n\}$ where every t_i is a term different from v_i and no two elements in the set have the same variable v_i.

Let σ be defined as above and E be an expression. Then $E\sigma$ is an expression obtained from E by replacing simultaneously each occurrence of v_i in E by the term t_i

Example: Let $\sigma = \{x \to z, z \to h[a, y]\}$ and $E = f[z, a, g[x], y]$. Then $E\sigma = f[h[a, y], a, g[z], y]$.

Substitution (cont'd)

Let

$$\theta = \{x_1 \to t_1, ..., x_n \to t_n\}$$
$$\lambda = \{y_1 \to u_1, ..., y_n \to u_n\}$$

Then the composition of θ and λ ($\theta \circ \lambda$) is obtained from the set

$$\{x_1 \to t_1\lambda, ..., x_n \to t_n\lambda, y_1 \to u_1, ..., y_n \to u_n\}$$

by deleting any element $x_j \to t_j\lambda$ for which $x_j = t_j\lambda$ and any element $y_i \to u_i$ such that y_i is among $\{x_1, ..., x_n\}$.

Substitution (cont'd)

Example 1:

$$\theta = \{x \to f[y], y \to z\}$$
$$\lambda = \{x \to a, y \to b, z \to y\}$$

Then

$$\theta \circ \lambda = \{x \to f[b], y \to y, x \to a, y \to b, z \to y\}$$
$$= \{x \to f[b], z \to y\}$$

Example 2:

$$\theta_1 = \{x \to a, y \to f[z], z \to y\}$$
$$\theta_2 = \{x \to b, y \to z, z \to g[x]\}$$

Then

$$\theta_1 \circ \theta_2 = \{x \to a, y \to f[g[x]], z \to z, x \to b, y \to z, z \to g[x]\}$$
$$= \{x \to a, y \to f[g[x]]\}$$

Substitution (cont'd)

Example 1:

$$\theta = \{x \to f[y], y \to z\}$$
$$\lambda = \{x \to a, y \to b, z \to y\}$$

Then

$$\begin{aligned}\theta \circ \lambda &= \{x \to f[b], y \to y, x \to a, y \to b, z \to y\} \\ &= \{x \to f[b], z \to y\}\end{aligned}$$

Example 2:

$$\theta_1 = \{x \to a, y \to f[z], z \to y\}$$
$$\theta_2 = \{x \to b, y \to z, z \to g[x]\}$$

Then

$$\begin{aligned}\theta_1 \circ \theta_2 &= \{x \to a, y \to f[g[x]], z \to z, x \to b, y \to z, z \to g[x]\} \\ &= \{x \to a, y \to f[g[x]]\}\end{aligned}$$

Substitution (cont'd)

Example 1:

$$\theta = \{x \to f[y], y \to z\}$$
$$\lambda = \{x \to a, y \to b, z \to y\}$$

Then

$$\begin{aligned}\theta \circ \lambda &= \{x \to f[b], y \to y, x \to a, y \to b, z \to y\} \\ &= \{x \to f[b], z \to y\}\end{aligned}$$

Example 2:

$$\theta_1 = \{x \to a, y \to f[z], z \to y\}$$
$$\theta_2 = \{x \to b, y \to z, z \to g[x]\}$$

Then

$$\begin{aligned}\theta_1 \circ \theta_2 &= \{x \to a, y \to f[g[x]], z \to z, x \to b, y \to z, z \to g[x]\} \\ &= \{x \to a, y \to f[g[x]]\}\end{aligned}$$

Substitution (cont'd)

Example 1:

$$\theta = \{x \to f[y], y \to z\}$$
$$\lambda = \{x \to a, y \to b, z \to y\}$$

Then

$$\begin{aligned} \theta \circ \lambda &= \{x \to f[b], y \to y, x \to a, y \to b, z \to y\} \\ &= \{x \to f[b], z \to y\} \end{aligned}$$

Example 2:

$$\theta_1 = \{x \to a, y \to f[z], z \to y\}$$
$$\theta_2 = \{x \to b, y \to z, z \to g[x]\}$$

Then

$$\begin{aligned} \theta_1 \circ \theta_2 &= \{x \to a, y \to f[g[x]], z \to z, x \to b, y \to z, z \to g[x]\} \\ &= \{x \to a, y \to f[g[x]]\} \end{aligned}$$