Takeuti's conjecture

And Prawitz's proof

Jelle Tjeerd Fokkens

Initial Types Club

April 8, 2021

Outline

Introduction

- Gerhard Gentzen, 1934 proved the Hauptsatz for first order sequent calculus.

Introduction

- Gerhard Gentzen, 1934 proved the Hauptsatz for first order sequent calculus.
- Gaisi Takeuti stated the conjecture for STT in 1953.

Introduction

- Gerhard Gentzen, 1934 proved the Hauptsatz for first order sequent calculus.
- Gaisi Takeuti stated the conjecture for STT in 1953.
- Proved by Tait for second-order logic in 1966.

Introduction

- Gerhard Gentzen, 1934 proved the Hauptsatz for first order sequent calculus.
- Gaisi Takeuti stated the conjecture for STT in 1953.
- Proved by Tait for second-order logic in 1966.
- Proved independently by Motoo Takahashi and Dag Prawitz in 1967.

The challenge

- Russell's paradox: 'The class of all classes that do not contain themselves as elements.'

The challenge

- Russell's paradox: 'The class of all classes that do not contain themselves as elements.'
- Supposed underlying problem: self-reference.

The challenge

- Russell's paradox: 'The class of all classes that do not contain themselves as elements.'
- Supposed underlying problem: self-reference.
- Impredicativity is supposed to be bad.

$$\frac{\Gamma \vdash \forall X A(X)}{\Gamma \vdash A(X := T)}$$, where T is any term.

The challenge

- Russell's paradox: 'The class of all classes that do not contain themselves as elements.'
- Supposed underlying problem: self-reference.
- Impredicativity is supposed to be bad.

$$\frac{\Gamma \vdash \forall X A(X)}{\Gamma \vdash A(X := T)}$$, where T is any term.

- Furthermore, extensionality:

$$\forall x (P(x) \leftrightarrow Q(x)) \rightarrow (RP \rightarrow RQ)$$

Proof outline

We're proving three claims which together imply the Hauptsatz.

1. if A is not derivable without cut then there is a semi-valuation with A false

Proof outline

We're proving three claims which together imply the Hauptsatz.

1. if A is not derivable without cut then there is a semi-valuation with A false
2. semi-valuations are extendable to total valuations

Proof outline

We're proving three claims which together imply the Hauptsatz.

1. if A is not derivable without cut then there is a semi-valuation with A false
2. semi-valuations are extendable to total valuations
3. if A is false in a total valuation, then A is not derivable

So, if A is derivable, it is derivable without cut.

STT

- Types:
  - 0 (individuals) and 1 (truth values) are types

STT

- Types:
  - 0 (individuals) and 1 (truth values) are types
  - if $\tau_1, ..., \tau_n$ are types, then $(\tau_1, ..., \tau_n)$ is a type (predicate with indicated argument places)

STT

- Types:
  - 0 (individuals) and 1 (truth values) are types
  - if $\tau_1, ..., \tau_n$ are types, then $(\tau_1, ..., \tau_n)$ is a type (predicate with indicated argument places)
- Basic symbols:
  - free (a_i^τ) and bound (x_i^τ) variables of any type.

STT

- Types:
  - 0 (individuals) and 1 (truth values) are types
  - if $\tau_1, \ldots, \tau_n$ are types, then $(\tau_1, \ldots, \tau_n)$ is a type (predicate with indicated argument places)
- Basic symbols:
  - free (a_i^τ) and bound (x_i^τ) variables of any type.
  - constants (c_i^τ) of any type.

STT

- Types:
  - 0 (individuals) and 1 (truth values) are types
  - if $\tau_1, ..., \tau_n$ are types, then $(\tau_1, ..., \tau_n)$ is a type (predicate with indicated argument places)
- Basic symbols:
  - free (a_i^τ) and bound (x_i^τ) variables of any type.
  - constants (c_i^τ) of any type.
  - function symbols (f_i) denoting functions with arguments and values of type 0.

STT

- Types:
  - 0 (individuals) and 1 (truth values) are types
  - if $\tau_1, ..., \tau_n$ are types, then $(\tau_1, ..., \tau_n)$ is a type (predicate with indicated argument places)
- Basic symbols:
  - free (a_i^τ) and bound (x_i^τ) variables of any type.
  - constants (c_i^τ) of any type.
  - function symbols (f_i) denoting functions with arguments and values of type 0.
  - $\neg$ for negation, $\vee$ for disjunction, $\exists$ for existence, λ for abstraction and $\in$ for membership.

STT, continued

Expressions:

STT, continued

Expressions:

- free variables and constants of type τ is an expression of type τ

STT, continued

Expressions:

- free variables and constants of type τ is an expression of type τ
- if ϕ is a function symbol with n argument places and $e_1, \ldots, e_n$ are expressions of type 0, then $\phi(e_1, \ldots, e_n)$ is an expression of type 0

STT, continued

Expressions:

- free variables and constants of type τ is an expression of type τ
- if ϕ is a function symbol with n argument places and $e_1, \ldots, e_n$ are expressions of type 0, then $\phi(e_1, \ldots, e_n)$ is an expression of type 0
- if $e_1, \ldots, e_n$ are expressions of types $\tau_1, \ldots, \tau_n$, and e is an expression of type $(\tau_1, \ldots, \tau_n)$, then $(e_1, \ldots, e_n \in e)$ is an expression of type 1

STT, continued

Expressions:

- free variables and constants of type τ is an expression of type τ
- if ϕ is a function symbol with n argument places and $e_1, \dots, e_n$ are expressions of type 0, then $\phi(e_1, \dots, e_n)$ is an expression of type 0
- if $e_1, \dots, e_n$ are expressions of types $\tau_1, \dots, \tau_n$, and e is an expression of type $(\tau_1, \dots, \tau_n)$, then $(e_1, \dots, e_n \in e)$ is an expression of type 1
- if A is an expression of type 1, then $\neg A$ is an expression of type 1

STT, continued

Expressions:

- free variables and constants of type τ is an expression of type τ
- if ϕ is a function symbol with n argument places and $e_1, \ldots, e_n$ are expressions of type 0, then $\phi(e_1, \ldots, e_n)$ is an expression of type 0
- if $e_1, \ldots, e_n$ are expressions of types $\tau_1, \ldots, \tau_n$, and e is an expression of type $(\tau_1, \ldots, \tau_n)$, then $(e_1, \ldots, e_n \in e)$ is an expression of type 1
- if A is an expression of type 1, then $\neg A$ is an expression of type 1
- if A and B are expressions of type 1, then $(A \vee B)$ is an expression of type 1

STT, continued

Expressions:

- free variables and constants of type τ is an expression of type τ
- if ϕ is a function symbol with n argument places and $e_1, ..., e_n$ are expressions of type 0, then $\phi(e_1, ..., e_n)$ is an expression of type 0
- if $e_1, ..., e_n$ are expressions of types $\tau_1, ..., \tau_n$, and e is an expression of type $(\tau_1, ..., \tau_n)$, then $(e_1, ..., e_n \in e)$ is an expression of type 1
- if A is an expression of type 1, then $\neg A$ is an expression of type 1
- if A and B are expressions of type 1, then $(A \vee B)$ is an expression of type 1
- if x^τ is a bound variable which does not occur in an expression $A(a^\tau)$ of type 1, then $\exists x^\tau A(x^\tau)$ is an expression of type 1.

STT, continued

Expressions:

- free variables and constants of type τ is an expression of type τ
- if ϕ is a function symbol with n argument places and $e_1, \ldots, e_n$ are expressions of type 0, then $\phi(e_1, \ldots, e_n)$ is an expression of type 0
- if $e_1, \ldots, e_n$ are expressions of types $\tau_1, \ldots, \tau_n$, and e is an expression of type $(\tau_1, \ldots, \tau_n)$, then $(e_1, \ldots, e_n \in e)$ is an expression of type 1
- if A is an expression of type 1, then $\neg A$ is an expression of type 1
- if A and B are expressions of type 1, then $(A \vee B)$ is an expression of type 1
- if x^τ is a bound variable which does not occur in an expression $A(a^\tau)$ of type 1, then $\exists x^\tau A(x^\tau)$ is an expression of type 1.
- if $x_1^{\tau_1}, \ldots, x_n^{\tau_n}$ are different bound variables which do not occur in an expression $A(a_1^{\tau_1}, \ldots, a_n^{\tau_n})$ of type 1, then $\lambda x_1^{\tau_1} \ldots x_n^{\tau_n} A(x_1^{\tau_1}, \ldots, x_n^{\tau_n})$ is an expression of type $(\tau_1, \ldots, \tau_n)$.

Step 1

Definition

A *semi-valuation* is an assignment of at most one of the two values t and f to any wff with the following conditions:

- if $\neg A$ is t, then A is f

Step 1

Definition

A *semi-valuation* is an assignment of at most one of the two values t and f to any wff with the following conditions:

- if $\neg A$ is t, then A is f
- if $\neg A$ is f, then A is t

Step 1

Definition

A *semi-valuation* is an assignment of at most one of the two values t and f to any wff with the following conditions:

- if $\neg A$ is t, then A is f
- if $\neg A$ is f, then A is t
- if $A \vee B$ is t, then at least one of the wffs A and B is t.

Step 1

Definition

A *semi-valuation* is an assignment of at most one of the two values t and f to any wff with the following conditions:

- if $\neg A$ is t, then A is f
- if $\neg A$ is f, then A is t
- if $A \vee B$ is t, then at least one of the wffs A and B is t.
- if $A \vee B$ is f, then both wffs A and B are f.

Step 1

Definition

A *semi-valuation* is an assignment of at most one of the two values t and f to any wff with the following conditions:

- if $\neg A$ is t, then A is f
- if $\neg A$ is f, then A is t
- if $A \vee B$ is t, then at least one of the wffs A and B is t.
- if $A \vee B$ is f, then both wffs A and B are f.
- if $\exists x^\tau A(x^\tau)$ is t, then there exists an expression e^τ of type τ such that $A(e^\tau)$ is t.

Step 1

Definition

A *semi-valuation* is an assignment of at most one of the two values t and f to any wff with the following conditions:

- if $\neg A$ is t, then A is f
- if $\neg A$ is f, then A is t
- if $A \vee B$ is t, then at least one of the wffs A and B is t.
- if $A \vee B$ is f, then both wffs A and B are f.
- if $\exists x^\tau A(x^\tau)$ is t, then there exists an expression e^τ of type τ such that $A(e^\tau)$ is t.
- if $\exists x^\tau A(x^\tau)$ is f, then $A(e^\tau)$ is f for every expression e^τ of type τ.

Step 1

Definition

A *semi-valuation* is an assignment of at most one of the two values t and f to any wff with the following conditions:

- if $\neg A$ is t, then A is f
- if $\neg A$ is f, then A is t
- if $A \vee B$ is t, then at least one of the wffs A and B is t.
- if $A \vee B$ is f, then both wffs A and B are f.
- if $\exists x^\tau A(x^\tau)$ is t, then there exists an expression e^τ of type τ such that $A(e^\tau)$ is t.
- if $\exists x^\tau A(x^\tau)$ is f, then $A(e^\tau)$ is f for every expression e^τ of type τ.
- if $(e_1, ..., e_n \in \lambda x_1 ... x_n A(x_1, ..., x_n))$ has a value, then $A(e_1, ..., e_n)$ has the same value.

Step 1.1

Definition

- Every variable or constant of type 1 is a *prime formula*

Step 1.1

Definition

- Every variable or constant of type 1 is a *prime formula*
- A formula $(e_1, ..., e_n \in e)$ is a *prime formula* if e is a free variable or a constant.

Step 1.1

Definition

- Every variable or constant of type 1 is a *prime formula*
- A formula $(e_1, ..., e_n \in e)$ is a *prime formula* if e is a free variable or a constant.

Definition

By recursion.

- F is a *positive part* (pp) of F

Step 1.1

Definition

- Every variable or constant of type 1 is a *prime formula*
- A formula $(e_1, \ldots, e_n \in e)$ is a *prime formula* if e is a free variable or a constant.

Definition

By recursion.

- F is a *positive part* (pp) of F
- if $\neg A$ is a pp of F, then A is a *negative part* (np) of F

Step 1.1

Definition

- Every variable or constant of type 1 is a *prime formula*
- A formula $(e_1, \ldots, e_n \in e)$ is a *prime formula* if e is a free variable or a constant.

Definition

By recursion.

- F is a *positive part* (pp) of F
- if $\neg A$ is a pp of F, then A is a *negative part* (np) of F
- if $\neg A$ is an np of F, then A is a pp of F

Step 1.1

Definition

- Every variable or constant of type 1 is a *prime formula*
- A formula $(e_1, \ldots, e_n \in e)$ is a *prime formula* if e is a free variable or a constant.

Definition

By recursion.

- F is a *positive part* (pp) of F
- if $\neg A$ is a pp of F, then A is a *negative part* (np) of F
- if $\neg A$ is an np of F, then A is a pp of F
- if $A \vee B$ is a pp of F, then A and B are pps of F

Step 1.1

Definition

- Every variable or constant of type 1 is a *prime formula*
- A formula $(e_1, \ldots, e_n \in e)$ is a *prime formula* if e is a free variable or a constant.

Definition

By recursion.

- F is a *positive part* (pp) of F
- if $\neg A$ is a pp of F, then A is a *negative part* (np) of F
- if $\neg A$ is an np of F, then A is a pp of F
- if $A \vee B$ is a pp of F, then A and B are pps of F

Remark: if A is a true pp or false np of F, then F is true.

Step 1.2

Definition

- Every axiom $F[P_+, P_-]$, where P is a prime formula, is *derivable*.

Step 1.2

Definition

- Every axiom $F[P_{+}, P_{-}]$, where P is a prime formula, is *derivable*.
- If the premises of a basic inference are *derivable*, then the conclusion is derivable.

Step 1.2

Definition

- Every axiom $F[P_+, P_-]$, where P is a prime formula, is *derivable*.
- If the premises of a basic inference are *derivable*, then the conclusion is derivable.
- A wff is *strictly derivable* if no use is made of S5.

Step 1.2

Definition

- Every axiom $F[P_+, P_-]$, where P is a prime formula, is *derivable*.
- If the premises of a basic inference are *derivable*, then the conclusion is derivable.
- A wff is *strictly derivable* if no use is made of S5.

The basic inferences:

$$\text{S1}\ \frac{F[A_-], F[B_-]}{F[(A \vee B)_-]} \qquad \text{S4a}\ \frac{F[A(e_1, ..., e_n)_+]}{F[(e_1, ..., e_n \in \lambda x_1 ... x_n A(x_1, ..., x_n))_+]}$$

$$\text{S2}\ \frac{F[A(a^\tau)_-]}{F[\exists x^\tau A(x^\tau)_-]} \qquad \text{S4b}\ \frac{F[A(e_1, ..., e_n)_-]}{F[(e_1, ..., e_n \in \lambda x_1 ... x_n A(x_1, ..., x_n))_-]}$$

$$\text{S3}\ \frac{F[\exists x^\tau A(x^\tau)_+] \vee A(e^\tau)}{F[\exists x^\tau A(x^\tau)_+]} \qquad \text{S5}\ \frac{F \vee \exists x^1 \neg(x^1 \vee \neg x^1)}{F}$$

Step 1.3

Definition

A *deduction string* of a wff F is a finite or infinite sequence of wffs $F_1, F_2, F_3, \ldots$ that satisfies:

- $F_1 = F$

Step 1.3

Definition

A *deduction string* of a wff F is a finite or infinite sequence of wffs $F_1, F_2, F_3, \ldots$ that satisfies:

- $F_1 = F$
- if F_n is neither an axiom nor a primitive wff, then F_{n+1} is a reductum or a weakening of F_n.

Step 1.3

Definition

A *deduction string* of a wff F is a finite or infinite sequence of wffs $F_1, F_2, F_3, \ldots$ that satisfies:

- $F_1 = F$
- if F_n is neither an axiom nor a primitive wff, then F_{n+1} is a reductum or a weakening of F_n.
- if F_n is an axiom or a primitive wff, then F_n is the last wff of the string.

Step 1.3

Definition

A *deduction string* of a wff F is a finite or infinite sequence of wffs $F_1, F_2, F_3, \ldots$ that satisfies:

- $F_1 = F$
- if F_n is neither an axiom nor a primitive wff, then F_{n+1} is a reductum or a weakening of F_n.
- if F_n is an axiom or a primitive wff, then F_n is the last wff of the string.

No use is made of rule S5.

Step 1.4

Theorem

If every deduction string of a wff F contains an axiom, then F is strictly derivable.

Step 1.4

Theorem

If every deduction string of a wff F contains an axiom, then F is strictly derivable.

Proof.

All the deduction strings can be combined together to form a deduction tree of F. This deduction tree is finite, because there are finitely many deduction strings and the tree has finite branching. Only inferences S1-S4 are being used in the derivation, so no use is made of the cut rule. $\square$

Step 1.5

Theorem

If a wff F has a deduction string which does not contain an axiom, then there exists a semi-valuation V in which F is f.

Step 1.5

Theorem

If a wff F has a deduction string which does not contain an axiom, then there exists a semi-valuation V in which F is f.

Proof.

Let ϕ be the deduction string $(F =)F_1, F_2, F_3, ...$ not containing an axiom. We define $V(e) = t$ for e a negative part in ϕ and $V(e) = f$ for e a positive part in ϕ. V is a semi-valuation, by induction on the rank of a wff A:

Step 1.5

Theorem

If a wff F has a deduction string which does not contain an axiom, then there exists a semi-valuation V in which F is f.

Proof.

Let ϕ be the deduction string $(F =)F_1, F_2, F_3, \ldots$ not containing an axiom. We define $V(e) = t$ for e a negative part in ϕ and $V(e) = f$ for e a positive part in ϕ. V is a semi-valuation, by induction on the rank of a wff A:

- IB: There is no prime formula P with $V(P) = t = f$.

Step 1.5

Theorem

If a wff F has a deduction string which does not contain an axiom, then there exists a semi-valuation V in which F is f.

Proof.

Let ϕ be the deduction string $(F =)F_1, F_2, F_3, \ldots$ not containing an axiom. We define $V(e) = t$ for e a negative part in ϕ and $V(e) = f$ for e a positive part in ϕ. V is a semi-valuation, by induction on the rank of a wff A:

- IB: There is no prime formula P with $V(P) = t = f$.
- IS: V satisfies all the conditions of a semi-valuation.

Step 1.5

Theorem

If a wff F has a deduction string which does not contain an axiom, then there exists a semi-valuation V in which F is f.

Proof.

Let ϕ be the deduction string $(F =)F_1, F_2, F_3, \ldots$ not containing an axiom. We define $V(e) = t$ for e a negative part in ϕ and $V(e) = f$ for e a positive part in ϕ. V is a semi-valuation, by induction on the rank of a wff A:

- IB: There is no prime formula P with $V(P) = t = f$.
- IS: V satisfies all the conditions of a semi-valuation.

Also, because F is a positive part of itself: $V(F) = f$.

□

Interlude

We proved:

Interlude

We proved:

Theorem

If every deduction string of a wff F contains an axiom, then F is strictly derivable.

Interlude

We proved:

Theorem

If every deduction string of a wff F contains an axiom, then F is strictly derivable.

And we proved also:

Interlude

We proved:

Theorem

If every deduction string of a wff F contains an axiom, then F is strictly derivable.

And we proved also:

Theorem

If a wff F has a deduction string which does not contain an axiom, then there exists a semi-valuation V in which F is f.

Interlude

We proved:

Theorem

If every deduction string of a wff F contains an axiom, then F is strictly derivable.

And we proved also:

Theorem

If a wff F has a deduction string which does not contain an axiom, then there exists a semi-valuation V in which F is f.

Combining this, we get:

Interlude

We proved:

Theorem

If every deduction string of a wff F contains an axiom, then F is strictly derivable.

And we proved also:

Theorem

If a wff F has a deduction string which does not contain an axiom, then there exists a semi-valuation V in which F is f.

Combining this, we get:

Theorem

If a wff F is not derivable without cut, then there exists a semi-valuation V in which F is f.

Interlude

We proved:

Theorem

If every deduction string of a wff F contains an axiom, then F is strictly derivable.

And we proved also:

Theorem

If a wff F has a deduction string which does not contain an axiom, then there exists a semi-valuation V in which F is f.

Combining this, we get:

Theorem

If a wff F is not derivable without cut, then there exists a semi-valuation V in which F is f.

And thereby we proved the first step.

Step 2

Assume a semi-valuation V fixed.

Definition

Basic set contains:

- variables of any type

Step 2

Assume a semi-valuation V fixed.

Definition

Basic set contains:

- variables of any type
- arbitrarily many constants of certain types

Step 2

Assume a semi-valuation V fixed.

Definition

Basic set contains:

- variables of any type
- arbitrarily many constants of certain types
- arbitrarily many function symbols with certain argument places.

Step 2

Assume a semi-valuation V fixed.

Definition

Basic set contains:

- variables of any type
- arbitrarily many constants of certain types
- arbitrarily many function symbols with certain argument places.

Definition

Basic set S:

- $P = \{e | V(e) \text{is defined}\}$

Step 2

Assume a semi-valuation V fixed.

Definition

Basic set contains:

- variables of any type
- arbitrarily many constants of certain types
- arbitrarily many function symbols with certain argument places.

Definition

Basic set S:

- $P = \{e | V(e) \text{is defined}\}$
- $Q = \{s | s \text{is a symbol occurring in } P\}$

Step 2

Assume a semi-valuation V fixed.

Definition

Basic set contains:

- variables of any type
- arbitrarily many constants of certain types
- arbitrarily many function symbols with certain argument places.

Definition

Basic set S:

- $P = \{e | V(e) \text{is defined}\}$
- $Q = \{s | s \text{is a symbol occurring in } P\}$
- $R = \{e | e \text{is expression with only symbols from } Q\}$

Step 2

Assume a semi-valuation V fixed.

Definition

Basic set contains:

- variables of any type
- arbitrarily many constants of certain types
- arbitrarily many function symbols with certain argument places.

Definition

Basic set S:

- $P = \{e | V(e) \text{is defined}\}$
- $Q = \{s | s \text{is a symbol occurring in } P\}$
- $R = \{e | e \text{is expression with only symbols from } Q\}$
- $S =$ least basic set containing R

Step 2.1

Definition

Entities of finite type are defined by induction on τ:

- E is an entity of type 0 iff E is an expression of type 0 over S.

Step 2.1

Definition

Entities of finite type are defined by induction on τ:

- E is an entity of type 0 iff E is an expression of type 0 over S.
- E is an entity of type 1 iff E is t or f.

Step 2.1

Definition

Entities of finite type are defined by induction on τ:

- E is an entity of type 0 iff E is an expression of type 0 over S.
- E is an entity of type 1 iff E is t or f.
- E is an entity of type $(\tau_1, ..., \tau_n)$ iff E is a set of n-tuples $((e_1, E_1), ..., (e_n, E_n))$ where e_i is an expression of type τ_i over S and E_i is an entity of type τ_i.

Step 2.2

Definition

Possible values of expressions. By induction over the types:

Step 2.2

Definition

Possible values of expressions. By induction over the types:

- if e is of type 0, then it is a possible value of itself.

Step 2.2

Definition

Possible values of expressions. By induction over the types:

- if e is of type 0, then it is a possible value of itself.
- if e is of type 1, then E is a possible value of e iff

Step 2.2

Definition

Possible values of expressions. By induction over the types:

- if e is of type 0, then it is a possible value of itself.
- if e is of type 1, then E is a possible value of e iff
  - E is t or f, and

Step 2.2

Definition

Possible values of expressions. By induction over the types:

- if e is of type 0, then it is a possible value of itself.
- if e is of type 1, then E is a possible value of e iff
  - E is t or f, and
  - if V assigns some value to e, then $E = V(e)$.

Step 2.2

Definition

Possible values of expressions. By induction over the types:

- if e is of type 0, then it is a possible value of itself.
- if e is of type 1, then E is a possible value of e iff
  - E is t or f, and
  - if V assigns some value to e, then $E = V(e)$.
- if e is of type $(\tau_1, ..., \tau_n)$, then E is a possible value of e iff:

Step 2.2

Definition

Possible values of expressions. By induction over the types:

- if e is of type 0, then it is a possible value of itself.
- if e is of type 1, then E is a possible value of e iff
  - E is t or f, and
  - if V assigns some value to e, then $E = V(e)$.
- if e is of type $(\tau_1, ..., \tau_n)$, then E is a possible value of e iff:
  - E is a set of elements are n-tuples $((e_1, E_1), ..., (e_n, E_n))$ where e_i is an expression of type τ_i over S and E_i is some possible value of e_i.

Step 2.2

Definition

Possible values of expressions. By induction over the types:

- if e is of type 0, then it is a possible value of itself.
- if e is of type 1, then E is a possible value of e iff
  - E is t or f, and
  - if V assigns some value to e, then $E = V(e)$.
- if e is of type $(\tau_1, ..., \tau_n)$, then E is a possible value of e iff:
  - E is a set of elements are n-tuples $((e_1, E_1), ..., (e_n, E_n))$ where e_i is an expression of type τ_i over S and E_i is some possible value of e_i.
  - if V assigns t to an expression $(e_1, ..., e_n) \in e$ over S and each E_i is a possible value of e_i, then $((e_1, E_1), ..., (e_n, E_n))$ belongs to E.

Step 2.2

Definition

Possible values of expressions. By induction over the types:

- if e is of type 0, then it is a possible value of itself.
- if e is of type 1, then E is a possible value of e iff
  - E is t or f, and
  - if V assigns some value to e, then $E = V(e)$.
- if e is of type $(\tau_1, ..., \tau_n)$, then E is a possible value of e iff:
  - E is a set of elements are n-tuples $((e_1, E_1), ..., (e_n, E_n))$ where e_i is an expression of type τ_i over S and E_i is some possible value of e_i.
  - if V assigns t to an expression $(e_1, ..., e_n) \in e$ over S and each E_i is a possible value of e_i, then $((e_1, E_1), ..., (e_n, E_n))$ belongs to E.
  - if V assigns f to an expression $(e_1, ..., e_n) \in e$ over S and each E_i is a possible value of e_i, then $((e_1, E_1), ..., (e_n, E_n))$ does not belong to E.

Step 2.3

Definition

New constants. To each pair (e, E) where e is an expression of type τ and E is a possible value of e, we assign a constant $c_{e,E}$ of type τ, different from each other and from all the symbols in the basic set S. The set of these constants is S_c and we define $S' = S \cup S_c$.

Step 2.3

Definition

New constants. To each pair (e, E) where e is an expression of type τ and E is a possible value of e, we assign a constant $c_{e,E}$ of type τ, different from each other and from all the symbols in the basic set S. The set of these constants is S_c and we define $S' = S \cup S_c$.

Definition

If e is an expression over S', then e^* is the expression obtained from e by replacing each constant $c_{e',E}$ with e'.

Step 2.4

Definition

V', the extension of V defined over expressions e over S'.

Step 2.4

Definition

V', *the extension of* V defined over expressions e over S'.

- if e is a constant $c_{e',E}$, then $V'(e) = E$.

Step 2.4

Definition

V', the extension of V defined over expressions e over S'.

- if e is a constant $c_{e',E}$, then $V'(e) = E$.
- if e is not in S_c and is of type 0, then $V'(e) = e^*$.

Step 2.4

Definition

V', *the extension of* V defined over expressions e over S'.

- if e is a constant $c_{e',E}$, then $V'(e) = E$.
- if e is not in S_c and is of type 0, then $V'(e) = e^*$.
- if e is not in S_c and is of type 1, then $V'(e) = t$ if $V(e) = t$ and otherwise $V'(e) = f$.

Step 2.4

Definition

V', the extension of V defined over expressions e over S'.

- if e is a constant $c_{e',E}$, then $V'(e) = E$.
- if e is not in S_c and is of type 0, then $V'(e) = e^*$.
- if e is not in S_c and is of type 1, then $V'(e) = t$ if $V(e) = t$ and otherwise $V'(e) = f$.
- if e is of type $(\tau_1, ..., \tau_n)$, then $V'(e)$ is the set of all n-tuples $((e_1, E_1), ..., (e_n, E_n))$ such that E_i is a possible value of e_i and $V'((e_1, ..., e_n \in e)) = t$.

Step 2.4

Definition

V', *the extension of* V defined over expressions e over S'.

- if e is a constant $c_{e',E}$, then $V'(e) = E$.
- if e is not in S_c and is of type 0, then $V'(e) = e^*$.
- if e is not in S_c and is of type 1, then $V'(e) = t$ if $V(e) = t$ and otherwise $V'(e) = f$.
- if e is of type $(\tau_1, ..., \tau_n)$, then $V'(e)$ is the set of all n-tuples $((e_1, E_1), ..., (e_n, E_n))$ such that E_i is a possible value of e_i and $V'((e_1, ..., e_n \in e)) = t$.
- if e is $(e_1, ..., e_n \in e)$, then $V'(e) = t$ if $((e_1^*, V'(e_1)), ..., (e_n^*, V'(e_n)))$ belongs to $V'(e)$ and $V'(e) = f$ otherwise.

Step 2.4, continued

Definition

- if e is $\neg A$, then $V'(e) = t$ if $V'(A) = f$ and vice versa.

Step 2.4, continued

Definition

- if e is $\neg A$, then $V'(e) = t$ if $V'(A) = f$ and vice versa.
- if e is $A \vee B$, then $V'(e) = t$ if $V'(A) = t$ or $V'(B) = t$ and $V'(e) = f$ otherwise.

Step 2.4, continued

Definition

- if e is $\neg A$, then $V'(e) = t$ if $V'(A) = f$ and vice versa.
- if e is $A \vee B$, then $V'(e) = t$ if $V'(A) = t$ or $V'(B) = t$ and $V'(e) = f$ otherwise.
- if e is $\exists x^\tau A(x^\tau)$, then $V'(e) = t$ if $V'(A(c)) = t$ for some constant c of type τ in S_c and $V'(e) = f$ if $V'(A(c)) = f$ for all such constants.

Step 2.4, continued

Definition

- if e is $\neg A$, then $V'(e) = t$ if $V'(A) = f$ and vice versa.
- if e is $A \vee B$, then $V'(e) = t$ if $V'(A) = t$ or $V'(B) = t$ and $V'(e) = f$ otherwise.
- if e is $\exists x^\tau A(x^\tau)$, then $V'(e) = t$ if $V'(A(c)) = t$ for some constant c of type τ in S_c and $V'(e) = f$ if $V'(A(c)) = f$ for all such constants.
- if e is $\lambda x_1^{\tau_1} ... x_n^{\tau_n} A(x_1^{\tau_1} ... x_n^{\tau_n})$, then $V'(e)$ is the set of all n-tuples $((e_1, E_1), ..., (e_n, E_n))$ such that e_i is an expression of type τ_i over S, E_i is a possible value of e_i and $V'(A(x_1^{\tau_1} ... x_n^{\tau_n})) = t$.

Step 2.5

Theorem

For each expression e over S_1, $V'(e)$ is a possible value of e^*.

Step 2.5

Theorem

For each expression e over S_1, $V'(e)$ is a possible value of e^*.

Proof.

By induction over the length of e. There are several cases to consider.

Step 2.5

Theorem

For each expression e over S_1, $V'(e)$ is a possible value of e^*.

Proof.

By induction over the length of e. There are several cases to consider.

- e is $c_{e',E}$. Then, by definition, E is a possible value of $e' = e^*$ and by definition of V', $V'(e) = E$.

Step 2.5

Theorem

For each expression e over S_1, $V'(e)$ is a possible value of e^*.

Proof.

By induction over the length of e. There are several cases to consider.

- e is $c_{e',E}$. Then, by definition, E is a possible value of $e' = e^*$ and by definition of V', $V'(e) = E$.
- e is of type 0 or a constant or variable in S. Then by definition $V'(e)$ is a possible value of e^*.

Step 2.5

Theorem

For each expression e over S_1, $V'(e)$ is a possible value of e^*.

Proof.

By induction over the length of e. There are several cases to consider.

- e is $c_{e',E}$. Then, by definition, E is a possible value of $e' = e^*$ and by definition of V', $V'(e) = E$.
- e is of type 0 or a constant or variable in S. Then by definition $V'(e)$ is a possible value of e^*.
- e is $(e_1, ..., e_n \in e')$. Assume V assigns some value to e^*. We have $e^* = (e_1^*, ..., e_n^* \in e'^*)$. By the IH, for all i, $V'(e_i)$ is a possible value E_i of e_i^*. In case $V(e^*) = t$, since $V'(e')$ is a possible value of e'^*, it follows that $((e_1^*, E_1), ..., (e_n^*, E_n))$ belongs to $V'(e')$, so $V'(e) = t$. Similar in case $V(e^*) = f$.

$\square$

Step 2.5, continued

- e is $\neg A$. Assume V assigns some value to e^*. We have $e^* = \neg A^*$, so if $V(e^*)$ is t, then $V(A^*) = f$. By IH $V'(A) = f$ and hence $V'(e) = t$. Similar if $V(e^*)$ is f.

Step 2.5, continued

- e is $\neg A$. Assume V assigns some value to e^*. We have $e^* = \neg A^*$, so if $V(e^*)$ is t, then $V(A^*) = f$. By IH $V'(A) = f$ and hence $V'(e) = t$. Similar if $V(e^*)$ is f.
- e is $A \vee B$. Assume V assigns some value to e^*. We have $e^* = A^* \vee B^*$. If $V(e^*) = t$, then $V(A^*) = t$ or $V(B^*) = t$. By IH $V'(A) = t$ or $V'(B) = t$ and hence $V'(e) = t$. Similar if $V(e^*) = f$.

Step 2.5, continued

- e is $\neg A$. Assume V assigns some value to e^*. We have $e^* = \neg A^*$, so if $V(e^*)$ is t, then $V(A^*) = f$. By IH $V'(A) = f$ and hence $V'(e) = t$. Similar if $V(e^*)$ is f.
- e is $A \vee B$. Assume V assigns some value to e^*. We have $e^* = A^* \vee B^*$. If $V(e^*) = t$, then $V(A^*) = t$ or $V(B^*) = t$. By IH $V'(A) = t$ or $V'(B) = t$ and hence $V'(e) = t$. Similar if $V(e^*) = f$.
- e is $\exists x^\tau A(x^\tau)$ and assume V assigns some value to e^*. We have $(\exists x^\tau A(x^\tau))^* = \exists x^\tau (A(x^\tau))^*$. If $V(e^*) = t$, then $V((A(e^\tau))^*) = t$ for some expression e^τ over S. For each possible value E of e^τ, $(A(c_{e^\tau,E}))^* = A^*(e^\tau)$. By IH $V'(A(c_{e^\tau,E})) = t$ for some $c_{e^\tau,E}$ and therefore $V'(e) = t$. Similar if $V(e^*) = f$.

Step 2.5, continued

- e is $\lambda x_1^{\tau_1}...x_n^{\tau_n} A(x_1^{\tau_1}...x_n^{\tau_n})$. By construction, $V'(e)$ is the set of n-tuples $((e_1, E_1), ..., (e_n, E_n))$, where e_i is an expression over S of type τ_i and E_i is a corresponding possible value. Assume V assigns t to $(e_1, ..., e_n \in e^*)$ and $E_1, ..., E_n$ are possible values for $e_1, ..., e_n$. We note that $e^* = \lambda x_1^{\tau_1}...x_n^{\tau_n} A^*(x_1^{\tau_1}...x_n^{\tau_n})$ Now: $V(A^*(e_1, ..., e_n)) = t$. Since $A(c_{e_1,E_1}, ..., c_{e_n,E_n})^* = A^*(e_1, ..., e_2)$, it follows from the IH that $V'(A(c_{e_1,E_1}, ..., c_{e_n,E_n})) = t$ and hence $((e_1, E_1), ..., (e_n, E_n))$ belongs to $V'(e)$.

QED.

Step 2.6

Two useful results.

Lemma

For each expression e over S', there is a constant $c_{e^*,E}$ such that $V'(e) = V'(c_{e^*,E})$.

Step 2.6

Two useful results.

Lemma

For each expression e over S', there is a constant $c_{e^*,E}$ such that $V'(e) = V'(c_{e^*,E})$.

Proof.

$V'(e)$ is a possible value of e^*, so there exists a constant $c_{e^*,V'(e)}$.
Now by definition: $V'(c_{e^*,V'(e)}) = V'(e)$. $\square$

Step 2.6

Two useful results.

Lemma

For each expression e over S', there is a constant $c_{e^*,E}$ such that $V'(e) = V'(c_{e^*,E})$.

Proof.

$V'(e)$ is a possible value of e^*, so there exists a constant $c_{e^*,V'(e)}$.
Now by definition: $V'(c_{e^*,V'(e)}) = V'(e)$. $\square$

Lemma

If e_2 is obtained from e_1 by replacing some occurrences of $c_{e^*,E}$ by e and $V'(e) = V'(c_{e^*,E})$ then $V'(e_1) = V'(e_2)$.

Step 2.6

Two useful results.

Lemma

For each expression e over S', there is a constant $c_{e^*,E}$ such that $V'(e) = V'(c_{e^*,E})$.

Proof.

$V'(e)$ is a possible value of e^*, so there exists a constant $c_{e^*,V'(e)}$.
Now by definition: $V'(c_{e^*,V'(e)}) = V'(e)$. $\square$

Lemma

If e_2 is obtained from e_1 by replacing some occurrences of $c_{e^*,E}$ by e and $V'(e) = V'(c_{e^*,E})$ then $V'(e_1) = V'(e_2)$.

Proof.

By induction on the length of e_1. $\square$

Step 2.6, continued

Theorem

V' is a total valuation

Step 2.6, continued

Theorem

V' is a total valuation

Proof.

V' assigns exactly one truth value to all the formulas over S'. By definition, it satisfies the first five conditions of a semi-valuation. The remaining two conditions need to be checked.

Step 2.6, continued

Theorem

V' is a total valuation

Proof.

V' assigns exactly one truth value to all the formulas over S'. By definition, it satisfies the first five conditions of a semi-valuation. The remaining two conditions need to be checked.

- Suppose V' assigns f to $\exists x^\tau A(x^\tau)$. Let e be an expression of type τ over S'. By definition V' assigns f to each formula $A(c_{e^*,E})$. By a previous lemma, $V'(e) = V(c_{e^*,E})$ for some constant $c_{e^*,E}$. By another previous lemma, $V'(A(c_{e^*,E})) = V(A(e))$ for such a constant, so $V'(A(e)) = f$.

Step 2.6, continued

Theorem

V' is a total valuation

Proof.

V' assigns exactly one truth value to all the formulas over S'. By definition, it satisfies the first five conditions of a semi-valuation. The remaining two conditions need to be checked.

- Suppose V' assigns f to $\exists x^\tau A(x^\tau)$. Let e be an expression of type τ over S'. By definition V' assigns f to each formula $A(c_{e^*,E})$. By a previous lemma, $V'(e) = V(c_{e^*,E})$ for some constant $c_{e^*,E}$. By another previous lemma, $V'(A(c_{e^*,E})) = V(A(e))$ for such a constant, so $V'(A(e)) = f$.
- Suppose V' assigns t to $e_1, ..., e_n \in \lambda x_1, ..., x_n A(x_1, ..., x_n)$. By definition of V', $((e_1^*, V'(e_1)), ..., (e_n^*, V'(e_n)))$ belongs to $V'(\lambda x_1, ..., x_n A(x_1, ..., x_n))$. So V' assigns t to $A(c_{e_1^*,V'(e_1)}, ..., c_{e_n^*,V'(e_n)})$. Because $V'(c_{e_i^*,V'(e_i)}) = V'(e_i)$ and a lemma, $V'(A(e_1, ..., e_n)) = t$. Idem for $V'(e) = f$.

Interlude 2

Earlier we saw that, for a wff F which is not derivable without cut, there exists a semi-valuation V that makes it f. We have proved that a semi-valuation V can be extended to a total valuation V'. Now we need to prove that any wff F which is f in a total valuation, is not derivable. The argument runs as follows:

Interlude 2

Earlier we saw that, for a wff F which is not derivable without cut, there exists a semi-valuation V that makes it f. We have proved that a semi-valuation V can be extended to a total valuation V'. Now we need to prove that any wff F which is f in a total valuation, is not derivable. The argument runs as follows:

- Assume $V(F) = f$ for some total valuation V.

Interlude 2

Earlier we saw that, for a wff F which is not derivable without cut, there exists a semi-valuation V that makes it f. We have proved that a semi-valuation V can be extended to a total valuation V'. Now we need to prove that any wff F which is f in a total valuation, is not derivable. The argument runs as follows:

- Assume $V(F) = f$ for some total valuation V.
- then there is a semi-valuation in which $F \vee E$ is false for a certain E.

Interlude 2

Earlier we saw that, for a wff F which is not derivable without cut, there exists a semi-valuation V that makes it f. We have proved that a semi-valuation V can be extended to a total valuation V'. Now we need to prove that any wff F which is f in a total valuation, is not derivable. The argument runs as follows:

- Assume $V(F) = f$ for some total valuation V.
- then there is a semi-valuation in which $F \vee E$ is false for a certain E.
- then $F \vee E$ is not strictly derivable.

Interlude 2

Earlier we saw that, for a wff F which is not derivable without cut, there exists a semi-valuation V that makes it f. We have proved that a semi-valuation V can be extended to a total valuation V'. Now we need to prove that any wff F which is f in a total valuation, is not derivable. The argument runs as follows:

- Assume $V(F) = f$ for some total valuation V.
- then there is a semi-valuation in which $F \vee E$ is false for a certain E.
- then $F \vee E$ is not strictly derivable.
- then F is not derivable.

Step 3

Definition

$E = \exists x^1 \neg (x^1 \vee \neg x^1)$

Step 3

Definition

$E = \exists x^1 \neg(x^1 \vee \neg x^1)$

$V'(E) = f$ iff V' is total. So, if $V'(F) = f$ for some total valuation V', then there exists a semi-valuation, namely V' itself, such that $V'(F \vee E) = f$.

Step 3.1

Definition

The *reducible parts* of a wff F are its negative parts of the form $(A \vee B)$ and $\exists x A(x)$ and the positive and negative parts of the form $(e_1, ..., e_n \in \lambda x_1, ..., x_n A(x_1, .., x_n))$. i.e. corresponding to conclusions of rules S1, S2, S4a and S4b.

Step 3.1

Definition

The *reducible parts* of a wff F are its negative parts of the form $(A \vee B)$ and $\exists x A(x)$ and the positive and negative parts of the form $(e_1, ..., e_n \in \lambda x_1, ..., x_n A(x_1, .., x_n))$. i.e. corresponding to conclusions of rules S1, S2, S4a and S4b.

Definition

The *rank* of a wff F is the length of the longest subformula string of F.

Step 3.1

Definition

The *reducible parts* of a wff F are its negative parts of the form $(A \vee B)$ and $\exists x A(x)$ and the positive and negative parts of the form $(e_1, ..., e_n \in \lambda x_1, ..., x_n A(x_1, .., x_n))$. i.e. corresponding to conclusions of rules S1, S2, S4a and S4b.

Definition

The *rank* of a wff F is the length of the longest subformula string of F.

Definition

The *reducibility rank* of a wff F is the ordinal number $\omega r + s$, where r is the maximal rank of reducible parts of F and s is the number of reducible parts of F.

Step 3.1, continued

Definition

The *derivability order* is defined inductively:

Step 3.1, continued

Definition

The *derivability order* is defined inductively:

- axioms have derivability order 0.

Step 3.1, continued

Definition

The *derivability order* is defined inductively:

- axioms have derivability order 0.
- if premises of S1 have derivability orders n_1 and n_2, the the conclusion has derivability order $\max(n_1, n_2) + 1$

Step 3.1, continued

Definition

The *derivability order* is defined inductively:

- axioms have derivability order 0.
- if premises of S1 have derivability orders n_1 and n_2, the the conclusion has derivability order $\max(n_1, n_2) + 1$
- if premises of any other basic inference has derivability order n, then the conclusion has derivability order $n + 1$.

Step 3.1, continued

Theorem

If F is strictly derivable, then there exists no semi-valuation in which F is f.

Step 3.1, continued

Theorem

If F is strictly derivable, then there exists no semi-valuation in which F is f.

Let F be a wff of reducibility rank ρ which has derivability order n. We use two IHs:

Step 3.1, continued

Theorem

If F is strictly derivable, then there exists no semi-valuation in which F is f.

Let F be a wff of reducibility rank ρ which has derivability order n. We use two IHs:

- for every wff F, strictly derivable with order $< n$ there is no semi-valuation in which F is f.

Step 3.1, continued

Theorem

If F is strictly derivable, then there exists no semi-valuation in which F is f.

Let F be a wff of reducibility rank ρ which has derivability order n. We use two IHs:

- for every wff F, strictly derivable with order $< n$ there is no semi-valuation in which F is f.
- for every wff F with reducibility rank $< \rho$ which is strictly derivable with order $< n$ there is no semi-valuation in which F is f.

There are the following cases:

Step 3.1, continued

Theorem

If F is strictly derivable, then there exists no semi-valuation in which F is f.

Let F be a wff of reducibility rank ρ which has derivability order n. We use two IHs:

- for every wff F, strictly derivable with order $< n$ there is no semi-valuation in which F is f.
- for every wff F with reducibility rank $< \rho$ which is strictly derivable with order $< n$ there is no semi-valuation in which F is f.

There are the following cases:

- if F is an axiom, i.e. of the form $G(P_+, P_-)$ then it cannot be f in any semi-valuation.

Step 3.1, continued

- if F is of the form $G((A \vee B)_{-})$, then $G(A_{-})$ an $G(B_{-})$ are strictly derivable with order $\leq n$ and reducibility rank $< \rho$. So there is no semi-valuation in which they are f, so there is no semi-valuation in which F is f.

Step 3.1, continued

- if F is of the form $G((A \vee B)_{-})$, then $G(A_{-})$ an $G(B_{-})$ are strictly derivable with order $\leq n$ and reducibility rank $< \rho$. So there is no semi-valuation in which they are f, so there is no semi-valuation in which F is f.
- if F is of the form $G(e_1, ..., e_n \in \lambda x_1, ..., x_n A(x_1, .., x_n))$, then $G(A(e_1, .., e_n))$ is strictly derivable with order $\leq n$ and reducibility rank $< \rho$. So, by IH there is no semi-valuation making $G(A(e_1, .., e_n))$ false, so the same thing holds for F.

Step 3.1, continued

- if F is of the form $G((A \vee B)_-)$, then $G(A_-)$ an $G(B_-)$ are strictly derivable with order $\leq n$ and reducibility rank $< \rho$. So there is no semi-valuation in which they are f, so there is no semi-valuation in which F is f.
- if F is of the form $G(e_1, ..., e_n \in \lambda x_1, ..., x_n A(x_1, .., x_n))$, then $G(A(e_1, .., e_n))$ is strictly derivable with order $\leq n$ and reducibility rank $< \rho$. So, by IH there is no semi-valuation making $G(A(e_1, .., e_n))$ false, so the same thing holds for F.
- if F is of the form $G(\exists x^\tau A(x^\tau)_-)$, then ... (complicated)

Step 3.1, continued

- if F is of the form $G((A \vee B)_{-})$, then $G(A_{-})$ an $G(B_{-})$ are strictly derivable with order $\leq n$ and reducibility rank $< \rho$. So there is no semi-valuation in which they are f, so there is no semi-valuation in which F is f.
- if F is of the form $G(e_1, ..., e_n \in \lambda x_1, ..., x_n A(x_1, .., x_n))$, then $G(A(e_1, .., e_n))$ is strictly derivable with order $\leq n$ and reducibility rank $< \rho$. So, by IH there is no semi-valuation making $G(A(e_1, .., e_n))$ false, so the same thing holds for F.
- if F is of the form $G(\exists x^{\tau} A(x^{\tau})_{-})$, then ... (complicated)
- if F is of the form $G(\exists x^{\tau} A(x^{\tau})_{+})$ and follows from rule S3. Then the premise of that rule is strictly derivable with order $n - 1$. By IH, there is no semi-valuation with F false.

Step 3.2

Theorem

If F is derivable, then $F \vee E$ is strictly derivable.

Step 3.2

Theorem

If F is derivable, then $F \vee E$ is strictly derivable.

Proof.

By induction on order n.

- if F is an axiom, then $F \vee E$ is also an axiom.

Step 3.2

Theorem

If F is derivable, then $F \vee E$ is strictly derivable.

Proof.

By induction on order n.

- if F is an axiom, then $F \vee E$ is also an axiom.
- if F is the conclusion of a basic inference S1, S2 or S4, whose premises F_i are derivable with orders $< n$, then $F_i \vee E$ are strictly derivable by IH, so by applying S1, S2, S4, $F \vee E$ is strictly derivable.

Step 3.2

Theorem

If F is derivable, then $F \vee E$ is strictly derivable.

Proof.

By induction on order n.

- if F is an axiom, then $F \vee E$ is also an axiom.
- if F is the conclusion of a basic inference S1, S2 or S4, whose premises F_i are derivable with orders $< n$, then $F_i \vee E$ are strictly derivable by IH, so by applying S1, S2, S4, $F \vee E$ is strictly derivable.
- if F is the conclusion of a basic inference S3, whose premise $F \vee G$ is derivable with order $n - 1$, then $F \vee E$ is strictly derivable by a short inference.

Step 3.2

Theorem

If F is derivable, then $F \vee E$ is strictly derivable.

Proof.

By induction on order n.

- if F is an axiom, then $F \vee E$ is also an axiom.
- if F is the conclusion of a basic inference S1, S2 or S4, whose premises F_i are derivable with orders $< n$, then $F_i \vee E$ are strictly derivable by IH, so by applying S1, S2, S4, $F \vee E$ is strictly derivable.
- if F is the conclusion of a basic inference S3, whose premise $F \vee G$ is derivable with order $n - 1$, then $F \vee E$ is strictly derivable by a short inference.
- if F is the conclusion of a basic inference S5, whose premise $F \vee E$ is derivable with order $n - 1$, then $F \vee E$ is strictly derivable by a short inference.

Step 3.3

So, we've seen:

Step 3.3

So, we've seen: if $V'(F) = f$ for some total valuation V', then there exists a semi-valuation, such that $V'(F \vee E) = f$.

Step 3.3

So, we've seen: if $V'(F) = f$ for some total valuation V', then there exists a semi-valuation, such that $V'(F \vee E) = f$. and

Step 3.3

So, we've seen: if $V'(F) = f$ for some total valuation V', then there exists a semi-valuation, such that $V'(F \vee E) = f$. and

Theorem

If F is strictly derivable, then there exists no semi-valuation in which F is f.

Step 3.3

So, we've seen: if $V'(F) = f$ for some total valuation V', then there exists a semi-valuation, such that $V'(F \vee E) = f$. and

Theorem

If F is strictly derivable, then there exists no semi-valuation in which F is f.

and

Step 3.3

So, we've seen: if $V'(F) = f$ for some total valuation V', then there exists a semi-valuation, such that $V'(F \vee E) = f$. and

Theorem

If F is strictly derivable, then there exists no semi-valuation in which F is f.

and

Theorem

If F is derivable, then $F \vee E$ is strictly derivable.

Step 3.3

So, we've seen: if $V'(F) = f$ for some total valuation V', then there exists a semi-valuation, such that $V'(F \vee E) = f$. and

Theorem

If F is strictly derivable, then there exists no semi-valuation in which F is f.

and

Theorem

If F is derivable, then $F \vee E$ is strictly derivable.

So,

Step 3.3

So, we've seen: if $V'(F) = f$ for some total valuation V', then there exists a semi-valuation, such that $V'(F \vee E) = f$. and

Theorem

If F is strictly derivable, then there exists no semi-valuation in which F is f.

and

Theorem

If F is derivable, then $F \vee E$ is strictly derivable.

So,

- If $V(F) = f$ for some total valuation V.

Step 3.3

So, we've seen: if $V'(F) = f$ for some total valuation V', then there exists a semi-valuation, such that $V'(F \vee E) = f$. and

Theorem

If F is strictly derivable, then there exists no semi-valuation in which F is f.

and

Theorem

If F is derivable, then $F \vee E$ is strictly derivable.

So,

- If $V(F) = f$ for some total valuation V.
- then there is a semi-valuation in which $F \vee E$ is false.

Step 3.3

So, we've seen: if $V'(F) = f$ for some total valuation V', then there exists a semi-valuation, such that $V'(F \vee E) = f$. and

Theorem

If F is strictly derivable, then there exists no semi-valuation in which F is f.

and

Theorem

If F is derivable, then $F \vee E$ is strictly derivable.

So,

- If $V(F) = f$ for some total valuation V.
- then there is a semi-valuation in which $F \vee E$ is false.
- so, $F \vee E$ is not strictly derivable.

Step 3.3

So, we've seen: if $V'(F) = f$ for some total valuation V', then there exists a semi-valuation, such that $V'(F \vee E) = f$. and

Theorem
If F is strictly derivable, then there exists no semi-valuation in which F is f.

and

Theorem
If F is derivable, then $F \vee E$ is strictly derivable.

So,

- If $V(F) = f$ for some total valuation V.
- then there is a semi-valuation in which $F \vee E$ is false.
- so, $F \vee E$ is not strictly derivable.
- so, F is not derivable.

Step 3.3

So, we've seen: if $V'(F) = f$ for some total valuation V', then there exists a semi-valuation, such that $V'(F \vee E) = f$. and

Theorem

If F is strictly derivable, then there exists no semi-valuation in which F is f.

and

Theorem

If F is derivable, then $F \vee E$ is strictly derivable.

So,

- If $V(F) = f$ for some total valuation V.
- then there is a semi-valuation in which $F \vee E$ is false.
- so, $F \vee E$ is not strictly derivable.
- so, F is not derivable.

And we proved the claim: if $V(F) = f$ for some total valuation V, then F is not derivable.

Summary

Having just proved the last step, we proved all three:

Summary

Having just proved the last step, we proved all three:

1. if A is not derivable without cut then there is a semi-valuation with A false

Summary

Having just proved the last step, we proved all three:

1. if A is not derivable without cut then there is a semi-valuation with A false
2. semi-valuations are extendable to total valuations

Summary

Having just proved the last step, we proved all three:

1. if A is not derivable without cut then there is a semi-valuation with A false
2. semi-valuations are extendable to total valuations
3. if A is false in a total valuation, then A is not derivable

Summary

Having just proved the last step, we proved all three:

1. if A is not derivable without cut then there is a semi-valuation with A false
2. semi-valuations are extendable to total valuations
3. if A is false in a total valuation, then A is not derivable

So, if A is derivable, it is derivable without cut.

References

- Gaisi Takeuti, 1953, "On a generalized logic calculus".
- Dag Prawitz, 1968, "Hauptsatz for higher order logic".
- Motoo Takahashi, 1967, "Proof of cut-elimination in simple type theory".
- Kurt Schütte, 1960, "Syntactical and semantical properties of type theory."
- Thierry Coquand, "Type Theory", The Stanford Encyclopedia of Philosophy (Fall 2018 Edition).