Lecture 2 — First-Order Logic

Ondřej Lengál

Faculty of Information Technology
Brno University of Technology

IAM'24

First-Order Logic

First-Order Logic (FOL)

- also called (first-order) predicate logic, predicate calculus, ...

First-Order Logic

First-Order Logic (FOL)

- also called (first-order) predicate logic, predicate calculus, . . .
- generalizes propositional logic by
  - interpreting ("looking inside") propositions

First-Order Logic

First-Order Logic (FOL)

- also called (first-order) predicate logic, predicate calculus, ...
- generalizes propositional logic by
  - interpreting (“looking inside”) propositions
  - talks about elements of a universe—denoted by terms formed from variables, constants, and functions
    - e.g., x, 5, $f(x, 2)$, $36 + 2 \cdot 3$, $\mathit{fatherOf}(\mathit{motherOf}(x))$, $\mathit{head}(\texttt{"abc"})$, $\sin(y)$, ...

First-Order Logic (FOL)

- also called (first-order) predicate logic, predicate calculus, ...
- generalizes propositional logic by
  - interpreting ("looking inside") propositions
  - talks about elements of a universe—denoted by terms formed from variables, constants, and functions
    - e.g., x, 5, $f(x, 2)$, $36 + 2 \cdot 3$, $\mathit{fatherOf}(\mathit{motherOf}(x))$, $\mathit{head}(\texttt{"abc"})$, $\sin(y)$, $\ldots$
  - propositions are substituted with predicates over terms
    - e.g., $x = y$, $\mathit{even}(x)$, $p(x, y, z)$, $\mathit{isFatherOf}(x, y)$, $\ldots$

First-Order Logic (FOL)

- also called (first-order) predicate logic, predicate calculus, ...
- generalizes propositional logic by
  - interpreting ("looking inside") propositions
  - talks about elements of a universe—denoted by terms formed from variables, constants, and functions
    - e.g., x, 5, $f(x, 2)$, $36 + 2 \cdot 3$, $\mathit{fatherOf}(\mathit{motherOf}(x))$, $\mathit{head}(\texttt{"abc"})$, $\sin(y)$, ...
  - propositions are substituted with predicates over terms
    - e.g., $x = y$, $\mathit{even}(x)$, $p(x, y, z)$, $\mathit{isFatherOf}(x, y)$, ...
  - introducing quantifiers to express existential or universal properties about elements of the universe (first-order quantification)
    - $\forall x$ — universal quantifier (all elements satisfy property)
    - $\exists x$ — existential quantifier (some element satisfies property)

First-Order Logic

First-Order Logic (FOL)

- also called (first-order) predicate logic, predicate calculus, ...
- generalizes propositional logic by
  - interpreting (“looking inside”) propositions
  - talks about elements of a universe—denoted by terms formed from variables, constants, and functions
    - e.g., x, 5, $f(x, 2)$, $36 + 2 \cdot 3$, $\mathit{fatherOf}(\mathit{motherOf}(x))$, $\mathit{head}(\texttt{"abc"})$, $\sin(y)$, ...
  - propositions are substituted with predicates over terms
    - e.g., $x = y$, $\mathit{even}(x)$, $p(x, y, z)$, $\mathit{isFatherOf}(x, y)$, ...
  - introducing quantifiers to express existential or universal properties about elements of the universe (first-order quantification)
    - $\forall x$ — universal quantifier (all elements satisfy property)
    - $\exists x$ — existential quantifier (some element satisfies property)
- much more expressive than propositional logic!
  - therefore, also more complex (in general undecidable)

First-Order Logic

Example

- All men are mortal. Socrates is a man. So Socrates is mortal.

First-Order Logic

Example

- All men are mortal. Socrates is a man. So Socrates is mortal.

$$\big(\forall x(man(x) \to mortal(x)) \land man(Socrates)\big) \to mortal(Socrates)$$

First-Order Logic

Example

- All men are mortal. Socrates is a man. So Socrates is mortal.

$$\big(\forall x(man(x) \to mortal(x)) \wedge man(Socrates)\big) \to mortal(Socrates)$$

- There are infinitely many prime numbers.

First-Order Logic

Example

- All men are mortal. Socrates is a man. So Socrates is mortal.

$$(\forall x(man(x) \rightarrow mortal(x)) \wedge man(Socrates)) \rightarrow mortal(Socrates)$$

- There are infinitely many prime numbers.

$$\forall x \exists y \Big(y > x \wedge \forall z((z \neq 1 \wedge z \neq y) \rightarrow \forall w(wz \neq y))\Big)$$

First-Order Logic

Example

- All men are mortal. Socrates is a man. So Socrates is mortal.

$$\big(\forall x(man(x) \rightarrow mortal(x)) \wedge man(Socrates)\big) \rightarrow mortal(Socrates)$$

- There are infinitely many prime numbers.

$$\forall x \exists y \Big(y > x \wedge \forall z\big((z \neq 1 \wedge z \neq y) \rightarrow \forall w(wz \neq y)\big)\Big)$$

- The relation R is transitive.

First-Order Logic

Example

- All men are mortal. Socrates is a man. So Socrates is mortal.

$$\big(\forall x(man(x) \rightarrow mortal(x)) \wedge man(Socrates)\big) \rightarrow mortal(Socrates)$$

- There are infinitely many prime numbers.

$$\forall x \exists y \Big(y > x \wedge \forall z\big((z \neq 1 \wedge z \neq y) \rightarrow \forall w(wz \neq y)\big)\Big)$$

- The relation R is transitive. $\forall x \forall y \forall z(\ (R(x,y) \wedge R(y,z)) \rightarrow R(x,z)\ )$

First-Order Logic

Example

- All men are mortal. Socrates is a man. So Socrates is mortal.

$$\big(\forall x(man(x) \rightarrow mortal(x)) \land man(Socrates)\big) \rightarrow mortal(Socrates)$$

- There are infinitely many prime numbers.

$$\forall x \exists y \Big(y > x \land \forall z\big((z \neq 1 \land z \neq y) \rightarrow \forall w(wz \neq y)\big)\Big)$$

- The relation R is transitive. $\forall x \forall y \forall z(\ (R(x,y) \land R(y,z)) \rightarrow R(x,z) \ )$
- Let `R[name, id]` and `S[id, age]` be tables in an SQL

```
select R.name from R join S on R.id = S.id where S.age = 42
```

First-Order Logic

Example

- All men are mortal. Socrates is a man. So Socrates is mortal.

$$(\forall x(man(x) \to mortal(x)) \land man(Socrates)) \to mortal(Socrates)$$

- There are infinitely many prime numbers.

$$\forall x \exists y \Big(y > x \land \forall z((z \neq 1 \land z \neq y) \to \forall w(wz \neq y)) \Big)$$

- The relation R is transitive. $\forall x \forall y \forall z(\ (R(x,y) \land R(y,z)) \to R(x,z)\ )$
- Let `R[name, id]` and `S[id, age]` be tables in an SQL

```
select R.name from R join S on R.id = S.id where S.age = 42
```

$$\exists z(R(x,z) \land S(z,42))$$

First-Order Logic

Example

- All men are mortal. Socrates is a man. So Socrates is mortal.

$$(\forall x(man(x) \rightarrow mortal(x)) \wedge man(Socrates)) \rightarrow mortal(Socrates)$$

- There are infinitely many prime numbers.

$$\forall x \exists y \Big(y > x \wedge \forall z((z \neq 1 \wedge z \neq y) \rightarrow \forall w(wz \neq y)) \Big)$$

- The relation R is transitive. $\forall x \forall y \forall z(\ (R(x,y) \wedge R(y,z)) \rightarrow R(x,z)\ )$
- Let `R[name, id]` and `S[id, age]` be tables in an SQL
  `select R.name from R join S on R.id = S.id where S.age = 42`

$$\exists z(R(x,z) \wedge S(z,42))$$

- Fermat's Last theorem

First-Order Logic

Example

- All men are mortal. Socrates is a man. So Socrates is mortal.

$$(\forall x(man(x) \to mortal(x)) \wedge man(Socrates)) \to mortal(Socrates)$$

- There are infinitely many prime numbers.

$$\forall x \exists y \Big(y > x \wedge \forall z((z \neq 1 \wedge z \neq y) \to \forall w(wz \neq y))\Big)$$

- The relation R is transitive. $\forall x \forall y \forall z(\ (R(x,y) \wedge R(y,z)) \to R(x,z)\ )$
- Let `R[name, id]` and `S[id, age]` be tables in an SQL
  `select R.name from R join S on R.id = S.id where S.age = 42`

$$\exists z(R(x,z) \wedge S(z,42))$$

- Fermat's Last theorem

$$\forall n \forall x \forall y \big(n > 2 \quad \to \quad \forall z(x^n + y^n \neq z^n)\big)$$

First-Order Logic

What is **NOT** expressible with FOL:

- “Adam is an ancestor of Socrates.” (using $isParentOf$)
  Attempts:

First-Order Logic

What is **NOT** expressible with FOL:

- "Adam is an ancestor of Socrates." (using $isParentOf$)
  Attempts:
  - $\exists x_1, \ldots, x_n (isParentOf(Adam, x_1) \land \ldots \land isParentOf(x_n, Socrates))$

    [n is bounded]

First-Order Logic

What is **NOT** expressible with FOL:

- "Adam is an ancestor of Socrates." (using $isParentOf$)
  Attempts:
  - $\exists x_1, \ldots, x_n \big(isParentOf(Adam, x_1) \wedge \ldots \wedge isParentOf(x_n, Socrates)\big)$

    [n is bounded]
  - $$\exists_2^{\text{fin}} X \Big(Socrates \in X \wedge Adam \in X \wedge \big(\forall y \in X : (\exists z \in X : isParentOf(z, y)) \vee y = Adam\big)\Big)$$

    [$\exists_2^{\text{fin}}$ — second-order finite quantification, cf. MSO]

First-Order Logic

What is **NOT** expressible with FOL:

- "Adam is an ancestor of Socrates." (using $\mathit{isParentOf}$)
  Attempts:
  - $\exists x_1, \ldots, x_n \big(\mathit{isParentOf}(\mathit{Adam}, x_1) \wedge \ldots \wedge \mathit{isParentOf}(x_n, \mathit{Socrates})\big)$

    [n is bounded]
  - $$\exists_2^{\mathsf{fin}} X \Big(\mathit{Socrates} \in X \wedge \mathit{Adam} \in X \wedge \big(\forall y \in X : (\exists z \in X : \mathit{isParentOf}(z, y)) \vee y = \mathit{Adam}\big)\Big)$$

    [$\exists_2^{\mathsf{fin}}$ — second-order finite quantification, cf. MSO]
  - $$\mathit{isAncestorOf}(x, y) \overset{\text{def}}{\Leftrightarrow} \mathit{isParentOf}(x, y) \vee \big(\exists z : \mathit{isAncestorOf}(x, z) \wedge \mathit{isParentOf}(z, y)\big)$$

    [recursive predicate]

First-Order Logic

What is **NOT** expressible with FOL:

- "Adam is an ancestor of Socrates." (using $\mathit{isParentOf}$)
  Attempts:
  - $\exists x_1, \ldots, x_n \big(\mathit{isParentOf}(\mathit{Adam}, x_1) \land \ldots \land \mathit{isParentOf}(x_n, \mathit{Socrates})\big)$
    [n is bounded]
  - $\exists_2^{\text{fin}} X \Big(\mathit{Socrates} \in X \land \mathit{Adam} \in X \land \big(\forall y \in X :$
    $(\exists z \in X : \mathit{isParentOf}(z, y)) \lor y = \mathit{Adam}\big)\Big)$
    [$\exists_2^{\text{fin}}$ — second-order finite quantification, cf. MSO]
  - $\mathit{isAncestorOf}(x, y) \overset{\text{def}}{\Leftrightarrow} \mathit{isParentOf}(x, y) \lor$
    $\big(\exists z : \mathit{isAncestorOf}(x, z) \land \mathit{isParentOf}(z, y)\big)$
    [recursive predicate]

- "Anakin is more likely than Gandalf the father of Luke."
  Attempts:
  - ?!$#dk*#R&Q

Syntax

Syntax:

- **Alphabet**:
  - logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, (\cdots)$ (from PL)
  - variables: $x, y, \ldots, x_1, x_2, \ldots$ (hold elements of a universe)
  - quantifiers: $\forall, \exists$

Syntax

Syntax:

- **Alphabet**:
  - logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, (\cdots)$ (from PL)
  - variables: $x, y, \ldots, x_1, x_2, \ldots$ (hold elements of a universe)
  - quantifiers: $\forall, \exists$
  - function symbols (with $_{/arity}$): $f_{/2}$, $+_{/2}$, $\sin_{/1}$, $fatherOf_{/1}$, $S_{/1}$, $\pi_{/0}$, $42_{/0}$,...
    - nullary functions (arity 0): constants
    - to be used as, e.g., $f(a,3)$, $+(40,2)$, $\sin(S(x))$, $fatherOf(Luke)$, $\pi()$
    - we often simplify the notation: $+(40,2) \mapsto 40+2$, $\pi() \mapsto \pi$, ...

Syntax

Syntax:

- **Alphabet**:
  - logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, (\cdots)$ (from PL)
  - variables: $x, y, \ldots, x_1, x_2, \ldots$ (hold elements of a universe)
  - quantifiers: $\forall, \exists$
  - function symbols (with $_{/arity}$): $f_{/2}$, $+_{/2}$, $\sin_{/1}$, $fatherOf_{/1}$, $S_{/1}$, $\pi_{/0}$, $42_{/0}$,...
    - nullary functions (arity 0): constants
    - to be used as, e.g., $f(a, 3)$, $+(40, 2)$, $\sin(S(x))$, $fatherOf(Luke)$, $\pi()$
    - we often simplify the notation: $+(40, 2) \mapsto 40 + 2$, $\pi() \mapsto \pi$, ...
  - predicate symbols (with $_{/arity}$): $p_{/3}$, $isFatherOf_{/2}$, $isJedi_{/1}$, $<_{/2}$, ...
    - to be used as, e.g., $p(a, x, 9)$, $isFatherOf(Anakin, Luke)$, $isJedi(Anakin)$, $<(x, \pi)$
    - we often simplify the notation: $<(x, \pi) \mapsto x < \pi$, ...
  - predicate symbol of equality $=_{/2}$

Syntax

Syntax:

- **Alphabet**:
  - logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, (\cdots)$ (from PL)
  - variables: $x, y, \ldots, x_1, x_2, \ldots$ (hold elements of a universe)
  - quantifiers: $\forall, \exists$
  - function symbols (with $_{/arity}$): $f_{/2}$, $+_{/2}$, $\sin_{/1}$, $fatherOf_{/1}$, $S_{/1}$, $\pi_{/0}$, $42_{/0}$,...
    - nullary functions (arity 0): constants
    - to be used as, e.g., $f(a, 3)$, $+(40, 2)$, $\sin(S(x))$, $fatherOf(Luke)$, $\pi()$
    - we often simplify the notation: $+(40, 2) \mapsto 40 + 2$, $\pi() \mapsto \pi$, ...
  - predicate symbols (with $_{/arity}$): $p_{/3}$, $isFatherOf_{/2}$, $isJedi_{/1}$, $<_{/2}$, ...
    - to be used as, e.g., $p(a, x, 9)$, $isFatherOf(Anakin, Luke)$, $isJedi(Anakin)$, $<(x, \pi)$
    - we often simplify the notation: $<(x, \pi) \mapsto x < \pi$, ...
  - predicate symbol of equality $=_{/2}$
- Signature $\langle \mathcal{F}, \mathcal{P} \rangle$ = function symbols + predicate symbols
  - language: given by the signature

First-Order Logic — Syntax

Example

- the language of order theory: $\langle \mathcal{F} = \emptyset, \mathcal{P} = \{<_{/2}\}\rangle$
  - no function symbol
  - one binary predicate symbol $<$

First-Order Logic — Syntax

Example

- the language of order theory: $\langle \mathcal{F} = \emptyset, \mathcal{P} = \{<_{/2}\}\rangle$
  - no function symbol
  - one binary predicate symbol $<$
- the language of group theory: $\langle \mathcal{F} = \{\cdot_{/2}, e_{/0}\}, \mathcal{P} = \emptyset\rangle$
  - binary function symbol $\cdot$ (group multiplication)
  - nullary function symbol e for neutral element

First-Order Logic — Syntax

Example

- the language of order theory: $\langle \mathcal{F} = \emptyset, \mathcal{P} = \{<_{/2}\}\rangle$
  - no function symbol
  - one binary predicate symbol $<$
- the language of group theory: $\langle \mathcal{F} = \{\cdot_{/2}, e_{/0}\}, \mathcal{P} = \emptyset\rangle$
  - binary function symbol $\cdot$ (group multiplication)
  - nullary function symbol e for neutral element
- the language of set theory: $\langle \mathcal{F} = \emptyset, \mathcal{P} = \{\in_{/2}\}\rangle$

First-Order Logic — Syntax

Example

- the language of order theory: $\langle \mathcal{F} = \emptyset, \mathcal{P} = \{<_{/2}\}\rangle$
  - no function symbol
  - one binary predicate symbol $<$
- the language of group theory: $\langle \mathcal{F} = \{\cdot_{/2}, e_{/0}\}, \mathcal{P} = \emptyset\rangle$
  - binary function symbol $\cdot$ (group multiplication)
  - nullary function symbol e for neutral element
- the language of set theory: $\langle \mathcal{F} = \emptyset, \mathcal{P} = \{\in_{/2}\}\rangle$
- the language of theory of arrays: $\langle \mathcal{F} = \{\cdot[\cdot]^r_{/2}, \cdot[\cdot,\cdot]^w_{/3}\}, \mathcal{P} = \emptyset\rangle$
  - binary function symbol for reading from array $\cdot[\cdot]^r$, e.g., $A[i]^r$
  - ternary function symbol for writing into array $\cdot[\cdot,\cdot]^w$, e.g., $A[i,y]^w$ (writing y at index i in array A)

First-Order Logic — Syntax

Example

- the language of order theory: $\langle \mathcal{F} = \emptyset, \mathcal{P} = \{<_{/2}\}\rangle$
  - no function symbol
  - one binary predicate symbol $<$
- the language of group theory: $\langle \mathcal{F} = \{\cdot_{/2}, e_{/0}\}, \mathcal{P} = \emptyset\rangle$
  - binary function symbol $\cdot$ (group multiplication)
  - nullary function symbol e for neutral element
- the language of set theory: $\langle \mathcal{F} = \emptyset, \mathcal{P} = \{\in_{/2}\}\rangle$
- the language of theory of arrays: $\langle \mathcal{F} = \{\cdot[\cdot]^r_{/2}, \cdot[\cdot,\cdot]^w_{/3}\}, \mathcal{P} = \emptyset\rangle$
  - binary function symbol for reading from array $\cdot[\cdot]^r$, e.g., $A[i]^r$
  - ternary function symbol for writing into array $\cdot[\cdot,\cdot]^w$, e.g., $A[i,y]^w$ (writing y at index i in array A)
- the language of theory of lists: $\langle \mathcal{F} = \{\mathsf{cons}_{/2}, \mathsf{car}_{/1}, \mathsf{cdr}_{/1}\}, \mathcal{P} = \{\mathsf{atom}_{/1}\}\rangle$

First-Order Logic — Syntax

Example

- the language of order theory: $\langle \mathcal{F} = \emptyset, \mathcal{P} = \{<_{/2}\}\rangle$
  - no function symbol
  - one binary predicate symbol $<$
- the language of group theory: $\langle \mathcal{F} = \{\cdot_{/2}, e_{/0}\}, \mathcal{P} = \emptyset\rangle$
  - binary function symbol $\cdot$ (group multiplication)
  - nullary function symbol e for neutral element
- the language of set theory: $\langle \mathcal{F} = \emptyset, \mathcal{P} = \{\in_{/2}\}\rangle$
- the language of theory of arrays: $\langle \mathcal{F} = \{\cdot[\cdot]^r_{/2}, \cdot[\cdot,\cdot]^w_{/3}\}, \mathcal{P} = \emptyset\rangle$
  - binary function symbol for reading from array $\cdot[\cdot]^r$, e.g., $A[i]^r$
  - ternary function symbol for writing into array $\cdot[\cdot,\cdot]^w$, e.g., $A[i,y]^w$ (writing y at index i in array A)
- the language of theory of lists: $\langle \mathcal{F} = \{\mathsf{cons}_{/2}, \mathsf{car}_{/1}, \mathsf{cdr}_{/1}\}, \mathcal{P} = \{\mathsf{atom}_{/1}\}\rangle$
- the language of elementary (so-called Peano) aritmetic: $\langle \mathcal{F} = \{0_{/0}, S_{/1}, +_{/2}, \cdot_{/2}\}, \mathcal{P} = \emptyset\rangle$

First-Order Logic — syntax

Grammar: formulae are composed of *terms*

- term (it will hold a value from the universe):

  $$t ::= x \quad | \quad f(t_1, \dots, t_n)$$

  where $x \in \mathbb{X}$ and $f_{/n}$ is a function symbol
  (the special case of a constant $c_{/0}$ is also a term)
- ground term: a term with no variables

First-Order Logic — syntax

Grammar: formulae are composed of *terms*

- term (it will hold a value from the universe):

$$t ::= x \quad | \quad f(t_1, \ldots, t_n)$$

  where $x \in \mathbb{X}$ and $f_{/n}$ is a function symbol
  (the special case of a constant $c_{/0}$ is also a term)

- ground term: a term with no variables
- examples of terms:
  - x, 5, $f(x, 2)$, $40 + 2$, $\mathsf{car}(\mathsf{cons}(x, y))$, $\mathit{head}(\texttt{"abc"})$, $\sin y$

First-Order Logic — syntax

Grammar (cont.):

- atomic formula:

$$\varphi_{atom} ::= p(t_1, \ldots, t_n)$$

  for a predicate symbol $p_{/n}$ and terms $t_1, \ldots, t_n$
  (also for p being the equality symbol $=_{/2}$)

First-Order Logic — syntax

Grammar (cont.):

- atomic formula:

$$\varphi_{atom} ::= p(t_1, \ldots, t_n)$$

  for a predicate symbol $p_{/n}$ and terms $t_1, \ldots, t_n$
  (also for p being the equality symbol $=_{/2}$)

- formula:

$$\begin{aligned}\varphi ::= \varphi_{atom} \mid (\neg\varphi_1) \mid \\ (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2) \mid \\ (\varphi_1 \rightarrow \varphi_2) \mid (\varphi_1 \leftrightarrow \varphi_2) \mid \\ (\forall x \varphi_1) \mid (\exists x \varphi_1)\end{aligned}$$

  - where x is a variable from the set of variables $\mathbb{X}$
  - (parentheses are often omitted)

First-Order Logic — syntax

Grammar (cont.):

- atomic formula:

$$\varphi_{atom} ::= p(t_1, \dots, t_n)$$

  for a predicate symbol $p_{/n}$ and terms $t_1, \dots, t_n$
  (also for p being the equality symbol $=_{/2}$)

- formula:

$$\begin{aligned}\varphi ::= \varphi_{atom} \mid (\neg\varphi_1) \mid \\ (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2) \mid \\ (\varphi_1 \rightarrow \varphi_2) \mid (\varphi_1 \leftrightarrow \varphi_2) \mid \\ (\forall x \varphi_1) \mid (\exists x \varphi_1)\end{aligned}$$

  - where x is a variable from the set of variables $\mathbb{X}$
  - (parentheses are often omitted)

- examples of formulae:
  - $\exists x(40 + x = 42 \wedge 40 \cdot x = 80)$,
  - $\forall x(\tan(x) = \frac{\sin(x)}{\cos(x)})$,
  - $\mathsf{atom}(\mathsf{car}(\mathsf{cons}(x, y)))$,
  - $\forall x(\exists y(x = y \cdot y \vee x = -y \cdot y))$

First-Order Logic — Variables

Variables in formulae:

- **bound**: occur in the scope of a quantifier
  - e.g. $\texttt{BOUND}(\exists x\big(x = 4 \land \neg(y = 5)\big)) = \{x\}$

First-Order Logic — Variables

Variables in formulae:

- **bound**: occur in the scope of a quantifier
  - e.g. $\texttt{BOUND}(\exists x\big(x = 4 \land \neg(y = 5)\big)) = \{x\}$
- **free**: there is an occurrence not bound by any quantifier
  - e.g. $\texttt{FREE}(x = 4 \land \exists y(y = 5)) = \{x\}$

First-Order Logic — Variables

Variables in formulae:

- **bound**: occur in the scope of a quantifier
  - e.g. $\texttt{BOUND}(\exists x\big(x = 4 \wedge \neg(y = 5)\big)) = \{x\}$
- **free**: there is an occurrence not bound by any quantifier
  - e.g. $\texttt{FREE}(x = 4 \wedge \exists y(y = 5)) = \{x\}$
- $\texttt{FREE}(\cdot)$ and $\texttt{BOUND}(\cdot)$ are symbols of the metalanguage
- a variable can occur both bound and free in a formula

First-Order Logic — Variables

Variables in formulae:

- **bound**: occur in the scope of a quantifier
  - e.g. $\texttt{BOUND}(\exists x\big(x = 4 \wedge \neg(y = 5)\big)) = \{x\}$
- **free**: there is an occurrence not bound by any quantifier
  - e.g. $\texttt{FREE}(x = 4 \wedge \exists y(y = 5)) = \{x\}$
- $\texttt{FREE}(\cdot)$ and $\texttt{BOUND}(\cdot)$ are symbols of the metalanguage
- a variable can occur both bound and free in a formula

Example

$$\forall x\big(\, p(f(x), y) \rightarrow \forall y(p(f(x), y))\,\big)$$

- x only occurs bound
- y occurs both free (antecedent) and bound (consequent)

First-Order Logic — Variables

Variables in formulae:

- **bound**: occur in the scope of a quantifier
  - e.g. $\mathtt{BOUND}(\exists x\big(x = 4 \land \neg(y = 5)\big)) = \{x\}$
- **free**: there is an occurrence not bound by any quantifier
  - e.g. $\mathtt{FREE}(x = 4 \land \exists y(y = 5)) = \{x\}$
- $\mathtt{FREE}(\cdot)$ and $\mathtt{BOUND}(\cdot)$ are symbols of the metalanguage
- a variable can occur both bound and free in a formula

Example

$$\forall x\big(\, p(f(x), y) \rightarrow \forall y(p(f(x), y))\, \big)$$

- x only occurs bound
- y occurs both free (antecedent) and bound (consequent)

- we often write $\varphi(x_1, \dots, x_n)$ when $\mathtt{FREE}(\varphi) \subseteq \{x_1, \dots, x_n\}$
  - $x_1, \dots, x_n$ serve as the "interface" of φ

First-Order Logic — Variables

Variables in formulae:

- **bound**: occur in the scope of a quantifier
  - e.g. $\texttt{BOUND}(\exists x\big(x = 4 \wedge \neg(y = 5)\big)) = \{x\}$
- **free**: there is an occurrence not bound by any quantifier
  - e.g. $\texttt{FREE}(x = 4 \wedge \exists y(y = 5)) = \{x\}$
- $\texttt{FREE}(\cdot)$ and $\texttt{BOUND}(\cdot)$ are symbols of the metalanguage
- a variable can occur both bound and free in a formula

Example

$$\forall x\big(\, p(f(x), y) \rightarrow \forall y(p(f(x), y))\, \big)$$

- x only occurs bound
- y occurs both free (antecedent) and bound (consequent)

- we often write $\varphi(x_1, \ldots, x_n)$ when $\texttt{FREE}(\varphi) \subseteq \{x_1, \ldots, x_n\}$
  - $x_1, \ldots, x_n$ serve as the "interface" of φ
- φ is ground (or closed) if $\texttt{FREE}(\varphi) = \emptyset$

Semantics

Semantics of FOL:

- so far, the symbols *did not have any meaning*!
- more complicated than for PL

Semantics

Semantics of FOL:

- so far, the symbols *did not have any meaning*!
- more complicated than for PL

Interpretation $I = (D_I, \alpha_I)$:

Semantics

Semantics of FOL:

- so far, the symbols *did not have any meaning*!
- more complicated than for PL

Interpretation $I = (D_I, \alpha_I)$:

- provides the meaning to the symbols
  - a formula may hold in one interpretation and not hold in another
- domain (universe) of discourse D_I: a non-empty set of elements
  - e.g., $\mathbb{N}$, $\{0, 1, 2, 3, 4\}$, $\mathbb{R}^3$, People, $\mathit{List}[\mathbb{N}]$, Σ^*, $\ldots$

Semantics

Semantics of FOL:

- so far, the symbols *did not have any meaning*!
- more complicated than for PL

Interpretation $I = (D_I, \alpha_I)$:

- provides the meaning to the symbols
  - a formula may hold in one interpretation and not hold in another
- domain (universe) of discourse D_I: a non-empty set of elements
  - e.g., $\mathbb{N}$, $\{0, 1, 2, 3, 4\}$, $\mathbb{R}^3$, $People$, $List[\mathbb{N}]$, Σ^*, ...
- assignment α_I:
  - for every function symbol $f_{/n}$, a function $f_I\colon \overbrace{D_I \times \ldots \times D_I}^{n} \to D_I$
    - e.g., $I(+) = \{(0,0) \mapsto 0, (0,1) \mapsto 1, (1,0) \mapsto 1, (1,1) \mapsto 2, \ldots\}$
    - for constants, this gives us one value, e.g., $I(\pi) = \{() \mapsto 3.1415926\ldots\}$

Semantics

Semantics of FOL:

- so far, the symbols *did not have any meaning*!
- more complicated than for PL

Interpretation $I = (D_I, \alpha_I)$:

- provides the meaning to the symbols
  - a formula may hold in one interpretation and not hold in another
- domain (universe) of discourse D_I: a non-empty set of elements
  - e.g., $\mathbb{N}$, $\{0,1,2,3,4\}$, $\mathbb{R}^3$, People, $\mathit{List}[\mathbb{N}]$, Σ^*, ...
- assignment α_I:
  - for every function symbol $f_{/n}$, a function $f_I \colon \overbrace{D_I \times \ldots \times D_I}^{n} \to D_I$
    - e.g., $I(+) = \{(0,0) \mapsto 0, (0,1) \mapsto 1, (1,0) \mapsto 1, (1,1) \mapsto 2, \ldots\}$
    - for constants, this gives us one value, e.g., $I(\pi) = \{() \mapsto 3.1415926\ldots\}$
  - for every predicate symbol $p_{/n}$, a relation $p_I \subseteq \overbrace{D_I \times \ldots \times D_I}^{n}$
    - e.g., $I(<) = \{(0,1), (0,2), (1,2), \ldots\}$
    - e.g., $I(\mathit{even}_{/1}) = \{0, 2, 4, \ldots\}$
    - e.g., $I(\mathit{edge}_{/2}) = \{(v_1, v_2), (v_2, v_3), \ldots\}$

Semantics

Semantics of FOL:

- so far, the symbols *did not have any meaning*!
- more complicated than for PL

Interpretation $I = (D_I, \alpha_I)$:

- provides the meaning to the symbols
  - a formula may hold in one interpretation and not hold in another
- domain (universe) of discourse D_I: a non-empty set of elements
  - e.g., $\mathbb{N}$, $\{0,1,2,3,4\}$, $\mathbb{R}^3$, People, $\mathit{List}[\mathbb{N}]$, Σ^*, ...
- assignment α_I:
  - for every function symbol $f_{/n}$, a function $f_I \colon \overbrace{D_I \times \ldots \times D_I}^{n} \to D_I$
    - e.g., $I(+) = \{(0,0) \mapsto 0, (0,1) \mapsto 1, (1,0) \mapsto 1, (1,1) \mapsto 2, \ldots\}$
    - for constants, this gives us one value, e.g., $I(\pi) = \{() \mapsto 3.1415926\ldots\}$
  - for every predicate symbol $p_{/n}$, a relation $p_I \subseteq \overbrace{D_I \times \ldots \times D_I}^{n}$
    - e.g., $I(<) = \{(0,1), (0,2), (1,2), \ldots\}$
    - e.g., $I(\mathit{even}_{/1}) = \{0, 2, 4, \ldots\}$
    - e.g., $I(\mathit{edge}_{/2}) = \{(v_1, v_2), (v_2, v_3), \ldots\}$
  - for every variable $x \in \mathbb{X}$ a value from D_I, e.g., $I(x) = 42$

First-Order Logic — Interpretations

Example

Examples of interpretations of the language with the signature $\langle \mathcal{F} = \{+_{/2}\}, \mathcal{P} = \emptyset \rangle$:

- Addition in $\mathbb{N}$: $D_I = \mathbb{N}$ where
  - $I(+) = (+_{\mathbb{N}})$ (addition of natural numbers)

First-Order Logic — Interpretations

Example

Examples of interpretations of the language with the signature $\langle \mathcal{F} = \{+_{/2}\}, \mathcal{P} = \emptyset \rangle$:

- Addition in $\mathbb{N}$: $D_I = \mathbb{N}$ where
  - $I(+) = (+_\mathbb{N})$ (addition of natural numbers)
- Addition in $\mathbb{R}^3$: $D_I = \mathbb{R}^3$ where
  - $I(+) = \{([x_1, y_1, z_1], [x_2, y_2, z_2]) \mapsto [x_1 +_\mathbb{R} x_2, y_1 +_\mathbb{R} y_2, z_1 +_\mathbb{R} z_2] \mid x_1, \ldots, z_2 \in \mathbb{R}\}$

First-Order Logic — Interpretations

Example

Examples of interpretations of the language with the signature $\langle \mathcal{F} = \{+_{/2}\}, \mathcal{P} = \emptyset \rangle$:

- Addition in $\mathbb{N}$: $D_I = \mathbb{N}$ where
  - $I(+) = (+_{\mathbb{N}})$ (addition of natural numbers)
- Addition in $\mathbb{R}^3$: $D_I = \mathbb{R}^3$ where
  - $I(+) = \{([x_1, y_1, z_1], [x_2, y_2, z_2]) \mapsto [x_1 +_{\mathbb{R}} x_2, y_1 +_{\mathbb{R}} y_2, z_1 +_{\mathbb{R}} z_2] \mid x_1, \ldots, z_2 \in \mathbb{R}\}$
- Disjunction in a Boolean algebra: $D_I = \{0, 1\}$ where
  - $I(+) = \{(0, 0) \mapsto 0, (0, 1) \mapsto 1, (1, 0) \mapsto 1, (1, 1) \mapsto 1\}$

First-Order Logic — Interpretations

Example

Examples of interpretations of the language with the signature $\langle \mathcal{F} = \{+_{/2}\}, \mathcal{P} = \emptyset \rangle$:

- Addition in $\mathbb{N}$: $D_I = \mathbb{N}$ where
  - $I(+) = (+_{\mathbb{N}})$ (addition of natural numbers)
- Addition in $\mathbb{R}^3$: $D_I = \mathbb{R}^3$ where
  - $I(+) = \{([x_1, y_1, z_1], [x_2, y_2, z_2]) \mapsto [x_1 +_{\mathbb{R}} x_2, y_1 +_{\mathbb{R}} y_2, z_1 +_{\mathbb{R}} z_2] \mid x_1, \ldots, z_2 \in \mathbb{R}\}$
- Disjunction in a Boolean algebra: $D_I = \{0, 1\}$ where
  - $I(+) = \{(0,0) \mapsto 0, (0,1) \mapsto 1, (1,0) \mapsto 1, (1,1) \mapsto 1\}$
- Modular addition in $\{0, 1, 2\}$: $D_I = \{0, 1, 2\}$ where
  - $I(+) = \{(x, y) \mapsto (x + y \mod 3) \mid x, y \in \{0, 1, 2\}\}$

First-Order Logic — Interpretations

Example

Language with the signature $\langle \mathcal{F} = \{+_{/2}, \cdot_{/2}, -_{/1}\}, \mathcal{P} = \{E_{/1}\}\rangle$:

- Addition, multiplication, and opposite number in $\mathbb{Z}$: $D_I = \mathbb{Z}$
  - $I(+) = (+_{\mathbb{Z}})$ (addition of natural numbers)
  - $I(\cdot) = (\cdot_{\mathbb{Z}})$ (multiplication of natural numbers)
  - $I(-) = \{x \mapsto (0 -_{\mathbb{Z}} x) \mid x \in \mathbb{Z}\}$
  - $I(E) = \{\ldots, -4, -2, 0, 2, 4, \ldots\}$

First-Order Logic — Interpretations

Example

Language with the signature $\langle \mathcal{F} = \{+_{/2}, \cdot_{/2}, -_{/1}\}, \mathcal{P} = \{E_{/1}\}\rangle$:

- Addition, multiplication, and opposite number in $\mathbb{Z}$: $D_I = \mathbb{Z}$
  - $I(+) = (+_{\mathbb{Z}})$ (addition of natural numbers)
  - $I(\cdot) = (\cdot_{\mathbb{Z}})$ (multiplication of natural numbers)
  - $I(-) = \{x \mapsto (0 -_{\mathbb{Z}} x) \mid x \in \mathbb{Z}\}$
  - $I(E) = \{\ldots, -4, -2, 0, 2, 4, \ldots\}$
- Disjunction, conjunction, and negation in a Boolean algebra: $D_I = \{0, 1\}$
  - $I(+) = \{(0,0) \mapsto 0, (0,1) \mapsto 1, (1,0) \mapsto 1, (1,1) \mapsto 1\}$
  - $I(\cdot) = \{(0,0) \mapsto 0, (0,1) \mapsto 0, (1,0) \mapsto 0, (1,1) \mapsto 1\}$
  - $I(-) = \{0 \mapsto 1, 1 \mapsto 0\}$
  - $I(E) = \{0\}$

First-Order Logic — Interpretations

Example

Language with the signature $\langle \mathcal{F} = \{+_{/2}, \cdot_{/2}, -_{/1}\}, \mathcal{P} = \{E_{/1}\}\rangle$:

- Addition, multiplication, and opposite number in $\mathbb{Z}$: $D_I = \mathbb{Z}$
  - $I(+) = (+_{\mathbb{Z}})$ (addition of natural numbers)
  - $I(\cdot) = (\cdot_{\mathbb{Z}})$ (multiplication of natural numbers)
  - $I(-) = \{x \mapsto (0 -_{\mathbb{Z}} x) \mid x \in \mathbb{Z}\}$
  - $I(E) = \{\ldots, -4, -2, 0, 2, 4, \ldots\}$
- Disjunction, conjunction, and negation in a Boolean algebra: $D_I = \{0, 1\}$
  - $I(+) = \{(0,0) \mapsto 0, (0,1) \mapsto 1, (1,0) \mapsto 1, (1,1) \mapsto 1\}$
  - $I(\cdot) = \{(0,0) \mapsto 0, (0,1) \mapsto 0, (1,0) \mapsto 0, (1,1) \mapsto 1\}$
  - $I(-) = \{0 \mapsto 1, 1 \mapsto 0\}$
  - $I(E) = \{0\}$
- Union, concatenation, and iteration of sets of words over Σ:

First-Order Logic — Interpretations

Example

Language with the signature $\langle \mathcal{F} = \{+_{/2}, \cdot_{/2}, -_{/1}\}, \mathcal{P} = \{E_{/1}\}\rangle$:

- Addition, multiplication, and opposite number in $\mathbb{Z}$: $D_I = \mathbb{Z}$
  - $I(+) = (+_\mathbb{Z})$ (addition of natural numbers)
  - $I(\cdot) = (\cdot_\mathbb{Z})$ (multiplication of natural numbers)
  - $I(-) = \{x \mapsto (0 -_\mathbb{Z} x) \mid x \in \mathbb{Z}\}$
  - $I(E) = \{\dots, -4, -2, 0, 2, 4, \dots\}$
- Disjunction, conjunction, and negation in a Boolean algebra: $D_I = \{0, 1\}$
  - $I(+) = \{(0,0) \mapsto 0, (0,1) \mapsto 1, (1,0) \mapsto 1, (1,1) \mapsto 1\}$
  - $I(\cdot) = \{(0,0) \mapsto 0, (0,1) \mapsto 0, (1,0) \mapsto 0, (1,1) \mapsto 1\}$
  - $I(-) = \{0 \mapsto 1, 1 \mapsto 0\}$
  - $I(E) = \{0\}$
- Union, concatenation, and iteration of sets of words over Σ: $D_I = 2^{\Sigma^*}$
  - $I(+) = \{(x, y) \mapsto (x \cup y) \mid x, y \subseteq \Sigma^*\}$
  - $I(\cdot) = \big\{(x, y) \mapsto \{uv \mid u \in x, v \in y\} \mid x, y \subseteq \Sigma^*\big\}$
  - $I(-) = \big\{x \mapsto \bigcup_{i \geq 0}\{u^i \mid u \in x\} \mid x \subseteq \Sigma^*\big\}$
  - $I(E) = \big\{\{\epsilon\}\big\}$

Semantics

Truth value: inductive definition:

- Terms: evaluate their value recursively

$$I[f(t_1, \ldots, t_n)] \overset{\text{def}}{=} I[f](I[t_1], \ldots, I[t_n])$$

Semantics

Truth value: inductive definition:

- Terms: evaluate their value recursively

$$I[f(t_1, \ldots, t_n)] \stackrel{\text{def}}{=} I[f](I[t_1], \ldots, I[t_n])$$

Then: $I \models p(t_1, \ldots, t_n)$ iff $(I[t_1], \ldots, I[t_n]) \in I[p]$

Semantics

Truth value: inductive definition:

- Terms: evaluate their value recursively

$$I[f(t_1, \ldots, t_n)] \stackrel{\text{def}}{=} I[f](I[t_1], \ldots, I[t_n])$$

Then: $I \models p(t_1, \ldots, t_n)$ iff $(I[t_1], \ldots, I[t_n]) \in I[p]$
(for equality: $I \models (t_1 = t_2)$ iff $I[t_1]$ and $I[t_2]$ denote the same element from D_I)

Semantics

Truth value: inductive definition:

- Terms: evaluate their value recursively

$$I[f(t_1, \ldots, t_n)] \stackrel{\text{def}}{=} I[f](I[t_1], \ldots, I[t_n])$$

  Then: $I \models p(t_1, \ldots, t_n)$ iff $(I[t_1], \ldots, I[t_n]) \in I[p]$
  (for equality: $I \models (t_1 = t_2)$ iff $I[t_1]$ and $I[t_2]$ denote the same element from D_I)

- logical connectives (same as for PL):
  - $I \models \neg\psi$ iff $I \not\models \psi$
  - $I \models \psi_1 \wedge \psi_2$ iff $I \models \psi_1$ and $I \models \psi_2$
  - $I \models \psi_1 \vee \psi_2$ iff $I \models \psi_1$ or $I \models \psi_2$
  - $I \models \psi_1 \rightarrow \psi_2$ iff, if $I \models \psi_1$ then $I \models \psi_2$
  - $I \models \psi_1 \leftrightarrow \psi_2$ iff $I \models \psi_1$ and $I \models \psi_2$, or $I \not\models \psi_1$ and $I \not\models \psi_2$

Semantics

Truth value: inductive definition:

- Terms: evaluate their value recursively

$$I[f(t_1,\dots,t_n)] \stackrel{\text{def}}{=} I[f](I[t_1],\dots,I[t_n])$$

  Then: $I \models p(t_1,\dots,t_n)$ iff $(I[t_1],\dots,I[t_n]) \in I[p]$
  (for equality: $I \models (t_1 = t_2)$ iff $I[t_1]$ and $I[t_2]$ denote the same element from D_I)

- logical connectives (same as for PL):
  - $I \models \neg\psi$ iff $I \not\models \psi$
  - $I \models \psi_1 \wedge \psi_2$ iff $I \models \psi_1$ and $I \models \psi_2$
  - $I \models \psi_1 \vee \psi_2$ iff $I \models \psi_1$ or $I \models \psi_2$
  - $I \models \psi_1 \to \psi_2$ iff, if $I \models \psi_1$ then $I \models \psi_2$
  - $I \models \psi_1 \leftrightarrow \psi_2$ iff $I \models \psi_1$ and $I \models \psi_2$, or $I \not\models \psi_1$ and $I \not\models \psi_2$

- quantifiers: let $I \lhd \{x \mapsto v\}$ denote an interpretation obtained from I by substituting $x \mapsto ?$ by $x \mapsto v$ in I ($I \lhd \{x \mapsto v\}$ is a variant)
  - $I \models \forall x\, \varphi$ iff for all $v \in D_I$ we have $I \lhd \{x \mapsto v\} \models \varphi$
  - $I \models \exists x\, \varphi$ iff there exists $v \in D_I$ such that $I \lhd \{x \mapsto v\} \models \varphi$

First-Order Logic — Semantics

Example

Let L be the language with the signature $\langle \mathcal{F} = \{+_{/2}, -_{/1}\}, \mathcal{P} = \{Z_{/1}\}\rangle$ and its interpretation I_L with $D_{I_L} = \{a, b, c\}$ and

$$I_L(+) = \begin{array}{c|ccc} & a & b & c \\ \hline a & a & b & c \\ b & b & c & a \\ c & c & a & b \end{array} \qquad I_L(-) = \{a \mapsto a, b \mapsto c, c \mapsto b\} \qquad I_L(Z) = \{a, b\}$$

Does the following formula hold in I_L: $\forall x \forall y (Z(x) \rightarrow x + y = -y)$?

First-Order Logic

model of a formula φ:

- is an interpretation I such that $I \models \varphi$

model of a formula φ:

- is an interpretation I such that $I \models \varphi$

satisfiability:

- formula φ is satisfiable if it has a model
- i.e., there is an interpretation I with the domain D_I, valuation of function symbols, predicate symbols, and variables α_I such that $I \models \varphi$

First-Order Logic

logical validity:

- formula φ is (logically) valid if it holds in **all** interpretations of the given language, i.e., for all domains, and valuations of function and predicate symbols and variables
- denoted as $\models \varphi$
- (equivalent to the notion of *tautology* in propositional logic)

First-Order Logic

logical validity:

- formula φ is (logically) valid if it holds in **all** interpretations of the given language, i.e., for all domains, and valuations of function and predicate symbols and variables
- denoted as $\models \varphi$
- (equivalent to the notion of *tautology* in propositional logic)

Example

Is the following formula valid?

$$\varphi\colon 1 + 1 = 2$$

First-Order Logic

logical validity:

- formula φ is (logically) valid if it holds in **all** interpretations of the given language, i.e., for all domains, and valuations of function and predicate symbols and variables
- denoted as $\models \varphi$
- (equivalent to the notion of *tautology* in propositional logic)

Example

Is the following formula valid?

$$\varphi : 1 + 1 = 2$$

why?

First-Order Logic

logical validity:

- formula φ is (logically) valid if it holds in **all** interpretations of the given language, i.e., for all domains, and valuations of function and predicate symbols and variables
- denoted as $\models \varphi$
- (equivalent to the notion of *tautology* in propositional logic)

Example

Is the following formula valid?

$$\varphi\colon 1 + 1 = 2$$

why?

- there is an interpretation I where φ does not hold
  - e.g., $D_I = \mathbb{N}$ with $I(+) = \{\ldots, (1,1) \mapsto 3, \ldots\}$

First-Order Logic

logical validity:

- formula φ is (logically) valid if it holds in **all** interpretations of the given language, i.e., for all domains, and valuations of function and predicate symbols and variables
- denoted as $\models \varphi$
- (equivalent to the notion of *tautology* in propositional logic)

Example

Is the following formula valid?

$$\varphi\colon 1 + 1 = 2$$

why?

- there is an interpretation I where φ does not hold
  - e.g., $D_I = \mathbb{N}$ with $I(+) = \{\ldots, (1,1) \mapsto 3, \ldots\}$
- we often want to restrict the considered interpretations $\varphi \rightsquigarrow$ **theory** (language + axioms)

logical equivalence:

- formulae φ and ψ are logically equivalent if $\models \varphi \leftrightarrow \psi$
- (or: if for any interpretation I of the given language it holds that $I \models \varphi$ iff $I \models \psi$)
- denoted as $\varphi \Leftrightarrow \psi$

First-Order Logic

logical equivalence:

- formulae φ and ψ are logically equivalent if $\models \varphi \leftrightarrow \psi$
- (or: if for any interpretation I of the given language it holds that $I \models \varphi$ iff $I \models \psi$)
- denoted as $\varphi \Leftrightarrow \psi$

logical consequence:

- formula ψ is a logical consequence of a formula φ if $\models \varphi \rightarrow \psi$
- (or: if for any interpretation I of the given language it holds that: if $I \models \varphi$, then $I \models \psi$)
- denoted as $\varphi \Rightarrow \psi$

Semantic Argument for FOL

To decide validity of FOL formulae, we extend the semantic argument method from PL using the following proof rules:

Semantic Argument for FOL

To decide validity of FOL formulae, we extend the semantic argument method from PL using the following proof rules:

- universal quantification 1: $$\frac{I \models \forall x\, \varphi}{I \triangleleft \{x \mapsto t\} \models \varphi}$$ for *any ground term* t

- existential quantification 1: $$\frac{I \not\models \exists x\, \varphi}{I \triangleleft \{x \mapsto t\} \not\models \varphi}$$ for *any ground term* t

  In practice, we often choose t containing symbols that were introduced earlier (to obtain a contradiction). We assume the language has at least one constant symbol.

Semantic Argument for FOL

To decide validity of FOL formulae, we extend the semantic argument method from PL using the following proof rules:

- universal quantification 1: $$\frac{I \models \forall x\, \varphi}{I \triangleleft \{x \mapsto t\} \models \varphi}$$ for *any ground term* t

- existential quantification 1: $$\frac{I \not\models \exists x\, \varphi}{I \triangleleft \{x \mapsto t\} \not\models \varphi}$$ for *any ground term* t

  In practice, we often choose t containing symbols that were introduced earlier (to obtain a contradiction). We assume the language has at least one constant symbol.

- universal quantification 2: $$\frac{I \not\models \forall x\, \varphi}{I \triangleleft \{x \mapsto c\} \not\models \varphi}$$ for a *fresh constant symbol* c

- existential quantification 2: $$\frac{I \models \exists x\, \varphi}{I \triangleleft \{x \mapsto c\} \models \varphi}$$ for a *fresh constant symbol* c

  The value c cannot have been used in the proof before.

Semantic Argument for FOL

- contradiction:

$$\frac{\begin{array}{c} J : I \lhd \cdots \models p(s_1, \ldots, s_n) \\ K : I \lhd \cdots \not\models p(t_1, \ldots, t_n) \end{array}}{I \models \bot}$$

for $1 \leq i \leq n$: $J[s_i] = K[t_i]$

Semantic Argument for FOL

- contradiction:

$$\frac{\begin{array}{c} J : I \triangleleft \cdots \models p(s_1, \ldots, s_n) \\ K : I \triangleleft \cdots \not\models p(t_1, \ldots, t_n) \end{array}}{I \models \bot} \qquad \text{for } 1 \leq i \leq n \colon J[s_i] = K[t_i]$$

- rules for $(=)$ will be introduced in the next lecture (about theories)

Semantic Argument for FOL (example)

Example

Prove that the formula $\psi\colon (\forall x(p(x))) \rightarrow (\forall y(p(y)))$ is valid.

Semantic Argument for FOL (example)

Example

Prove that the formula $\psi : (\forall x(p(x))) \rightarrow (\forall y(p(y)))$ is valid.

Solution.
Assume ψ is invalid, i.e., there exists I s.t. $I \not\models \psi$. Then,

1.	$I \not\models (\forall x(p(x))) \rightarrow (\forall y(p(y)))$	assumption
2.	$I \models \forall x(p(x))$	by 1 and semantics of $\rightarrow$
3.	$I \models \forall y(p(y))$	by 1 and semantics of $\rightarrow$
4.	$I \triangleleft \{y \mapsto v_1\} \not\models p(y)$	by 3 and semantics of $\forall$
5.	$I \triangleleft \{x \mapsto v_1\} \models p(x)$	by 2 and semantics of $\forall$
6.	$I \models \bot$	from 4 and 5

□

Substitution

Substitution

- again, more involved than for PL (because of quantifiers)

Substitution

Substitution

- again, more involved than for PL (because of quantifiers)
- **Renaming**: Let $\varphi = \forall x\, \psi$. The renaming of x to a fresh variable x' in φ is the formula $\varphi[x/x'] = \forall x'\, \psi'$ where ψ' is obtained ψ by replacing every free occurrence of x by x'.

Substitution

Substitution

- again, more involved than for PL (because of quantifiers)
- **Renaming**: Let $\varphi = \forall x\, \psi$. The renaming of x to a fresh variable x' in φ is the formula $\varphi[x/x'] = \forall x'\, \psi'$ where ψ' is obtained ψ by replacing every free occurrence of x by x'.
- **Substitution**: mapping from formulae to formulae

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

Substitution

Substitution

- again, more involved than for PL (because of quantifiers)
- **Renaming**: Let $\varphi = \forall x\,\psi$. The renaming of x to a fresh variable x' in φ is the formula $\varphi[x/x'] = \forall x'\,\psi'$ where ψ' is obtained ψ by replacing every free occurrence of x by x'.
- **Substitution**: mapping from formulae to formulae

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

- **Safe substitution**: $F\sigma$
  - for each quantified variable x in F that also occurs free in σ, rename x to a fresh variable x' to produce F'
    - (e.g., $\exists x(x = y)$)
  - compute $F'\sigma$

Substitution

Substitution

- again, more involved than for PL (because of quantifiers)
- **Renaming**: Let $\varphi = \forall x\, \psi$. The renaming of x to a fresh variable x' in φ is the formula $\varphi[x/x'] = \forall x'\, \psi'$ where ψ' is obtained ψ by replacing every free occurrence of x by x'.
- **Substitution**: mapping from formulae to formulae

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

- **Safe substitution**: $F\sigma$
  - for each quantified variable x in F that also occurs free in σ, rename x to a fresh variable x' to produce F'
    - (e.g., $\exists x(x = y)$)
  - compute $F'\sigma$

Proposition (Substitution of Equivalent Formulae)

If, given σ, for each i it holds that $F_i \Leftrightarrow G_i$, then $F \Leftrightarrow F\sigma$ where $F\sigma$ is computed as a safe substitution.

Useful Equivalences

$$
\begin{array}{rcll}
\forall x(\neg\varphi) & \Leftrightarrow & \neg\exists x\,\varphi & \\
\exists x(\neg\varphi) & \Leftrightarrow & \neg\forall x\,\varphi & \\
(\forall x\,\varphi(x)) \wedge (\forall y\,\psi(y)) & \Leftrightarrow & \forall x(\varphi(x) \wedge \psi(x)) & \text{if } x \notin \mathtt{FREE}(\psi) \\
(\exists x\,\varphi(x)) \vee (\exists y\,\psi(y)) & \Leftrightarrow & \exists x(\varphi(x) \vee \psi(x)) & \text{if } x \notin \mathtt{FREE}(\psi) \\
\forall x\,\varphi & \Leftrightarrow & \varphi & \text{if } x \notin \mathtt{FREE}(\varphi) \\
\exists x\,\varphi & \Leftrightarrow & \varphi & \text{if } x \notin \mathtt{FREE}(\varphi) \\
\forall x(\varphi \vee \psi) & \Leftrightarrow & (\forall x\,\varphi) \vee \psi & \text{if } x \notin \mathtt{FREE}(\psi) \\
\exists x(\varphi \wedge \psi) & \Leftrightarrow & (\exists x\,\varphi) \wedge \psi & \text{if } x \notin \mathtt{FREE}(\psi)
\end{array}
$$

Normal Forms (NNF)

Negation Normal Form (NNF):

- similar as for PL
- contains only $\wedge$, $\vee$, $\neg$, $\underline{\exists\text{, and } \forall}$ as connectives
- $\neg$ appears only in front of predicates

Normal Forms (NNF)

Negation Normal Form (NNF):

- similar as for PL
- contains only $\wedge$, $\vee$, $\neg$, $\exists$, and $\forall$ as connectives
- $\neg$ appears only in front of predicates

Example

Let

$$\varphi\colon \neg\exists n\exists x\exists y\big(n > 2 \quad\wedge\quad \exists z(x^n + y^n = z^n)\big).$$

Normal Forms (NNF)

Negation Normal Form (NNF):

- similar as for PL
- contains only $\wedge$, $\vee$, $\neg$, $\exists$, and $\forall$ as connectives
- $\neg$ appears only in front of predicates

Example

Let

$$\varphi\colon \neg\exists n \exists x \exists y \big(n > 2 \quad \wedge \quad \exists z (x^n + y^n = z^n)\big).$$

The formula

$$\psi\colon \forall n \forall x \forall y \big(\neg(n > 2) \quad \vee \quad \forall z (\neg(x^n + y^n = z^n))\big)$$

is equivalent to φ and is in NNF.

Normal Forms (PNF)

Prenex Normal Form (PNF):

- formula is of the form

$$\varphi = \underbrace{Q_1 x_1 \ . \ \ldots Q_n x_n}_{\text{prefix}} \big(\underbrace{\psi(x_1, \ldots, x_n, y_1, \ldots, y_m)}_{\text{matrix}}\big)$$

  where $Q_i \in \{\forall, \exists\}$ and ψ is quantifier-free; $\{y_1, \ldots, y_m\}$ are the free variables of φ

Normal Forms (PNF)

Prenex Normal Form (PNF):

- formula is of the form

$$\varphi = \underbrace{Q_1 x_1 \ldots Q_n x_n}_{\text{prefix}} \underbrace{(\psi(x_1, \ldots, x_n, y_1, \ldots, y_m))}_{\text{matrix}}$$

where $Q_i \in \{\forall, \exists\}$ and ψ is quantifier-free; $\{y_1, \ldots, y_m\}$ are the free variables of φ

Example

Let

$$\psi : \forall n \forall x \forall y \big(\neg(n > 2) \quad \vee \quad \forall z(\neg(x^n + y^n = z^n))\big).$$

Normal Forms (PNF)

Prenex Normal Form (PNF):

- formula is of the form

$$\varphi = \underbrace{Q_1 x_1 \ldots Q_n x_n}_{\text{prefix}} \underbrace{\big(\psi(x_1, \ldots, x_n, y_1, \ldots, y_m)\big)}_{\text{matrix}}$$

where $Q_i \in \{\forall, \exists\}$ and ψ is quantifier-free; $\{y_1, \ldots, y_m\}$ are the free variables of φ

Example

Let

$$\psi : \forall n \forall x \forall y \big(\neg(n > 2) \quad \vee \quad \forall z(\neg(x^n + y^n = z^n))\big).$$

The formula

$$\chi : \forall n \forall x \forall y \forall z \big(\neg(n > 2) \quad \vee \quad \neg(x^n + y^n = z^n)\big)$$

is equivalent to ψ and is in PNF.

Prenex Normal Form

Conversion to PNF:

1. elimination of useless quantifiers: $Qx\varphi \rightsquigarrow \varphi$ for $Q \in \{\exists, \forall\}$ if $x \notin \mathtt{FREE}(\varphi)$

Prenex Normal Form

Conversion to PNF:

1. elimination of useless quantifiers: $Qx\varphi \rightsquigarrow \varphi$ for $Q \in \{\exists, \forall\}$ if $x \notin \mathtt{FREE}(\varphi)$
2. eliminination of occurrences of $\leftrightarrow$: $\varphi \leftrightarrow \psi \rightsquigarrow (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$

Prenex Normal Form

Conversion to PNF:

1. elimination of useless quantifiers: $Qx\varphi \rightsquigarrow \varphi$ for $Q \in \{\exists, \forall\}$ if $x \notin \mathtt{FREE}(\varphi)$
2. eliminination of occurrences of $\leftrightarrow$: $\varphi \leftrightarrow \psi \rightsquigarrow (\varphi \to \psi) \wedge (\psi \to \varphi)$
3. renaming of variables:
   - if there exists $x \in \mathbb{X}$ such that
     - it is in the intersection of $\mathtt{FREE}(\varphi)$ and $\mathtt{BOUND}(\varphi)$ or
     - it is quantified more than once

     then substitute in φ the subformula $Qx\psi$ (for $Q \in \{\exists, \forall\}$) for the formula $Qy(\psi[x/y])$
     - where y is a new variable that has no occurrence in φ

Prenex Normal Form

Conversion to PNF:

1. elimination of useless quantifiers: $Qx\varphi \rightsquigarrow \varphi$ for $Q \in \{\exists, \forall\}$ if $x \notin \mathtt{FREE}(\varphi)$
2. eliminination of occurrences of $\leftrightarrow$: $\varphi \leftrightarrow \psi \rightsquigarrow (\varphi \to \psi) \land (\psi \to \varphi)$
3. renaming of variables:
   - if there exists $x \in \mathbb{X}$ such that
     - it is in the intersection of $\mathtt{FREE}(\varphi)$ and $\mathtt{BOUND}(\varphi)$ or
     - it is quantified more than once

     then substitute in φ the subformula $Qx\psi$ (for $Q \in \{\exists, \forall\}$) for the formula $Qy(\psi[x/y])$
     - where y is a new variable that has no occurrence in φ
4. push negation inside:

$$
\begin{array}{lcl|lcl|lcl}
\neg\exists x\varphi & \rightsquigarrow & \forall x(\neg\varphi) & \neg(\varphi\land\psi) & \rightsquigarrow & \neg\varphi\lor\neg\psi & \neg(\varphi\to\psi) & \rightsquigarrow & \varphi\land\neg\psi \\
\neg\forall x\varphi & \rightsquigarrow & \exists x(\neg\varphi) & \neg(\varphi\lor\psi) & \rightsquigarrow & \neg\varphi\land\neg\psi & \neg\neg\varphi & \rightsquigarrow & \varphi
\end{array}
$$

Prenex Normal Form

Conversion to PNF:

1. elimination of useless quantifiers: $Qx\varphi \rightsquigarrow \varphi$ for $Q \in \{\exists, \forall\}$ if $x \notin \mathtt{FREE}(\varphi)$
2. eliminination of occurrences of $\leftrightarrow$: $\varphi \leftrightarrow \psi \rightsquigarrow (\varphi \to \psi) \wedge (\psi \to \varphi)$
3. renaming of variables:
   - if there exists $x \in \mathbb{X}$ such that
     - it is in the intersection of $\mathtt{FREE}(\varphi)$ and $\mathtt{BOUND}(\varphi)$ or
     - it is quantified more than once

   then substitute in φ the subformula $Qx\psi$ (for $Q \in \{\exists, \forall\}$) for the formula $Qy(\psi[x/y])$
     - where y is a new variable that has no occurrence in φ
4. push negation inside:

$$\begin{array}{lcl|lcl|lcl} \neg\exists x\varphi & \rightsquigarrow & \forall x(\neg\varphi) & \neg(\varphi \wedge \psi) & \rightsquigarrow & \neg\varphi \vee \neg\psi & \neg(\varphi \to \psi) & \rightsquigarrow & \varphi \wedge \neg\psi \\ \neg\forall x\varphi & \rightsquigarrow & \exists x(\neg\varphi) & \neg(\varphi \vee \psi) & \rightsquigarrow & \neg\varphi \wedge \neg\psi & \neg\neg\varphi & \rightsquigarrow & \varphi \end{array}$$

5. move quantifiers to the left:

$$\begin{array}{lcl@{\qquad}lcl} Qx(\varphi) \wedge \psi & \rightsquigarrow & Qx(\varphi \wedge \psi) & Qx(\varphi) \to \psi & \rightsquigarrow & \overline{Q}x(\varphi \to \psi) \\ Qx(\varphi) \vee \psi & \rightsquigarrow & Qx(\varphi \vee \psi) & \varphi \to Qx(\psi) & \rightsquigarrow & Qx(\varphi \to \psi) \end{array}$$

for $Q \in \{\exists, \forall\}$, where $\overline{Q}$ is the quantifier “opposite” to Q ($\overline{\exists} \mapsto \forall$ a $\overline{\forall} \mapsto \exists$).

Prenex Normal Form

Example

Convert the following formula to PNF.

$$\forall n \big(n > 2 \quad \rightarrow \quad \neg\exists x \exists y \exists z (x^n + y^n = z^n)\big) \Leftrightarrow$$

Prenex Normal Form

Example

Convert the following formula to PNF.

$$\forall n \big(n > 2 \quad \to \quad \neg \exists x \exists y \exists z (x^n + y^n = z^n) \big) \Leftrightarrow$$

$$\Leftrightarrow \forall n \big(n > 2 \quad \to \quad \forall x \neg \exists y \exists z (x^n + y^n = z^n) \big)$$

Prenex Normal Form

Example

Convert the following formula to PNF.

$$\forall n\big(n > 2 \quad\to\quad \neg\exists x\exists y\exists z(x^n + y^n = z^n)\big) \Leftrightarrow$$

$$\Leftrightarrow \forall n\big(n > 2 \quad\to\quad \forall x\neg\exists y\exists z(x^n + y^n = z^n)\big)$$

$$\Leftrightarrow \forall n\big(n > 2 \quad\to\quad \forall x\forall y\neg\exists z(x^n + y^n = z^n)\big)$$

Prenex Normal Form

Example

Convert the following formula to PNF.

$$\forall n \big(n > 2 \quad \rightarrow \quad \neg \exists x \exists y \exists z (x^n + y^n = z^n)\big) \Leftrightarrow$$

$$\Leftrightarrow \forall n \big(n > 2 \quad \rightarrow \quad \forall x \neg \exists y \exists z (x^n + y^n = z^n)\big)$$

$$\Leftrightarrow \forall n \big(n > 2 \quad \rightarrow \quad \forall x \forall y \neg \exists z (x^n + y^n = z^n)\big)$$

$$\Leftrightarrow \forall n \big(n > 2 \quad \rightarrow \quad \forall x \forall y \forall z (\neg (x^n + y^n = z^n))\big)$$

Prenex Normal Form

Example

Convert the following formula to PNF.

$$\forall n\big(n > 2 \quad \to \quad \neg\exists x\exists y\exists z(x^n + y^n = z^n)\big) \Leftrightarrow$$

$$\Leftrightarrow \forall n\big(n > 2 \quad \to \quad \forall x\neg\exists y\exists z(x^n + y^n = z^n)\big)$$

$$\Leftrightarrow \forall n\big(n > 2 \quad \to \quad \forall x\forall y\neg\exists z(x^n + y^n = z^n)\big)$$

$$\Leftrightarrow \forall n\big(n > 2 \quad \to \quad \forall x\forall y\forall z(\neg(x^n + y^n = z^n))\big)$$

$$\Leftrightarrow \forall n\forall x\big(n > 2 \quad \to \quad \forall y\forall z(\neg(x^n + y^n = z^n))\big)$$

Prenex Normal Form

Example

Convert the following formula to PNF.

$$\forall n\big(n > 2 \quad\to\quad \neg\exists x\exists y\exists z(x^n + y^n = z^n)\big) \Leftrightarrow$$

$$\Leftrightarrow \forall n\big(n > 2 \quad\to\quad \forall x\neg\exists y\exists z(x^n + y^n = z^n)\big)$$

$$\Leftrightarrow \forall n\big(n > 2 \quad\to\quad \forall x\forall y\neg\exists z(x^n + y^n = z^n)\big)$$

$$\Leftrightarrow \forall n\big(n > 2 \quad\to\quad \forall x\forall y\forall z(\neg(x^n + y^n = z^n))\big)$$

$$\Leftrightarrow \forall n\forall x\big(n > 2 \quad\to\quad \forall y\forall z(\neg(x^n + y^n = z^n))\big)$$

$$\Leftrightarrow \forall n\forall x\forall y\big(n > 2 \quad\to\quad \forall z(\neg(x^n + y^n = z^n))\big)$$

Prenex Normal Form

Example

Convert the following formula to PNF.

$$\forall n\big(n > 2 \quad \rightarrow \quad \neg\exists x\exists y\exists z(x^n + y^n = z^n)\big) \Leftrightarrow$$

$$\Leftrightarrow \forall n\big(n > 2 \quad \rightarrow \quad \forall x\neg\exists y\exists z(x^n + y^n = z^n)\big)$$

$$\Leftrightarrow \forall n\big(n > 2 \quad \rightarrow \quad \forall x\forall y\neg\exists z(x^n + y^n = z^n)\big)$$

$$\Leftrightarrow \forall n\big(n > 2 \quad \rightarrow \quad \forall x\forall y\forall z(\neg(x^n + y^n = z^n))\big)$$

$$\Leftrightarrow \forall n\forall x\big(n > 2 \quad \rightarrow \quad \forall y\forall z(\neg(x^n + y^n = z^n))\big)$$

$$\Leftrightarrow \forall n\forall x\forall y\big(n > 2 \quad \rightarrow \quad \forall z(\neg(x^n + y^n = z^n))\big)$$

$$\Leftrightarrow \forall n\forall x\forall y\forall z\big(n > 2 \quad \rightarrow \quad \neg(x^n + y^n = z^n)\big)$$

Prenex Normal Form

Example

Convert the following formula to PNF.

$$\forall y \Big(\exists x (P(x,y)) \to Q(y,z) \Big) \wedge \neg \exists z \Big(\forall x \big(R(x,y) \vee Q(x,y) \big) \Big) \Leftrightarrow$$

Prenex Normal Form

Example

Convert the following formula to PNF.

$$\forall y \Big(\exists x (P(x,y)) \to Q(y,z) \Big) \land \neg \exists z \Big(\forall x \big(R(x,y) \lor Q(x,y) \big) \Big) \Leftrightarrow$$

$$\Leftrightarrow \forall y \Big(\exists x (P(x,y)) \to Q(y,z) \Big) \land \neg \forall x \big(R(x,y) \lor Q(x,y) \big)$$

Prenex Normal Form

Example

Convert the following formula to PNF.

$$\forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \land \neg\exists z\Big(\forall x\big(R(x,y) \lor Q(x,y)\big)\Big) \Leftrightarrow$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \land \neg\forall x\big(R(x,y) \lor Q(x,y)\big)$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \land \neg\forall u\big(R(u,y) \lor Q(u,y)\big)$$

Prenex Normal Form

Example

Convert the following formula to PNF.

$$\forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \wedge \neg\exists z\Big(\forall x\big(R(x,y) \vee Q(x,y)\big)\Big) \Leftrightarrow$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \wedge \neg\forall x\big(R(x,y) \vee Q(x,y)\big)$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \wedge \neg\forall u\big(R(u,y) \vee Q(u,y)\big)$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \wedge \exists u\Big(\neg\big(R(u,y) \vee Q(u,y)\big)\Big)$$

Prenex Normal Form

Example

Convert the following formula to PNF.

$$\forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \wedge \neg\exists z\Big(\forall x\big(R(x,y) \vee Q(x,y)\big)\Big) \Leftrightarrow$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \wedge \neg\forall x\big(R(x,y) \vee Q(x,y)\big)$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \wedge \neg\forall u\big(R(u,y) \vee Q(u,y)\big)$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \wedge \exists u\Big(\neg\big(R(u,y) \vee Q(u,y)\big)\Big)$$

$$\Leftrightarrow \forall y\Big(\big(\exists x(P(x,y)) \to Q(y,z)\big) \wedge \exists u\big(\neg(R(u,y) \vee Q(u,y))\big)\Big)$$

Prenex Normal Form

Example

Convert the following formula to PNF.

$$\forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \land \neg\exists z\Big(\forall x\big(R(x,y) \lor Q(x,y)\big)\Big) \Leftrightarrow$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \land \neg\forall x\big(R(x,y) \lor Q(x,y)\big)$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \land \neg\forall u\big(R(u,y) \lor Q(u,y)\big)$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \land \exists u\Big(\neg\big(R(u,y) \lor Q(u,y)\big)\Big)$$

$$\Leftrightarrow \forall y\Big(\big(\exists x(P(x,y)) \to Q(y,z)\big) \land \exists u\big(\neg(R(u,y) \lor Q(u,y))\big)\Big)$$

$$\Leftrightarrow \forall y \exists u\Big(\big(\exists x(P(x,y)) \to Q(y,z)\big) \land \neg\big(R(u,y) \lor Q(u,y)\big)\Big)$$

Prenex Normal Form

Example

Convert the following formula to PNF.

$$\forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \land \neg\exists z\Big(\forall x\big(R(x,y) \lor Q(x,y)\big)\Big) \Leftrightarrow$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \land \neg\forall x\big(R(x,y) \lor Q(x,y)\big)$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \land \neg\forall u\big(R(u,y) \lor Q(u,y)\big)$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \land \exists u\Big(\neg\big(R(u,y) \lor Q(u,y)\big)\Big)$$

$$\Leftrightarrow \forall y\Big(\big(\exists x(P(x,y)) \to Q(y,z)\big) \land \exists u\big(\neg(R(u,y) \lor Q(u,y))\big)\Big)$$

$$\Leftrightarrow \forall y\exists u\Big(\big(\exists x(P(x,y)) \to Q(y,z)\big) \land \neg\big(R(u,y) \lor Q(u,y)\big)\Big)$$

$$\Leftrightarrow \forall y\exists u\Big(\forall x\big(P(x,y) \to Q(y,z)\big) \land \neg\big(R(u,y) \lor Q(u,y)\big)\Big)$$

Prenex Normal Form

Example

Convert the following formula to PNF.

$$\forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \land \neg\exists z\Big(\forall x\big(R(x,y) \lor Q(x,y)\big)\Big) \Leftrightarrow$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \land \neg\forall x\big(R(x,y) \lor Q(x,y)\big)$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \land \neg\forall u\big(R(u,y) \lor Q(u,y)\big)$$

$$\Leftrightarrow \forall y\Big(\exists x(P(x,y)) \to Q(y,z)\Big) \land \exists u\Big(\neg\big(R(u,y) \lor Q(u,y)\big)\Big)$$

$$\Leftrightarrow \forall y\Big((\exists x(P(x,y)) \to Q(y,z)) \land \exists u(\neg(R(u,y) \lor Q(u,y)))\Big)$$

$$\Leftrightarrow \forall y\exists u\Big((\exists x(P(x,y)) \to Q(y,z)) \land \neg\big(R(u,y) \lor Q(u,y)\big)\Big)$$

$$\Leftrightarrow \forall y\exists u\Big(\forall x\big(P(x,y) \to Q(y,z)\big) \land \neg\big(R(u,y) \lor Q(u,y)\big)\Big)$$

$$\Leftrightarrow \forall y\exists u\forall x\Big(\big(P(x,y) \to Q(y,z)\big) \land \neg\big(R(u,y) \lor Q(u,y)\big)\Big)$$

Normal Forms (DNF, CNF)

- disjunctive normal form (DNF): PNF where matrix is in DNF
- conjuctive normal form (CNF): PNF where matrix is in CNF

Skolem Normal Form

- Skolem Normal Form (SNF):
  - formula is in the PNF
  - formula does not contain any existential quantifier $\exists$
- Given a FOL formula φ, there might not be an equivalent formula in the SNF.
- There will, however, always be an equisatisfiable formula φ' in the SNF.
  - equisatisfiable: φ is satisfiable iff φ' is satisfiable
- Skolemization:
  - Assume the following formula:

$$\varphi\colon \forall x_1 \forall x_2 \dots \forall x_k \exists y (\psi)$$

Skolem Normal Form

- Skolem Normal Form (SNF):
  - formula is in the PNF
  - formula does not contain any existential quantifier $\exists$
- Given a FOL formula φ, there might not be an equivalent formula in the SNF.
- There will, however, always be an equisatisfiable formula φ' in the SNF.
  - equisatisfiable: φ is satisfiable iff φ' is satisfiable
- Skolemization:
  - Assume the following formula:

$$\varphi\colon \forall x_1 \forall x_2 \ldots \forall x_k \exists y(\psi)$$

  - y depends on $x_1, \ldots, x_k$
  - φ is satisfiable iff for every tuple $(x_1, \ldots, x_k)$, there exists a y such that ψ is satisifable under such an incomplete assignment
  - i.e., if there exists a k-ary function f_y that for every tuple $(x_1, \ldots, x_k)$ assigns a corresponding y
  - $\rightsquigarrow$ we can remove $\exists y$ and substitute all free occurrences of y in ψ for $f_y(x_1, \ldots, x_k)$

Skolem Normal Form

Example

Transform the following formula into an equisatisfiable formula in the SNF:

$$\exists x \forall y \exists z \forall u \exists v (x + y + z = u + v)$$

Skolem Normal Form

Example

Transform the following formula into an equisatisfiable formula in the SNF:

$$\exists x \forall y \exists z \forall u \exists v (x + y + z = u + v)$$

$$\rightsquigarrow \forall y \exists z \forall u \exists v (f_x + y + z = u + v)$$

$$\rightsquigarrow \forall y \forall u \exists v (f_x + y + f_z(y) = u + v)$$

$$\rightsquigarrow \forall y \forall u (f_x + y + f_z(y) = u + f_v(y, u))$$

Soundness and Completeness of Semantic Argument

Soundness

- a proof method is **sound** if it never proves a wrong formula:

$$\text{if} \quad \vdash \varphi \qquad \text{then} \qquad \models \varphi$$

$\vdash \varphi$: φ is provable

Theorem

The semantic argument is sound.

Soundness and Completeness of Semantic Argument

Soundness

- a proof method is **sound** if it never proves a wrong formula:

$$\text{if} \quad \vdash \varphi \qquad \text{then} \qquad \models \varphi$$

$\vdash \varphi$: φ is provable

Theorem

The semantic argument is sound.

Completeness

- a proof method is **complete** if it can prove every valid formula:

$$\text{if} \quad \models \varphi \qquad \text{then} \qquad \vdash \varphi$$

Theorem (Gödel's completeness theorem)

The semantic argument is complete.

Soundness and Completeness of Semantic Argument

Soundness

- a proof method is **sound** if it never proves a wrong formula:

$$\text{if} \quad \vdash \varphi \qquad \text{then} \qquad \models \varphi$$

$\vdash \varphi$: φ is provable

Theorem

The semantic argument is sound.

Completeness

- a proof method is **complete** if it can prove every valid formula:

$$\text{if} \quad \models \varphi \qquad \text{then} \qquad \vdash \varphi$$

Theorem (Gödel's completeness theorem)

The semantic argument is complete.

There are also other sound and complete methods for FOL (e.g. natural deduction, Hilbert system, resolution).

Löwenheim-Skolem Theorem

Theorem (Löwenheim-Skolem (simplified))

If an FOL formula has a model of an infinite cardinality then it has a model of ***any*** *infinite cardinality.*

Herbrand Interpretation

Herbrand interpretation of a language L

- a special kind of interpretation I_H
- the domain D_H is fixed as the set of all ground terms of L (i.e., no variables),
  - if L does not contain any constant symbol, we create a new one
- interpretation of function symbols is "*natural*"

Herbrand Interpretation

Herbrand interpretation of a language L

- a special kind of interpretation I_H
- the domain D_H is fixed as the set of all ground terms of L (i.e., no variables),
  - if L does not contain any constant symbol, we create a new one
- interpretation of function symbols is "*natural*"

Example

Let us assume the FOL language L with the signature $\langle \mathcal{F} = \{a_{/0}, f_{/2}, g_{/1}\}, \mathcal{P} = \{p_{/2}\}\rangle$

Herbrand Interpretation

Herbrand interpretation of a language L

- a special kind of interpretation I_H
- the domain D_H is fixed as the set of all ground terms of L (i.e., no variables),
  - if L does not contain any constant symbol, we create a new one
- interpretation of function symbols is "*natural*"

Example

Let us assume the FOL language L with the signature $\langle \mathcal{F} = \{a_{/0}, f_{/2}, g_{/1}\}, \mathcal{P} = \{p_{/2}\}\rangle$

- $D_H = \{\text{“}a\text{”}, \text{“}g(a)\text{”}, \text{“}f(a,a)\text{”}, \text{“}g(g(a))\text{”}, \text{“}g(f(a,a))\text{”}, \text{“}f(g(a),a)\text{”}, \text{“}g(f(a,g(g(a))))\text{”}, \ldots\}$

Herbrand Interpretation

Herbrand interpretation of a language L

- a special kind of interpretation I_H
- the domain D_H is fixed as the set of all ground terms of L (i.e., no variables),
  - if L does not contain any constant symbol, we create a new one
- interpretation of function symbols is "*natural*"

Example

Let us assume the FOL language L with the signature $\langle \mathcal{F} = \{a_{/0}, f_{/2}, g_{/1}\}, \mathcal{P} = \{p_{/2}\}\rangle$

- $D_H = \{$"a", "$g(a)$", "$f(a,a)$", "$g(g(a))$", "$g(f(a,a))$", "$f(g(a),a)$", "$g(f(a,g(g(a))))$", $\ldots\}$
- $I_H(a) = \{() \mapsto$ "a"$\}$

Herbrand Interpretation

Herbrand interpretation of a language L

- a special kind of interpretation I_H
- the domain D_H is fixed as the set of all ground terms of L (i.e., no variables),
  - if L does not contain any constant symbol, we create a new one
- interpretation of function symbols is "*natural*"

Example

Let us assume the FOL language L with the signature $\langle \mathcal{F} = \{a_{/0}, f_{/2}, g_{/1}\}, \mathcal{P} = \{p_{/2}\} \rangle$

- $D_H = \{\text{``}a\text{''}, \text{``}g(a)\text{''}, \text{``}f(a,a)\text{''}, \text{``}g(g(a))\text{''}, \text{``}g(f(a,a))\text{''}, \text{``}f(g(a),a)\text{''}, \text{``}g(f(a,g(g(a))))\text{''}, \ldots\}$
- $I_H(a) = \{() \mapsto \text{``}a\text{''}\}$
- $I_H(g) = \{\text{``}a\text{''} \mapsto \text{``}g(a)\text{''}, \text{``}g(a)\text{''} \mapsto \text{``}g(g(a))\text{''}, \text{``}f(a,a)\text{''} \mapsto \text{``}g(f(a,a))\text{''}, \ldots\}$

Herbrand Interpretation

Herbrand interpretation of a language L

- a special kind of interpretation I_H
- the domain D_H is fixed as the set of all ground terms of L (i.e., no variables),
  - if L does not contain any constant symbol, we create a new one
- interpretation of function symbols is "*natural*"

Example

Let us assume the FOL language L with the signature $\langle \mathcal{F} = \{a_{/0}, f_{/2}, g_{/1}\}, \mathcal{P} = \{p_{/2}\}\rangle$

- $D_H = \{\text{“}a\text{”}, \text{“}g(a)\text{”}, \text{“}f(a,a)\text{”}, \text{“}g(g(a))\text{”}, \text{“}g(f(a,a))\text{”}, \text{“}f(g(a),a)\text{”}, \text{“}g(f(a,g(g(a))))\text{”}, \dots\}$
- $I_H(a) = \{() \mapsto \text{“}a\text{”}\}$
- $I_H(g) = \{\text{“}a\text{”} \mapsto \text{“}g(a)\text{”}, \text{“}g(a)\text{”} \mapsto \text{“}g(g(a))\text{”}, \text{“}f(a,a)\text{”} \mapsto \text{“}g(f(a,a))\text{”}, \dots\}$
- $I_H(f) = \{(\text{“}a\text{”}, \text{“}a\text{”}) \mapsto \text{“}f(a,a)\text{”}, (\text{“}g(a)\text{”}, \text{“}a\text{”}) \mapsto \text{“}f(g(a),a)\text{”}, \dots\}$

Herbrand model

Herbrand model: a model of a formula that is also a Herbrand interpretation

- i.e., we need to provide interpretation of predicate symbols and variables only

Theorem (Herbrand's theorem (simplified))

A set of FOL formulae has a model iff it has a Herbrand model.

- $\rightsquigarrow$ it is enough to search for Herbrand models (e.g., model construct. in SMT solvers)
- minimal Herbrand model (semantics of PROLOG programs)

Notes

- Exists exactly one:

$$\exists! x\, \varphi(x) \quad \overset{\text{def}}{\Leftrightarrow} \quad \exists x(\varphi(x) \wedge \forall y(\varphi(y) \;\rightarrow\; x = y))$$

  where y is not free in φ

- many-sorted logics:
  - capture the natural requirement to distinguish **types** of variables
  - e.g. in

$$\forall w \in \Sigma^* \big(safe(w) \rightarrow \exists n \in \mathbb{N}(\#_{'('}(w) = \#_{')'}(w))\big)$$

References

[A.R. Bradley and Z. Manna. The Calculus of Computation.]