Proof theory of first order logic

Ondřej Ježil

November 10, 2022

First order logic – recap

- Extends propositional logic.

First order logic – recap

- Extends propositional logic.
- Allowed symbols:

First order logic – recap

- Extends propositional logic.
- Allowed symbols:
  - propositional connectives ($\wedge$, $\vee$, $\neg$, $\supset$)

First order logic – recap

- Extends propositional logic.
- Allowed symbols:
  - propositional connectives ($\wedge$, $\vee$, $\neg$, $\supset$)
  - quantifiers ($(\forall x)$, $(\exists x)$, $\dots$)

First order logic – recap

- Extends propositional logic.
- Allowed symbols:
  - propositional connectives ($\wedge$, $\vee$, $\neg$, $\supset$)
  - quantifiers (($\forall x$), ($\exists x$), ...)
  - variables ($x, y, z \ldots$, a, b, , ...)

First order logic – recap

- Extends propositional logic.
- Allowed symbols:
  - propositional connectives ($\wedge$, $\vee$, $\neg$, $\supset$)
  - quantifiers (($\forall x$), ($\exists x$), ...)
  - variables ($x, y, z \ldots,\ a, b, , \ldots$)
  - function symbols ($f, g, h, \ldots$)

First order logic – recap

- Extends propositional logic.
- Allowed symbols:
  - propositional connectives ($\wedge$, $\vee$, $\neg$, $\supset$)
  - quantifiers ($(\forall x)$, $(\exists x)$, ...)
  - variables ($x, y, z \ldots, a, b, , \ldots$)
  - function symbols ($f, g, h, \ldots$)
  - constant symbols (function symbols of arity 0)

First order logic – recap

- Extends propositional logic.
- Allowed symbols:
  - propositional connectives ($\wedge$, $\vee$, $\neg$, $\supset$)
  - quantifiers ($(\forall x)$, $(\exists x)$, $\dots$)
  - variables ($x, y, z \dots$, a, b, $, \dots$)
  - function symbols ($f, g, h, \dots$)
  - constant symbols (function symbols of arity 0)
  - relational symbols (P, Q, R, $\dots$)

First order logic – recap cont.

Definition (L-formulas)

*Let L be a set of function and relational symbols (a language), we inductively define L-**terms**. Every variable is a term and if $f \in L$ is k-ary, $t_1, \dots, t_k$ are terms, then $f(t_1, \dots, t_k)$ are L-terms. If $P \in L$ is k-ary and $t_1, \dots, t_k$ are terms, then the string $P(t_1, \dots, t_k)$ is an L-**atomic formula**. L-**formulas** are inductively defined starting from L-atomic formulas as follows. If A and B are L-formulas, then so are:*

First order logic – recap cont.

Definition (L-formulas)

*Let L be a set of function and relational symbols (a language), we inductively define L-**terms**. Every variable is a term and if $f \in L$ is k-ary, $t_1, \ldots, t_k$ are terms, then $f(t_1, \ldots, t_k)$ are L-terms. If $P \in L$ is k-ary and $t_1, \ldots, t_k$ are terms, then the string $P(t_1, \ldots, t_k)$ is an L-**atomic formula**. L-**formulas** are inductively defined starting from L-atomic formulas as follows. If A and B are L-formulas, then so are:*

- $(\neg A)$

First order logic – recap cont.

Definition (L-formulas)

*Let L be a set of function and relational symbols (a language), we inductively define L-**terms**. Every variable is a term and if $f \in L$ is k-ary, $t_1, \ldots, t_k$ are terms, then $f(t_1, \ldots, t_k)$ are L-terms. If $P \in L$ is k-ary and $t_1, \ldots, t_k$ are terms, then the string $P(t_1, \ldots, t_k)$ is an L-**atomic formula**. L-**formulas** are inductively defined starting from L-atomic formulas as follows. If A and B are L-formulas, then so are:*

- $(\neg A)$
- $(A \wedge B)$

First order logic – recap cont.

Definition (L-formulas)

*Let L be a set of function and relational symbols (a language), we inductively define L-**terms**. Every variable is a term and if $f \in L$ is k-ary, $t_1, \ldots, t_k$ are terms, then $f(t_1, \ldots, t_k)$ are L-terms. If $P \in L$ is k-ary and $t_1, \ldots, t_k$ are terms, then the string $P(t_1, \ldots, t_k)$ is an L-**atomic formula**. L-**formulas** are inductively defined starting from L-atomic formulas as follows. If A and B are L-formulas, then so are:*

- $(\neg A)$
- $(A \wedge B)$
- $(A \vee B)$

First order logic – recap cont.

Definition (L-formulas)

*Let L be a set of function and relational symbols (a language), we inductively define L-***terms***. Every variable is a term and if $f \in L$ is k-ary, $t_1, \ldots, t_k$ are terms, then $f(t_1, \ldots, t_k)$ are L-terms. If $P \in L$ is k-ary and $t_1, \ldots, t_k$ are terms, then the string $P(t_1, \ldots, t_k)$ is an L-***atomic formula***. L-***formulas*** *are inductively defined starting from L-atomic formulas as follows. If A and B are L-formulas, then so are:*

- $(\neg A)$
- $(A \wedge B)$
- $(A \vee B)$
- $(A \supset B)$

First order logic – recap cont.

Definition (L-formulas)

*Let L be a set of function and relational symbols (a language), we inductively define L-**terms**. Every variable is a term and if $f \in L$ is k-ary, $t_1, \ldots, t_k$ are terms, then $f(t_1, \ldots, t_k)$ are L-terms. If $P \in L$ is k-ary and $t_1, \ldots, t_k$ are terms, then the string $P(t_1, \ldots, t_k)$ is an L-**atomic formula**. L-**formulas** are inductively defined starting from L-atomic formulas as follows. If A and B are L-formulas, then so are:*

- $(\neg A)$
- $(A \wedge B)$
- $(A \vee B)$
- $(A \supset B)$
- $(\forall x)A$

First order logic – recap cont.

Definition (L-formulas)

*Let L be a set of function and relational symbols (a language), we inductively define L-***terms***. Every variable is a term and if $f \in L$ is k-ary, $t_1, \dots, t_k$ are terms, then $f(t_1, \dots, t_k)$ are L-terms. If $P \in L$ is k-ary and $t_1, \dots, t_k$ are terms, then the string $P(t_1, \dots, t_k)$ is an L-***atomic formula***. L-***formulas*** are inductively defined starting from L-atomic formulas as follows. If A and B are L-formulas, then so are:*

- $(\neg A)$
- $(A \wedge B)$
- $(A \vee B)$
- $(A \supset B)$
- $(\forall x)A$
- $(\exists x)A$

First order logic – free and bound occurances, substitution

- An occurance of a variable x in a formula A is **free** if x is not in a scope of a quantifier.

First order logic – free and bound occurances, substitution

- An occurance of a variable x in a formula A is **free** if x is not in a scope of a quantifier.
- An occurance which is not free is called **bound**.

First order logic – free and bound occurances, substitution

- An occurance of a variable x in a formula A is **free** if x is not in a scope of a quantifier.
- An occurance which is not free is called **bound**.
- Example: $((x = 0) \vee (\forall x)(x \neq f(x)))$

First order logic – free and bound occurances, substitution

- An occurance of a variable x in a formula A is **free** if x is not in a scope of a quantifier.
- An occurance which is not free is called **bound**.
- Example: $((x = 0) \vee (\forall x)(x \neq f(x)))$
- We call a formula without any free variables a **sentance**.

First order logic – free and bound occurances, substitution

- An occurance of a variable x in a formula A is **free** if x is not in a scope of a quantifier.
- An occurance which is not free is called **bound**.
- Example: $((x = 0) \vee (\forall x)(x \neq f(x)))$
- We call a formula without any free variables a **sentance**.
- If t is a term, x a variable, then $A(t/x)$ denotes the formula obtained from A by replacing every free occurance of x with t.

First order logic – free and bound occurances, substitution

- An occurance of a variable x in a formula A is **free** if x is not in a scope of a quantifier.
- An occurance which is not free is called **bound**.
- Example: $((x = 0) \vee (\forall x)(x \neq f(x)))$
- We call a formula without any free variables a **sentance**.
- If t is a term, x a variable, then $A(t/x)$ denotes the formula obtained from A by replacing every free occurance of x with t.
- We say t is **freely substitutable** for x in A if no free variable in t becomes bound as a result of this substitution.

First order logic – free and bound occurances, substitution

- An occurance of a variable x in a formula A is **free** if x is not in a scope of a quantifier.
- An occurance which is not free is called **bound**.
- Example: $((x = 0) \vee (\forall x)(x \neq f(x)))$
- We call a formula without any free variables a **sentance**.
- If t is a term, x a variable, then $A(t/x)$ denotes the formula obtained from A by replacing every free occurance of x with t.
- We say t is **freely substitutable** for x in A if no free variable in t becomes bound as a result of this substitution.
- For simplicity, if we denote a formula by $A(x)$, then $A(t)$ denotes $A(t/x)$.

First order logic - semantics

- To define a truth value for a formula, we need to give an *interpretation* of non-logical symbols appearing in it.

First order logic - semantics

- To define a truth value for a formula, we need to give an *interpretation* of non-logical symbols appearing in it.
- Let L be a language, a tuple $\mathcal{M} = \{M, f^{\mathcal{M}}, g^{\mathcal{M}}, \dots, P^{\mathcal{M}}, Q^{\mathcal{M}}, \dots\}$ is an L-structure, if M is an non-empty set, $f^{\mathcal{M}}$ is a function $f^{\mathcal{M}} : M^{a(f)} \to M$, for every function symbol in L and $R^{\mathcal{M}}$ is a relation $R^{\mathcal{M}} \subseteq M^{a(R)}$, for every relational symbol in L.

First order logic - semantics

- To define a truth value for a formula, we need to give an *interpretation* of non-logical symbols appearing in it.
- Let L be a language, a tuple $\mathcal{M} = \{M, f^{\mathcal{M}}, g^{\mathcal{M}}, \ldots, P^{\mathcal{M}}, Q^{\mathcal{M}}, \ldots\}$ is an L-structure, if M is an non-empty set, $f^{\mathcal{M}}$ is a function $f^{\mathcal{M}} : M^{a(f)} \to M$, for every function symbol in L and $R^{\mathcal{M}}$ is a relation $R^{\mathcal{M}} \subseteq M^{a(R)}$, for every relational symbol in L.
- if $= \in L$, then $=^{\mathcal{M}}$ gets interpreted as the true equality

First order logic - semantics

- To define a truth value for a formula, we need to give an *interpretation* of non-logical symbols appearing in it.
- Let L be a language, a tuple $\mathcal{M} = \{M, f^{\mathcal{M}}, g^{\mathcal{M}}, \dots, P^{\mathcal{M}}, Q^{\mathcal{M}}, \dots\}$ is an L-structure, if M is an non-empty set, $f^{\mathcal{M}}$ is a function $f^{\mathcal{M}} : M^{a(f)} \to M$, for every function symbol in L and $R^{\mathcal{M}}$ is a relation $R^{\mathcal{M}} \subseteq M^{a(R)}$, for every relational symbol in L.
- if $= \in L$, then $=^{\mathcal{M}}$ gets interpreted as the true equality
- For an L-formula φ, an L-structure $\mathcal{M}$, and an assignment $\sigma : \{\text{free variables in } \varphi\} \to \mathcal{M}$, we define

$$M \models \varphi[\sigma]$$

inductively by first interpreting terms and then logical connectives/quantifiers.

First order logic - semantics

- To define a truth value for a formula, we need to give an *interpretation* of non-logical symbols appearing in it.
- Let L be a language, a tuple $\mathcal{M} = \{M, f^{\mathcal{M}}, g^{\mathcal{M}}, \dots, P^{\mathcal{M}}, Q^{\mathcal{M}}, \dots\}$ is an L-structure, if M is an non-empty set, $f^{\mathcal{M}}$ is a function $f^{\mathcal{M}} : M^{a(f)} \to M$, for every function symbol in L and $R^{\mathcal{M}}$ is a relation $R^{\mathcal{M}} \subseteq M^{a(R)}$, for every relational symbol in L.
- if $= \in L$, then $=^{\mathcal{M}}$ gets interpreted as the true equality
- For an L-formula φ, an L-structure $\mathcal{M}$, and an assignment $\sigma : \{\text{free variables in } \varphi\} \to \mathcal{M}$, we define

$$M \models \varphi[\sigma]$$

inductively by first interpreting terms and then logical connectives/quantifiers.

First order logic - semantics

- To define a truth value for a formula, we need to give an *interpretation* of non-logical symbols appearing in it.
- Let L be a language, a tuple $\mathcal{M} = \{M, f^{\mathcal{M}}, g^{\mathcal{M}}, \dots, P^{\mathcal{M}}, Q^{\mathcal{M}}, \dots\}$ is an L-structure, if M is an non-empty set, $f^{\mathcal{M}}$ is a function $f^{\mathcal{M}} : M^{a(f)} \to M$, for every function symbol in L and $R^{\mathcal{M}}$ is a relation $R^{\mathcal{M}} \subseteq M^{a(R)}$, for every relational symbol in L.
- if $= \in L$, then $=^{\mathcal{M}}$ gets interpreted as the true equality
- For an L-formula φ, an L-structure $\mathcal{M}$, and an assignment $\sigma : \{\text{free variables in } \varphi\} \to \mathcal{M}$, we define

  $$M \models \varphi[\sigma]$$

  inductively by first interpreting terms and then logical connectives/quantifiers.

  $$\mathcal{M} \models \varphi \iff \forall \sigma : \mathcal{M} \models \varphi[\sigma]$$

First order logic - semantics cont.

- If Γ is a set of L-formulas and $\mathcal{M}$ an L-structure, then

$$M \models \Gamma \iff \forall \varphi \in \Gamma : M \models \varphi.$$

First order logic - semantics cont.

- If Γ is a set of L-formulas and $\mathcal{M}$ an L-structure, then

$$M \models \Gamma \iff \forall \varphi \in \Gamma : M \models \varphi.$$

- If $\mathcal{M} \models \Gamma$ we say Γ **is satisfied** by $\mathcal{M}$ and that $\mathcal{M}$ is a model of Γ.

First order logic - semantics cont.

- If Γ is a set of L-formulas and $\mathcal{M}$ an L-structure, then

$$M \models \Gamma \iff \forall \varphi \in \Gamma : M \models \varphi.$$

- If $\mathcal{M} \models \Gamma$ we say Γ **is satisfied** by $\mathcal{M}$ and that $\mathcal{M}$ is a model of Γ.
- Γ is said to be **satisfiable** iff it is satisfied by some model.

First order logic - semantics cont.

- If Γ is a set of L-formulas and $\mathcal{M}$ an L-structure, then

$$M \models \Gamma \iff \forall \varphi \in \Gamma : M \models \varphi.$$

- If $\mathcal{M} \models \Gamma$ we say Γ **is satisfied** by $\mathcal{M}$ and that $\mathcal{M}$ is a model of Γ.
- Γ is said to be **satisfiable** iff it is satisfied by some model.
- We say that Γ **logically implies** A, $\Gamma \models A$, if for every $\mathcal{M} \models \Gamma$ we have $\mathcal{M} \models A$.

First order logic - semantics cont.

- If Γ is a set of L-formulas and $\mathcal{M}$ an L-structure, then

$$M \models \Gamma \iff \forall \varphi \in \Gamma : M \models \varphi.$$

- If $\mathcal{M} \models \Gamma$ we say Γ **is satisfied** by $\mathcal{M}$ and that $\mathcal{M}$ is a model of Γ.
- Γ is said to be **satisfiable** iff it is satisfied by some model.
- We say that Γ **logically implies** A, $\Gamma \models A$, if for every $\mathcal{M} \models \Gamma$ we have $\mathcal{M} \models A$.
- If $\Gamma = \varnothing$ and $\Gamma \models A$, we write $\models A$ and say that A **is valid**.

First order logic - semantics cont.

- If Γ is a set of L-formulas and $\mathcal{M}$ an L-structure, then

$$M \models \Gamma \iff \forall \varphi \in \Gamma : M \models \varphi.$$

- If $\mathcal{M} \models \Gamma$ we say Γ **is satisfied** by $\mathcal{M}$ and that $\mathcal{M}$ is a model of Γ.
- Γ is said to be **satisfiable** iff it is satisfied by some model.
- We say that Γ **logically implies** A, $\Gamma \models A$, if for every $\mathcal{M} \models \Gamma$ we have $\mathcal{M} \models A$.
- If $\Gamma = \varnothing$ and $\Gamma \models A$, we write $\models A$ and say that A **is valid**.
- If a set of L-sentences T is closed under logical implication, then T is called a **theory**. A set of L-sentaces Γ is called an **axiomatization** of T is precisely the set of sentences logically implied by Γ.

Hilbert-style calculus

- $\mathcal{F}_{FO}$ is an extension of propositional Hilbert calculus $\mathcal{F}$

Hilbert-style calculus

- $\mathcal{F}_{FO}$ is an extension of propositional Hilbert calculus $\mathcal{F}$
- In addition to axioms of $\mathcal{F}$ there are two axiom schemes:

$$A(t) \supset (\exists x)A(x) \qquad \text{and} \qquad (\forall x)A(x) \supset A(t)$$

Hilbert-style calculus

- $\mathcal{F}_{FO}$ is an extension of propositional Hilbert calculus $\mathcal{F}$
- In addition to axioms of $\mathcal{F}$ there are two axiom schemes:

$$A(t) \supset (\exists x)A(x) \qquad \text{and} \qquad (\forall x)A(x) \supset A(t)$$

- And in addition to modus ponens there are two quantifier rules of inference:

$$\frac{C \supset A(x)}{C \supset (\forall x)A(x)} \qquad \text{and} \qquad \frac{A(x) \supset C}{(\exists x)A(x)}$$

where x does not appear freely in C.

Hilbert-style calculus

- $\mathcal{F}_{FO}$ is an extension of propositional Hilbert calculus $\mathcal{F}$
- In addition to axioms of $\mathcal{F}$ there are two axiom schemes:

$$A(t) \supset (\exists x)A(x) \qquad \text{and} \qquad (\forall x)A(x) \supset A(t)$$

- And in addition to modus ponens there are two quantifier rules of inference:

$$\frac{C \supset A(x)}{C \supset (\forall x)A(x)} \qquad \text{and} \qquad \frac{A(x) \supset C}{(\exists x)A(x)}$$

  where x does not appear freely in C.
- If $=$ is in the language we are considering, then we also include for every k-ary f and P the following

$$(\forall x)(x = x)$$
$$(\forall \overline{x})(\forall \overline{y})(x_1 = y_1 \wedge \cdots \wedge x_k = y_k \supset f(\overline{x}) = f(\overline{y}))$$
$$(\forall \overline{x})(\forall \overline{y})(x_1 = y_1 \wedge \cdots \wedge x_k = y_k \supset P(\overline{x}) \supset P(\overline{y})).$$

Hilbert-style calculus – soundness

Theorem (The Soundness Theorem)

Hilbert-style calculus – soundness

Theorem (The Soundness Theorem)

1. *If* $\mathcal{F}_{FO} \vdash A$, *then* $\models A$.

Hilbert-style calculus – soundness

Theorem (The Soundness Theorem)

1. *If* $\mathcal{F}_{FO} \vdash A$*, then* $\models A$.
2. *Let* Γ *bet a se t of sentences, if there is an* $\mathcal{F}_{FO}$ *proof of* A *using sentences from* Γ *as additional axioms, then* $\Gamma \models A$.

Hilbert-style calculus – soundness

Theorem (The Soundness Theorem)

1. *If* $\mathcal{F}_{FO} \vdash A$*, then* $\models A$*.*
2. *Let* Γ *bet a se t of sentences, if there is an* $\mathcal{F}_{FO}$ *proof of* A *using sentences from* Γ *as additional axioms, then* $\Gamma \models A$*.*

Proof.

By induction on the number of lines in a proofs. □

Hilbert-style calculus – completeness

Theorem (The Completeness Theorem)

[1] See the outline on wiki.

Hilbert-style calculus – completeness

Theorem (The Completeness Theorem)

1. *If* $\models A$, *then* $\mathcal{F}_{FO} \vdash A$.

[1] See the outline on wiki.

Hilbert-style calculus – completeness

Theorem (The Completeness Theorem)

1. *If* $\models A$*, then* $\mathcal{F}_{FO} \vdash A$*.*
2. *Let* Γ *bet a set of sentences. If* $\Gamma \models A$ *then there is an* $\mathcal{F}_{FO}$ *proof of* A *using sentences from* Γ *as additional axioms.*

[1] See the outline on wiki.

Hilbert-style calculus – completeness

Theorem (The Completeness Theorem)

1. *If* $\models A$*, then* $\mathcal{F}_{FO} \vdash A$*.*
2. *Let* Γ *bet a set of sentences. If* $\Gamma \models A$ *then there is an* $\mathcal{F}_{FO}$ *proof of* A *using sentences from* Γ *as additional axioms.*

- The original proof is due to Gödel [1930]. [1]

[1] See the outline on wiki.

Hilbert-style calculus – completeness

Theorem (The Completeness Theorem)

1. *If* $\models A$, *then* $\mathcal{F}_{FO} \vdash A$.
2. *Let* Γ *bet a set of sentences. If* $\Gamma \models A$ *then there is an* $\mathcal{F}_{FO}$ *proof of* A *using sentences from* Γ *as additional axioms.*

- The original proof is due to Gödel [1930]. [1]
- The now standard textbook proof is due to Henkin [1949].

[1] See the outline on wiki.

Hilbert-style calculus – completeness

Theorem (The Completeness Theorem)

1. *If* $\models A$, *then* $\mathcal{F}_{FO} \vdash A$.
2. *Let* Γ *bet a set of sentences. If* $\Gamma \models A$ *then there is an* $\mathcal{F}_{FO}$ *proof of* A *using sentences from* Γ *as additional axioms.*

- The original proof is due to Gödel [1930]. [1]
- The now standard textbook proof is due to Henkin [1949].
- We shall not give proof directly, but since $\mathcal{F}_{FO}$ can simulate cut-free fragment of first order sequent calculus it suffices to show completeness of cut-free fragment of LK proof system.

[1] See the outline on wiki.

First order sequent calculus LK

- We shall define the system LK which is an extention of PK to first-order logic.

First order sequent calculus LK

- We shall define the system LK which is an extention of PK to first-order logic.
- **!!!** Now we have two classes of variables:

First order sequent calculus LK

- We shall define the system LK which is an extention of PK to first-order logic.
- **!!!** Now we have two classes of variables:
  - Free variables: $a, b, c, \dots$ (cannot be quantified)

First order sequent calculus LK

- We shall define the system LK which is an extention of PK to first-order logic.
- **!!!** Now we have two classes of variables:
  - Free variables: $a, b, c, \ldots$ (cannot be quantified)
  - Bound variables: $z, y, x, \ldots$ (cannot occur free in formulas)

First order sequent calculus LK

- We shall define the system LK which is an extention of PK to first-order logic.
- **!!!** Now we have two classes of variables:
  - Free variables: $a, b, c, \dots$ (cannot be quantified)
  - Bound variables: $z, y, x, \dots$ (cannot occur free in formulas)
- Terms only use free variables, whereas semiterms can use both free and bound variables.

First order sequent calculus LK

- We shall define the system LK which is an extention of PK to first-order logic.
- **!!!** Now we have two classes of variables:
  - Free variables: $a, b, c, \dots$ (cannot be quantified)
  - Bound variables: $z, y, x, \dots$ (cannot occur free in formulas)
- Terms only use free variables, whereas semiterms can use both free and bound variables.
- Formulas now have the restriction, that only free variables may occur freely.

First order sequent calculus LK

- We shall define the system LK which is an extention of PK to first-order logic.
- **!!!** Now we have two classes of variables:
  - Free variables: $a, b, c, \dots$ (cannot be quantified)
  - Bound variables: $z, y, x, \dots$ (cannot occur free in formulas)
- Terms only use free variables, whereas semiterms can use both free and bound variables.
- Formulas now have the restriction, that only free variables may occur freely.
- Semiformulas may contain free occurances of bound variables.

First order sequent calculus *LK*

- We shall define the system *LK* which is an extention of *PK* to first-order logic.
- **!!!** Now we have two classes of variables:
  - Free variables: $a, b, c, \ldots$ (cannot be quantified)
  - Bound variables: $z, y, x, \ldots$ (cannot occur free in formulas)
- Terms only use free variables, whereas semiterms can use both free and bound variables.
- Formulas now have the restriction, that only free variables may occur freely.
- Semiformulas may contain free occurances of bound variables.
- **!** Generally subformula of a formula is now just a semiformula.

LK rules

- *LK* contains all the rules of inference of *PK* plus **The Quantifier Rules:**

$$\forall : \mathit{left}\,\frac{A(t), \Gamma \to \Delta}{(\forall x)A(x), \Gamma \to \Delta} \qquad \forall : \mathit{right}\,\frac{\Gamma \to \Delta, A(b)}{\Gamma \to \Delta, (\forall x)A(x)}$$

$$\exists : \mathit{left}\,\frac{A(b), \Gamma \to \Delta}{(\exists x)A(x), \Gamma \to \Delta} \qquad \exists : \mathit{right}\,\frac{\Gamma \to \Delta, A(t)}{\Gamma \to \Delta, (\exists x)A(x)}$$

in these rules, A may be an arbitrary formula, t an arbitrary term and the free variable b of $\forall$: *right* and $\exists$: *left* is called the eigenvariable of the inference and it must not appear in Γ, Δ.

LK rules

- *LK* contains all the rules of inference of *PK* plus **The Quantifier Rules:**

$$\forall : left \frac{A(t), \Gamma \rightarrow \Delta}{(\forall x)A(x), \Gamma \rightarrow \Delta} \qquad \forall : right \frac{\Gamma \rightarrow \Delta, A(b)}{\Gamma \rightarrow \Delta, (\forall x)A(x)}$$

$$\exists : left \frac{A(b), \Gamma \rightarrow \Delta}{(\exists x)A(x), \Gamma \rightarrow \Delta} \qquad \exists : right \frac{\Gamma \rightarrow \Delta, A(t)}{\Gamma \rightarrow \Delta, (\exists x)A(x)}$$

  in these rules, A may be an arbitrary formula, t an arbitrary term and the free variable b of $\forall$: *right* and $\exists$: *left* is called the eigenvariable of the inference and it must not appear in Γ, Δ.
- The propositional rules together with the quantifier rules are collectively called **logical rules**.

LK rules cont.

- Most of syntactic definitions of PK carry over to LK

LK rules cont.

- Most of syntactic definitions of PK carry over to LK
  - (direct) descendants and ancestors

LK rules cont.

- Most of syntactic definitions of *PK* carry over to *LK*
  - (direct) descendants and ancestors
  - proof lengths still measured in the number of strong inferences which include quantifier inferences.

LK rules cont.

- Most of syntactic definitions of *PK* carry over to *LK*
  - (direct) descendants and ancestors
  - proof lengths still measured in the number of strong inferences which include quantifier inferences.
- For a sequent S of the form $\Gamma \to \Delta$, we let A_S to be the formula $(\bigwedge \Gamma) \supset (\bigvee \Delta)$.

LK rules cont.

- Most of syntactic definitions of *PK* carry over to *LK*
  - (direct) descendants and ancestors
  - proof lengths still measured in the number of strong inferences which include quantifier inferences.
- For a sequent S of the form $\Gamma \to \Delta$, we let A_S to be the formula $(\bigwedge \Gamma) \supset (\bigvee \Delta)$.
- Now all the definitions of validity and logical implication apply also to sequents, where S is understood to have the meaning as A_S.

LK rules cont.

- Most of syntactic definitions of *PK* carry over to *LK*
  - (direct) descendants and ancestors
  - proof lengths still measured in the number of strong inferences which include quantifier inferences.
- For a sequent S of the form $\Gamma \to \Delta$, we let A_S to be the formula $(\bigwedge \Gamma) \supset (\bigvee \Delta)$.
- Now all the definitions of validity and logical implication apply also to sequents, where S is understood to have the meaning as A_S.
- Let the free variables of A_S be $\overline{b}$, so $A_S = A_S(\overline{b})$ and we let $\forall S$ denote the formula $(\forall x)A_S(\overline{x})$.

$LK_{\mathfrak{S}}$

- LK only allows initial sequents of the form $A \to A$ with A atomic.

$LK_{\mathfrak{S}}$

- LK only allows initial sequents of the form $A \to A$ with A atomic.
- We define $LK_{\mathfrak{S}}$ to be the system extending LK by allowing sequents from a set $\mathfrak{S}$ to be initial sequents too.

$LK_{\mathfrak{S}}$

- LK only allows initial sequents of the form $A \to A$ with A atomic.
- We define $LK_{\mathfrak{S}}$ to be the system extending LK by allowing sequents from a set $\mathfrak{S}$ to be initial sequents too.
- Important example: LK_e is the theory for first order logic with equality and we have the following additional initial segments:

$$\to s = s$$

$$s_1 = t_1, \ldots, s_k = t_k \to f(\bar{s}) = f(\bar{t})$$

$$s_1 = t_1, \ldots, s_k = t_k, P(\bar{s}) \supset P(\bar{t}).$$

$LK_{\mathfrak{S}}$

- LK only allows initial sequents of the form $A \to A$ with A atomic.
- We define $LK_{\mathfrak{S}}$ to be the system extending LK by allowing sequents from a set $\mathfrak{S}$ to be initial sequents too.
- Important example: LK_e is the theory for first order logic with equality and we have the following additional initial segments:

$$\to s = s$$

$$s_1 = t_1, \ldots, s_k = t_k \to f(\bar{s}) = f(\bar{t})$$

$$s_1 = t_1, \ldots, s_k = t_k, P(\bar{s}) \supset P(\bar{t}).$$

- We say $\mathfrak{S}$ is **closed under substitution**, if whenever $\Gamma(a) \to \Delta(a)$ is in $\mathfrak{S}$, and t is a term, then $\Gamma(t) \to \Delta(t)$ is also in $\mathfrak{S}$.

$LK_{\mathfrak{S}}$ and substitutions

Definition

We say $\mathfrak{S}$ is **closed under substitution***, if whenever $\Gamma(a) \rightarrow \Delta(a)$ is in $\mathfrak{S}$, and t is a term, then $\Gamma(t) \rightarrow \Delta(t)$ is also in $\mathfrak{S}$.*

$LK_{\mathfrak{S}}$ and substitutions

Definition

We say $\mathfrak{S}$ is **closed under substitution**, *if whenever $\Gamma(a) \to \Delta(a)$ is in $\mathfrak{S}$, and t is a term, then $\Gamma(t) \to \Delta(t)$ is also in $\mathfrak{S}$.*

Theorem

Let $\mathfrak{S}$ be a set of sequents which is closed under substitution. If $p(b)$ is a $LK_{\mathfrak{S}}$-proof, and if neither b nor any variable in t is used as an eigenvariable in $p(b)$, then $p(t)$ is a valid $LK_{\mathfrak{S}}$-proof.

Free variable normal form

Definition

A free variable in the endsequent of a proof is called a **parameter variable** *of the proof. A proof p is said to be* **free variable normal form** *provided that:*

Free variable normal form

Definition

A free variable in the endsequent of a proof is called a **parameter variable** *of the proof. A proof p is said to be* **free variable normal form** *provided that:*

1. *no parameter variable is used as an eigenvariable*

Free variable normal form

Definition

A free variable in the endsequent of a proof is called a **parameter variable** *of the proof. A proof p is said to be* **free variable normal form** *provided that:*

1. *no parameter variable is used as an eigenvariable*
2. *every other free variable appering in p is used exactly once as an eigenvariable and appears in p only in sequents above the inference for which it is used as an eigenvarible.*

Free variable normal form

Definition

A free variable in the endsequent of a proof is called a **parameter variable** *of the proof. A proof p is said to be* **free variable normal form** *provided that:*

1. *no parameter variable is used as an eigenvariable*
2. *every other free variable appering in p is used exactly once as an eigenvariable and appears in p only in sequents above the inference for which it is used as an eigenvarible.*

Free variable normal form

Definition

A free variable in the endsequent of a proof is called a **parameter variable** *of the proof. A proof p is said to be* **free variable normal form** *provided that:*

1. *no parameter variable is used as an eigenvariable*
2. *every other free variable appering in p is used exactly once as an eigenvariable and appears in p only in sequents above the inference for which it is used as an eigenvarible.*

In the rest of the talk, we only consider tree-like proofs and thus any proof may be put in free variable normal form by renaming variables.

Soundness of $LK_{\mathfrak{S}}$

Theorem

Let $\Gamma \rightarrow \Delta$ be an arbitrary sequent.

Soundness of $LK_{\mathfrak{S}}$

Theorem

Let $\Gamma \to \Delta$ be an arbitrary sequent.

1. *If $\Gamma \to \Delta$ has a LK-proof, then $\Gamma \to \Delta$ is valid.*

Soundness of $LK_{\mathfrak{S}}$

Theorem

Let $\Gamma \to \Delta$ be an arbitrary sequent.

1. *If $\Gamma \to \Delta$ has a LK-proof, then $\Gamma \to \Delta$ is valid.*
2. *Let $\mathfrak{S}$ be a set of sequents. If $\Gamma \to \Delta$ has an $LK_{\mathfrak{S}}$-proof, then $\mathfrak{S} \models \Gamma \to \Delta$.*

Completeness of cut-free LK and cut-full $LK_{\mathfrak{S}}$

Theorem

Let $\Gamma \rightarrow \Delta$ be a sequent in a first order language which does not contain equality.

Completeness of cut-free LK and cut-full $LK_{\mathfrak{S}}$

Theorem

Let $\Gamma \to \Delta$ be a sequent in a first order language which does not contain equality.

1. *If $\Gamma \to \Delta$ is valid, then it has a cut-free LK-proof.*

Completeness of cut-free LK and cut-full $LK_{\mathfrak{S}}$

Theorem

Let $\Gamma \to \Delta$ be a sequent in a first order language which does not contain equality.

1. *If $\Gamma \to \Delta$ is valid, then it has a cut-free LK-proof.*
2. *Let Π be a set of sentences. If Π logically implies $\Gamma \to \Delta$, then there are $C_1, \ldots, C_k \in \Pi$, such that $C_1, \ldots, C_k, \Gamma \to \Delta$ has a cut-free LK-proof.*

Completeness of cut-free LK and cut-full $LK_{\mathfrak{S}}$

Theorem

Let $\Gamma \to \Delta$ be a sequent in a first order language which does not contain equality.

1. *If $\Gamma \to \Delta$ is valid, then it has a cut-free LK-proof.*
2. *Let Π be a set of sentences. If Π logically implies $\Gamma \to \Delta$, then there are $C_1, \ldots, C_k \in \Pi$, such that $C_1, \ldots, C_k, \Gamma \to \Delta$ has a cut-free LK-proof.*

Corollary

Let $\mathfrak{S}$ be a set of sequents. If $\mathfrak{S}$ logically implies $\Gamma \to \Delta$, then $\Gamma \to \Delta$ has a $LK_{\mathfrak{S}}$-proof.

Completeness of cut-free LK and cut-full $LK_{\mathfrak{S}}$

Theorem

Let $\Gamma \to \Delta$ be a sequent in a first order language which does not contain equality.

1. *If $\Gamma \to \Delta$ is valid, then it has a cut-free LK-proof.*
2. *Let Π be a set of sentences. If Π logically implies $\Gamma \to \Delta$, then there are $C_1, \ldots, C_k \in \Pi$, such that $C_1, \ldots, C_k, \Gamma \to \Delta$ has a cut-free LK-proof.*

Corollary

Let $\mathfrak{S}$ be a set of sequents. If $\mathfrak{S}$ logically implies $\Gamma \to \Delta$, then $\Gamma \to \Delta$ has a $LK_{\mathfrak{S}}$-proof.

Proof.

If $\mathfrak{S} \models \Gamma \to \Delta$, then (2) implies that there are $S_1, \ldots, S_k \in \mathfrak{S}$ such that $\forall S_1, \ldots, \forall S_k, \Gamma \to \Delta$ has an LK-proof. Each $\to \forall S_i$ has an $LK_{\mathfrak{S}}$-proof, so with k further cut inferences, we obtain $\Gamma \to \Delta$. $\square$

Proof of completeness

Theorem

Let $\Gamma \to \Delta$ be a sequent in a first order language which does not contain equality.

Proof of completeness

Theorem

Let $\Gamma \to \Delta$ be a sequent in a first order language which does not contain equality.

2. *Let Π be a set of sentences. If Π logically implies $\Gamma \to \Delta$, then there are $C_1, \ldots, C_k \in \Pi$, such that $C_1, \ldots, C_k, \Gamma \to \Delta$ has a cut-free LK-proof.*

Proof of completeness

Theorem

Let $\Gamma \to \Delta$ be a sequent in a first order language which does not contain equality.

2. *Let Π be a set of sentences. If Π logically implies $\Gamma \to \Delta$, then there are $C_1, \ldots, C_k \in \Pi$, such that $C_1, \ldots, C_k, \Gamma \to \Delta$ has a cut-free LK-proof.*

Proof.

We shall only prove the case where Π is countable (and therefore L is). We assume $\Pi \models \Gamma \to \Delta$. We shall try to build up a proof of $C_1, \ldots, C_k, \Gamma \to \Delta$ from the bottom up. Quantifiers make this a potentially infinite process so we need to show that the construction terminates.

Proof of completeness

Theorem

Proof of completeness

Theorem

2. *Let* Π *be a set of sentences. If* Π *logically implies* $\Gamma \to \Delta$*, then there are* $C_1, \ldots, C_k \in \Pi$*, such that* $C_1, \ldots, C_k, \Gamma \to \Delta$ *has a cut-free* LK*-proof.*

Proof of completeness

Theorem

2. *Let Π be a set of sentences. If Π logically implies $\Gamma \to \Delta$, then there are $C_1, \ldots, C_k \in \Pi$, such that $C_1, \ldots, C_k, \Gamma \to \Delta$ has a cut-free LK-proof.*

Proof cont.

We start with an (incomplete) proof p_0 which consists of just the sequent $\Gamma \to \Delta$ and we will proceed by building stages p_i. A leaf sequent of p_i is called **active** if no formula occurs in both its cedents.
Let $A_1, A_2, \ldots$ be a sequence of all L-formulas where every formula is repeated infinitely many times. Let $t_1, t_2, \ldots$ be a sequence of all L-terms where every term is repeated. And we enumerate all pairs (A_j, t_k) using diagonal enumeration. We shall construct p_{i+1} using (A_{j_i}, t_{k_i}) and p_i.

Proof of completeness

Theorem

Proof of completeness

Theorem

2. *Let Π be a set of sentences. If Π logically implies $\Gamma \to \Delta$, then there are $C_1, \ldots, C_k \in \Pi$, such that $C_1, \ldots, C_k, \Gamma \to \Delta$ has a cut-free LK-proof.*

Proof of completeness

Theorem

2. *Let Π be a set of sentences. If Π logically implies $\Gamma \to \Delta$, then there are $C_1, \dots, C_k \in \Pi$, such that $C_1, \dots, C_k, \Gamma \to \Delta$ has a cut-free LK-proof.*

Proof cont.

Let (A_j, t_k) be the pair for p_i.
(1) If $A_j \in \Pi$ then replace every sequent $\Gamma' \to \Delta'$ in P with the sequent $\Gamma', A_i \to \Delta'$.

Proof of completeness

Theorem

Proof of completeness

Theorem

2. *Let* Π *be a set of sentences. If* Π *logically implies* $\Gamma \to \Delta$*, then there are* $C_1, \dots, C_k \in \Pi$*, such that* $C_1, \dots, C_k, \Gamma \to \Delta$ *has a cut-free* LK*-proof.*

Proof of completeness

Theorem

2. *Let Π be a set of sentences. If Π logically implies $\Gamma \to \Delta$, then there are $C_1, \ldots, C_k \in \Pi$, such that $C_1, \ldots, C_k, \Gamma \to \Delta$ has a cut-free LK-proof.*

Proof cont.

(2) If $A_j \notin \Pi$ is not atomic, we shall modify all active sequents which contain A_i as follows:
(2a) If A_j is $\neg B$, then active sequents of the form $\Gamma', \neg B, \Gamma'' \to \Delta'$ are replaced by the derivation:

$$\frac{\Gamma', \neg B, \Gamma'' \to \Delta', B}{\Gamma', \neg B, \Gamma'' \to \Delta'}$$

and ones of the form $\Gamma' \to \Delta', \neg B, \Delta''$ with the dual derivation.

Proof of completeness

Theorem

Proof of completeness

Theorem

2. *Let* Π *be a set of sentences. If* Π *logically implies* $\Gamma \rightarrow \Delta$*, then there are* $C_1, \ldots, C_k \in \Pi$*, such that* $C_1, \ldots, C_k, \Gamma \rightarrow \Delta$ *has a cut-free LK-proof.*

Proof of completeness

Theorem

2. *Let Π be a set of sentences. If Π logically implies $\Gamma \to \Delta$, then there are $C_1, \dots, C_k \in \Pi$, such that $C_1, \dots, C_k, \Gamma \to \Delta$ has a cut-free LK-proof.*

Proof cont.

(2b) If A_j is $B \vee C$, then active sequents of the form $\Gamma', B \vee C, \Gamma'' \to \Delta'$ are replaced by the derivation:

$$\frac{B, \Gamma', B \vee C, \Gamma'' \to \Delta' \qquad C, \Gamma', B \vee C, \Gamma'' \to \Delta'}{B, \Gamma', B \vee C, \Gamma'' \to \Delta', B}$$

and ones of the form $\Gamma' \to \Delta', B \vee C, \Delta''$ with

$$\frac{\Gamma' \to \Delta', B \vee C, \Delta'', B, C}{\Gamma' \to \Delta', B \vee C, \Delta''}.$$

Proof of completeness

Theorem

Proof of completeness

Theorem

2. *Let* Π *be a set of sentences. If* Π *logically implies* $\Gamma \to \Delta$*, then there are* $C_1, \ldots, C_k \in \Pi$*, such that* $C_1, \ldots, C_k, \Gamma \to \Delta$ *has a cut-free* LK*-proof.*

Proof of completeness

Theorem

2. *Let Π be a set of sentences. If Π logically implies $\Gamma \to \Delta$, then there are $C_1, \dots, C_k \in \Pi$, such that $C_1, \dots, C_k, \Gamma \to \Delta$ has a cut-free LK-proof.*

Proof cont.

(2c)(2d) Analogously for $\wedge$ and $\supset$. (2e) A_j is $(\exists x)B(c)$ and an active sequent in p_i is of the form $\Gamma', (\exists x)B, \Gamma'' \to \Delta'$ we replace it with

$$\frac{B(c), \Gamma', (\exists x)B(x), \Gamma'' \to \Delta'}{\Gamma', (\exists B), \Gamma'' \to \Delta'}$$

where c is a new variable. In the case where the form is $\Gamma' \to \Delta', (\exists x)B(x), \Delta''$ we replace it with

$$\frac{\Gamma' \to \Delta', (\exists x), \Delta'', B(t_j)}{\Gamma' \to \Delta', (\exists x)B(x), \Delta''.}$$

Proof of completeness – bulilding $\mathcal{M}$ from p_i's

Lemma

If the proccess of building p_i's never stops (so no complete proof is formed), we can build $\mathcal{M} \models \Pi$ and σ such that $\mathcal{M} \not\models (\Gamma \to \Delta)[\sigma]$.

Proof of completeness – bulilding $\mathcal{M}$ from p_i's

Lemma

If the proccess of building p_i's never stops (so no complete proof is formed), we can build $\mathcal{M} \models \Pi$ and σ such that $\mathcal{M} \not\models (\Gamma \to \Delta)[\sigma]$.

Proof cont.

Assume it is always possible to build p_{i+1} and consider $p = \bigcup_{i\to\infty} p_i$ (form a union of the proof-trees).
Unless $\Gamma \to \Delta$ contains only atomic formulas, p is an infinite finitely branching tree and thus by König's lemma there is an infinite branch π in p.
We define the universe of $\mathcal{M}$ to be the set of all L-terms, we let σ map all variables to themselves, and define $f^{\mathcal{M}}(r_1, \ldots, r_k)$ to be just $f(r_1, \ldots, r_k)$ and $P^{\mathcal{M}}(r_1, \ldots, r_k)$ holds iff $P(r_1, \ldots, r_k)$ appears in an antecedent in π.
Note that if $P(r_1, \ldots, r_k)$ were in both antecedent and succedent of some sequent in π, π would have not become infinite.

Proof of completeness – bulilding $\mathcal{M}$ from p_i's

Lemma

If the proccess of building p_i's never stops (so no complete proof is formed), we can build $\mathcal{M} \models \Pi$ and σ such that $\mathcal{M} \not\models (\Gamma \rightarrow \Delta)[\sigma]$.

Proof of completeness – bulilding $\mathcal{M}$ from p_i's

Lemma

If the proccess of building p_i's never stops (so no complete proof is formed), we can build $\mathcal{M} \models \Pi$ and σ such that $\mathcal{M} \not\models (\Gamma \to \Delta)[\sigma]$.

Proof cont.

Now it is enough to show that every formula in an antecedent (resp. a succedent) along π is true (resp. false) in $\mathcal{M}$. We proceed by the induction on the complexity of A. For A atomic it holds by definition. For A of the form $(\exists x)B(x)$ in an antecedent, we have $B(c)$ further up π also in an antecedent by construction, if such A is in the succedent, then, for every t, $B(t)$ eventually appears in the succedent. The rest of the cases are analogous to these. $\square$

$$\frac{\text{from my ignorance?} \rightarrow \text{Excuse me}}{\text{Thank you} \rightarrow \text{for your attention!}}$$