# Arithmetical semantics of first-order logic of proofs

Tatiana Yavorskaya[1] Sergei Artemov[2]

[1]Moscow State University

[2]City University of New York

June 2012, Bern

# Plan

1. Introduction
   - Propositional logic of proofs
   - Quantifiers in the Logic of Proofs
2. First-Order Logic of Proofs FOLP
   - Language
   - Axiomatization
   - Realization of first-order modal and intuitionistic logic
3. Provability interpretation of FOLP
   - Parametric semantics
   - Invariant semantics
   - Generic semantics.

# Propositional logic of proofs LP (S.Artemov, 1995)

$$t{:}F \approx \text{“}t \textit{ is a proof of } F\text{”}$$

Here $t$ is a *proof term*, $F$ is a *formula*.

- Terms are constructed from *proof variables and constants* by functional symbols for operations on proofs, binary $\cdot$, $+$, and unary $!$
- Formulas are the usual ones and those of the form $t{:}F$.

Axioms and rules of LP are those of propositional calculus plus

| | |
|---|---|
| $t{:}(A \to B) \to (s{:}A \to (t \cdot s){:}B)$ | *application* |
| $t{:}A \to (t+s){:}A, \quad s{:}A \to (t+s){:}A$ | *choice* |
| $t{:}A \to\ !t{:}t{:}A$ | *proof checker* |
| $t{:}A \to A$ | *reflexivity* |
| $\vdash c{:}A$, $c$ is a proof constant, $A$ is an axiom of LP | *axiom necessitation* |

# Proof semantics for S4 and Int via LP (S.A., 1995)

## Step 1. Realization of modal and intuitionistic logic

**Theorem.** S4 $\vdash F$ *iff there exists an assignment* $(\cdot)^r$ *("realization") of proof terms to all* $\Box$*'s in* $F$ *such that* LP $\vdash F^r$.

$\Box$ is treated in the style of Skolem as the existential quantifier on proofs:

| | | | | |
|---|---|---|---|---|
| *negative* $\Box$ | $\longmapsto$ | $\forall$ | $\longmapsto$ | *proof variables* |
| *positive* $\Box$ | $\longmapsto$ | $\exists$ | $\longmapsto$ | *terms depending on them* |

## Step 2. Provability semantics in Peano Arithmetic

| | | |
|---|---|---|
| *proof terms* | $\longmapsto$ | *codes of arithmetical derivations* |
| $\cdot, +, !$ | $\longmapsto$ | *total recursive functions on such codes* |
| LP–*formulas* | $\longmapsto$ | *closed arithmetical formulas* |
| $t{:}F$ | $\longmapsto$ | *a proof predicate for* PA |

**Theorem.** LP is sound and complete wrt the above semantics.

# Proof semantics for S4 and Int via LP (S.A., 1995)

## Step 1. Realization of modal and intuitionistic logic

**Theorem.** $\mathsf{S4} \vdash F$ *iff there exists an assignment* $(\cdot)^r$ *("realization") of proof terms to all* $\Box$*'s in* $F$ *such that* $\mathsf{LP} \vdash F^r$.

$\Box$ is treated in the style of Skolem as the existential quantifier on proofs:

| | | | | |
|---|---|---|---|---|
| *negative* $\Box$ | $\longmapsto$ | $\forall$ | $\longmapsto$ | *proof variables* |
| *positive* $\Box$ | $\longmapsto$ | $\exists$ | $\longmapsto$ | *terms depending on them* |

## Step 2. Provability semantics in Peano Arithmetic

| | | |
|---|---|---|
| *proof terms* | $\longmapsto$ | *codes of arithmetical derivations* |
| $\cdot, +, !$ | $\longmapsto$ | *total recursive functions on such codes* |
| LP–*formulas* | $\longmapsto$ | *closed arithmetical formulas* |
| $t{:}F$ | $\longmapsto$ | *a proof predicate for* PA |

**Theorem.** LP is sound and complete wrt the above semantics.

# Proof semantics for S4 and Int via LP (S.A., 1995)

## Step 1. Realization of modal and intuitionistic logic

**Theorem.** $\mathsf{S4} \vdash F$ *iff there exists an assignment* $(\cdot)^r$ *("realization") of proof terms to all* $\Box$*'s in* $F$ *such that* $\mathsf{LP} \vdash F^r$.

$\Box$ is treated in the style of Skolem as the existential quantifier on proofs:

| | | | | |
|---|---|---|---|---|
| *negative* $\Box$ | $\longmapsto$ | $\forall$ | $\longmapsto$ | *proof variables* |
| *positive* $\Box$ | $\longmapsto$ | $\exists$ | $\longmapsto$ | *terms depending on them* |

## Step 2. Provability semantics in Peano Arithmetic

| | | |
|---|---|---|
| *proof terms* | $\longmapsto$ | *codes of arithmetical derivations* |
| $\cdot, +, !$ | $\longmapsto$ | *total recursive functions on such codes* |
| LP–*formulas* | $\longmapsto$ | *closed arithmetical formulas* |
| $t{:}F$ | $\longmapsto$ | *a proof predicate for* PA |

**Theorem.** LP is sound and complete wrt the above semantics.

# Two possible types of quantifiers in the logic of proofs

on proofs (e.g. $\forall x(x{:}F)$) $\longmapsto$ Quantifiers on proofs

- includes propositional provability logic: $\Box F \cong \exists x(x{:}F)$
- includes logic of proofs: $x{:}F \to \exists y(y{:}(x{:}F))$
- is not recursively enumerable (R. Yavorsky, 2001)

conventional ($\exists x(p{:}F(x))$ etc.) $\longmapsto$ Quantifiers on individual variables

- requires an appropriate language: proofs may or may not depend on individual variables;
- even in the operation-free language is not recursively enumerable (S.A.& T.Y. 2001);
- the only known positive result (R. Yavorsky, 2001) does not admit free variables in the scope of the proof predicate; thus, the corresponding logic is not suitable for realization of first-order S4 and Int.

# Two possible types of quantifiers in the logic of proofs

on proofs (e.g. $\forall x(x{:}F)$) $\longmapsto$ Quantifiers on proofs

- includes propositional provability logic: $\Box F \cong \exists x(x{:}F)$
- includes logic of proofs: $x{:}F \to \exists y(y{:}(x{:}F))$
- is not recursively enumerable (R. Yavorsky, 2001)

conventional ($\exists x(p{:}F(x))$ etc.) $\longmapsto$ Quantifiers on individual variables

- requires an appropriate language: proofs may or may not depend on individual variables;
- even in the operation-free language is not recursively enumerable (S.A.& T.Y. 2001);
- the only known positive result (R. Yavorsky, 2001) does not admit free variables in the scope of the proof predicate; thus, the corresponding logic is not suitable for realization of first-order S4 and Int.

# Two possible types of quantifiers in the logic of proofs

on proofs (e.g. $\forall x(x{:}F)$) $\longmapsto$ Quantifiers on proofs

- includes propositional provability logic: $\Box F \cong \exists x(x{:}F)$
- includes logic of proofs: $x{:}F \to \exists y(y{:}(x{:}F))$
- is not recursively enumerable (R. Yavorsky, 2001)

conventional ($\exists x(p{:}F(x))$ etc.) $\longmapsto$ Quantifiers on individual variables

- requires an appropriate language: proofs may or may not depend on individual variables;
- even in the operation-free language is not recursively enumerable (S.A.& T.Y. 2001);
- the only known positive result (R. Yavorsky, 2001) does not admit free variables in the scope of the proof predicate; thus, the corresponding logic is not suitable for realization of first-order S4 and Int.

# Two possible types of quantifiers in the logic of proofs

on proofs (e.g. $\forall x(x{:}F)$) $\longmapsto$ Quantifiers on proofs

- includes propositional provability logic: $\Box F \cong \exists x(x{:}F)$
- includes logic of proofs: $x{:}F \to \exists y(y{:}(x{:}F))$
- is not recursively enumerable (R. Yavorsky, 2001)

conventional ($\exists x(p{:}F(x))$ etc.) $\longmapsto$ Quantifiers on individual variables

- requires an appropriate language: proofs may or may not depend on individual variables;
- even in the operation-free language is not recursively enumerable (S.A.& T.Y. 2001);
- the only known positive result (R. Yavorsky, 2001) does not admit free variables in the scope of the proof predicate; thus, the corresponding logic is not suitable for realization of first-order S4 and Int.

# Towards the language of FOLP.

The standard reading of $\Box A(x)$:

"*given $x = n$, $A(n)$ is provable*" $x$ is *global*

There are two possible readings of $p{:}A(x)$

- "*given $x = n$, $p$ is a proof of $A(n)$*" $x$ is *global*
- "*$p$ is a proof of a formula $A(x)$ with the parameter $x$*" $x$ is *local*

**Example**. Let $p$ be $\{0 = 0\}$, $A(x)$ be $x = 0$.

- if $x$ is global, then $p{:}A(x)$ depends on $x$; it is true iff $x = 0$
- if $x$ is local, then $p{:}A(x)$ does not depend on $x$; it is false

In FOLP we have tools to represent local and global variables:

$t{:}_X A \approx$ "$t$ is a proof of $A$ with the set of global parameters $X$."

# Towards the language of FOLP.

The standard reading of $\Box A(x)$:

*"given $x = n$, $A(n)$ is provable"* — $x$ is *global*

There are two possible readings of $p{:}A(x)$

- *"given $x = n$, $p$ is a proof of $A(n)$"* — $x$ is *global*
- *"$p$ is a proof of a formula $A(x)$ with the parameter $x$"* — $x$ is *local*

**Example**. Let $p$ be $\{0 = 0\}$, $A(x)$ be $x = 0$.

- if $x$ is global, then $p{:}A(x)$ depends on $x$; it is true iff $x = 0$
- if $x$ is local, then $p{:}A(x)$ does not depend on $x$; it is false

In FOLP we have tools to represent local and global variables:

$$t{:}_X A \approx \text{"}t \text{ is a proof of } A \text{ with the set of global parameters } X.\text{"}$$

# Towards the language of FOLP.

The standard reading of $\Box A(x)$:

*"given $x = n$, $A(n)$ is provable"* — $x$ is *global*

There are two possible readings of $p{:}A(x)$

- *"given $x = n$, $p$ is a proof of $A(n)$"* — $x$ is *global*
- *"$p$ is a proof of a formula $A(x)$ with the parameter $x$"* — $x$ is *local*

**Example**. Let $p$ be $\{0 = 0\}$, $A(x)$ be $x = 0$.

- if $x$ is global, then $p{:}A(x)$ depends on $x$; it is true iff $x = 0$
- if $x$ is local, then $p{:}A(x)$ does not depend on $x$; it is false

In FOLP we have tools to represent local and global variables:

$$t{:}_X A \approx \text{"}t \text{ is a proof of } A \text{ with the set of global parameters } X.\text{"}$$

# Towards the language of FOLP.

The standard reading of $\Box A(x)$:

"*given* $x = n$, $A(n)$ *is provable*" $\qquad$ $x$ is *global*

There are two possible readings of $p{:}A(x)$

- "*given* $x = n$, $p$ *is a proof of* $A(n)$" $\qquad$ $x$ is *global*
- "$p$ *is a proof of a formula* $A(x)$ *with the parameter* $x$" $\qquad$ $x$ is *local*

**Example**. Let $p$ be $\{0 = 0\}$, $A(x)$ be $x = 0$.

- if $x$ is global, then $p{:}A(x)$ depends on $x$; it is true iff $x = 0$
- if $x$ is local, then $p{:}A(x)$ does not depend on $x$; it is false

In FOLP we have tools to represent local and global variables:

$$t{:}_X A \approx \text{"}t \text{ is a proof of } A \text{ with the set of global parameters } X\text{."}$$

# Towards the language of FOLP.

The standard reading of $\Box A(x)$:

"*given* $x = n$, $A(n)$ *is provable*" — $x$ is *global*

There are two possible readings of $p{:}A(x)$

- "*given* $x = n$, $p$ *is a proof of* $A(n)$" — $x$ is *global*
- "$p$ *is a proof of a formula* $A(x)$ *with the parameter* $x$" — $x$ is *local*

**Example**. Let $p$ be $\{0 = 0\}$, $A(x)$ be $x = 0$.

- if $x$ is global, then $p{:}A(x)$ depends on $x$; it is true iff $x = 0$
- if $x$ is local, then $p{:}A(x)$ does not depend on $x$; it is false

In FOLP we have tools to represent local and global variables:

$$t{:}_X A \approx \text{"}t \text{ is a proof of } A \text{ with the set of global parameters } X.\text{"}$$

# Towards the language of FOLP.

The standard reading of $\Box A(x)$:

*"given $x = n$, $A(n)$ is provable"* — $x$ is *global*

There are two possible readings of $p{:}A(x)$

- *"given $x = n$, $p$ is a proof of $A(n)$"* — $x$ is *global*
- *"$p$ is a proof of a formula $A(x)$ with the parameter $x$"* — $x$ is *local*

**Example**. Let $p$ be $\{0 = 0\}$, $A(x)$ be $x = 0$.

- if $x$ is global, then $p{:}A(x)$ depends on $x$; it is true iff $x = 0$
- if $x$ is local, then $p{:}A(x)$ does not depend on $x$; it is false

In FOLP we have tools to represent local and global variables:

$$t{:}_X A \approx \text{“}t \text{ is a proof of } A \text{ with the set of global parameters } X.\text{”}$$

# Towards the language of FOLP.

The standard reading of $\Box A(x)$:

*"given $x = n$, $A(n)$ is provable"* — $x$ is *global*

There are two possible readings of $p{:}A(x)$

- *"given $x = n$, $p$ is a proof of $A(n)$"* — $x$ is *global*
- *"$p$ is a proof of a formula $A(x)$ with the parameter $x$"* — $x$ is *local*

**Example**. Let $p$ be $\{0 = 0\}$, $A(x)$ be $x = 0$.

- if $x$ is global, then $p{:}A(x)$ depends on $x$; it is true iff $x = 0$
- if $x$ is local, then $p{:}A(x)$ does not depend on $x$; it is false

In FOLP we have tools to represent local and global variables:

$t{:}_X A \approx$ "$t$ is a proof of $A$ with the set of global parameters $X$."

# The language of FOLP

$t{:}_X A \approx$ "$t$ is a proof of $A$ with the set of global parameters $X$."

All occurrences of variables from $X$ that are free in $A$ are also free in $t{:}_X A$. All other free variables of $A$ are considered local and hence bound in $t{:}_X A$.

**Example.** Let $A(x, y)$ be atomic. Then

| | |
|---|---|
| $p{:}_{\{x\}} A(x, y)$ | $x$ is free (global), $y$ is bound (local) |
| $p{:}_{\{x,y\}} A(x, y)$ | both $x$ and $y$ are free (global) |
| $p{:}_{\emptyset} A(x, y)$ | both $x$ and $y$ are bound (local) |

*Proof terms* are generated from proof variables and constants by functional symbols for operations on proofs: $+$, $\cdot$ and $!$, borrowed from LP and unary $\mathrm{gen}_x$ for each individual variable $x$.

*Formulas* are defined standardly with the only new clause: for a proof term $t$, a finite set of individual variables $X$ and a formula $F$, $t{:}_X F$ is a formula.

# The language of FOLP

$t{:}_X A \approx$ "$t$ is a proof of $A$ with the set of global parameters $X$."

All occurrences of variables from $X$ that are free in $A$ are also free in $t{:}_X A$. All other free variables of $A$ are considered local and hence bound in $t{:}_X A$.

**Example.** Let $A(x, y)$ be atomic. Then

$p{:}_{\{x\}} A(x, y)$ — $x$ is free (global), $y$ is bound (local)
$p{:}_{\{x,y\}} A(x, y)$ — both $x$ and $y$ are free (global)
$p{:}_{\emptyset} A(x, y)$ — both $x$ and $y$ are bound (local)

*Proof terms* are generated from proof variables and constants by functional symbols for operations on proofs: $+$, $\cdot$ and $!$, borrowed from LP and unary $\mathrm{gen}_x$ for each individual variable $x$.

*Formulas* are defined standardly with the only new clause: for a proof term $t$, a finite set of individual variables $X$ and a formula $F$, $t{:}_X F$ is a formula.

# The language of FOLP

$t{:}_X A \approx$ "$t$ is a proof of $A$ with the set of global parameters $X$."

All occurrences of variables from $X$ that are free in $A$ are also free in $t{:}_X A$. All other free variables of $A$ are considered local and hence bound in $t{:}_X A$.

**Example.** Let $A(x,y)$ be atomic. Then

| | |
|---|---|
| $p{:}_{\{x\}}A(x,y)$ | $x$ is free (global), $y$ is bound (local) |
| $p{:}_{\{x,y\}}A(x,y)$ | both $x$ and $y$ are free (global) |
| $p{:}_{\emptyset}A(x,y)$ | both $x$ and $y$ are bound (local) |

*Proof terms* are generated from proof variables and constants by functional symbols for operations on proofs: $+$, $\cdot$ and !, borrowed from LP and unary $\mathrm{gen}_x$ for each individual variable $x$.

*Formulas* are defined standardly with the only new clause: for a proof term $t$, a finite set of individual variables $X$ and a formula $F$, $t{:}_X F$ is a formula.

# The language of FOLP

$t{:}_X A \approx$ "$t$ is a proof of $A$ with the set of global parameters $X$."

All occurrences of variables from $X$ that are free in $A$ are also free in $t{:}_X A$. All other free variables of $A$ are considered local and hence bound in $t{:}_X A$.

**Example.** Let $A(x, y)$ be atomic. Then

$p{:}_{\{x\}} A(x, y)$ — $x$ is free (global), $y$ is bound (local)
$p{:}_{\{x,y\}} A(x, y)$ — both $x$ and $y$ are free (global)
$p{:}_{\emptyset} A(x, y)$ — both $x$ and $y$ are bound (local)

*Proof terms* are generated from proof variables and constants by functional symbols for operations on proofs: $+$, $\cdot$ and $!$, borrowed from LP and unary $\mathrm{gen}_x$ for each individual variable $x$.

*Formulas* are defined standardly with the only new clause: for a proof term $t$, a finite set of individual variables $X$ and a formula $F$, $t{:}_X F$ is a formula.

# The language of FOLP

$t{:}_X A \approx$ "$t$ is a proof of $A$ with the set of global parameters $X$."

All occurrences of variables from $X$ that are free in $A$ are also free in $t{:}_X A$. All other free variables of $A$ are considered local and hence bound in $t{:}_X A$.

**Example.** Let $A(x, y)$ be atomic. Then

$p{:}_{\{x\}} A(x, y)$ — $x$ is free (global), $y$ is bound (local)

$p{:}_{\{x,y\}} A(x, y)$ — both $x$ and $y$ are free (global)

$p{:}_{\emptyset} A(x, y)$ — both $x$ and $y$ are bound (local)

*Proof terms* are generated from proof variables and constants by functional symbols for operations on proofs: $+$, $\cdot$ and !, borrowed from LP and unary $\mathrm{gen}_x$ for each individual variable $x$.

*Formulas* are defined standardly with the only new clause: for a proof term $t$, a finite set of individual variables $X$ and a formula $F$, $t{:}_X F$ is a formula.

# The language of FOLP

$t{:}_X A \approx$ "$t$ is a proof of $A$ with the set of global parameters $X$."

All occurrences of variables from $X$ that are free in $A$ are also free in $t{:}_X A$. All other free variables of $A$ are considered local and hence bound in $t{:}_X A$.

**Example.** Let $A(x, y)$ be atomic. Then

$p{:}_{\{x\}} A(x, y)$ — $x$ is free (global), $y$ is bound (local)

$p{:}_{\{x,y\}} A(x, y)$ — both $x$ and $y$ are free (global)

$p{:}_{\emptyset} A(x, y)$ — both $x$ and $y$ are bound (local)

*Proof terms* are generated from proof variables and constants by functional symbols for operations on proofs: $+$, $\cdot$ and $!$, borrowed from LP and unary $\mathrm{gen}_x$ for each individual variable $x$.

*Formulas* are defined standardly with the only new clause: for a proof term $t$, a finite set of individual variables $X$ and a formula $F$, $t{:}_X F$ is a formula.

# The language of FOLP

$t{:}_X A \approx$ "$t$ is a proof of $A$ with the set of global parameters $X$."

All occurrences of variables from $X$ that are free in $A$ are also free in $t{:}_X A$. All other free variables of $A$ are considered local and hence bound in $t{:}_X A$.

**Example.** Let $A(x, y)$ be atomic. Then

$p{:}_{\{x\}} A(x, y)$ — $x$ is free (global), $y$ is bound (local)

$p{:}_{\{x,y\}} A(x, y)$ — both $x$ and $y$ are free (global)

$p{:}_{\emptyset} A(x, y)$ — both $x$ and $y$ are bound (local)

*Proof terms* are generated from proof variables and constants by functional symbols for operations on proofs: $+$, $\cdot$ and $!$, borrowed from LP and unary $\mathrm{gen}_x$ for each individual variable $x$.

*Formulas* are defined standardly with the only new clause: for a proof term $t$, a finite set of individual variables $X$ and a formula $F$, $t{:}_X F$ is a formula.

# The language of FOLP

$t{:}_X A \approx$ "$t$ is a proof of $A$ with the set of global parameters $X$."

All occurrences of variables from $X$ that are free in $A$ are also free in $t{:}_X A$. All other free variables of $A$ are considered local and hence bound in $t{:}_X A$.

**Example.** Let $A(x, y)$ be atomic. Then

$p{:}_{\{x\}}A(x, y)$ — $x$ is free (global), $y$ is bound (local)
$p{:}_{\{x,y\}}A(x, y)$ — both $x$ and $y$ are free (global)
$p{:}_{\emptyset}A(x, y)$ — both $x$ and $y$ are bound (local)

*Proof terms* are generated from proof variables and constants by functional symbols for operations on proofs: $+$, $\cdot$ and $!$, borrowed from LP and unary $\mathrm{gen}_x$ for each individual variable $x$.

*Formulas* are defined standardly with the only new clause: for a proof term $t$, a finite set of individual variables $X$ and a formula $F$, $t{:}_X F$ is a formula.

# The language of FOLP

$t{:}_X A \approx$ "$t$ is a proof of $A$ with the set of global parameters $X$."

All occurrences of variables from $X$ that are free in $A$ are also free in $t{:}_X A$. All other free variables of $A$ are considered local and hence bound in $t{:}_X A$.

**Example.** Let $A(x, y)$ be atomic. Then

$p{:}_{\{x\}} A(x, y)$ — $x$ is free (global), $y$ is bound (local)
$p{:}_{\{x,y\}} A(x, y)$ — both $x$ and $y$ are free (global)
$p{:}_{\emptyset} A(x, y)$ — both $x$ and $y$ are bound (local)

*Proof terms* are generated from proof variables and constants by functional symbols for operations on proofs: $+$, $\cdot$ and $!$, borrowed from LP and unary $\mathrm{gen}_x$ for each individual variable $x$.

*Formulas* are defined standardly with the only new clause: for a proof term $t$, a finite set of individual variables $X$ and a formula $F$, $t{:}_X F$ is a formula.

# The language of FOLP

$t{:}_X A \approx$ "$t$ is a proof of $A$ with the set of global parameters $X$."

All occurrences of variables from $X$ that are free in $A$ are also free in $t{:}_X A$. All other free variables of $A$ are considered local and hence bound in $t{:}_X A$.

**Example.** Let $A(x, y)$ be atomic. Then

$p{:}_{\{x\}} A(x, y)$ — $x$ is free (global), $y$ is bound (local)

$p{:}_{\{x,y\}} A(x, y)$ — both $x$ and $y$ are free (global)

$p{:}_{\emptyset} A(x, y)$ — both $x$ and $y$ are bound (local)

*Proof terms* are generated from proof variables and constants by functional symbols for operations on proofs: $+$, $\cdot$ and $!$, borrowed from LP and unary $\mathrm{gen}_x$ for each individual variable $x$.

*Formulas* are defined standardly with the only new clause: for a proof term $t$, a finite set of individual variables $X$ and a formula $F$, $t{:}_X F$ is a formula.

# The language of FOLP

$t{:}_X A \approx$ "$t$ is a proof of $A$ with the set of global parameters $X$."

All occurrences of variables from $X$ that are free in $A$ are also free in $t{:}_X A$. All other free variables of $A$ are considered local and hence bound in $t{:}_X A$.

**Example.** Let $A(x, y)$ be atomic. Then

$p{:}_{\{x\}} A(x, y)$ — $x$ is free (global), $y$ is bound (local)
$p{:}_{\{x,y\}} A(x, y)$ — both $x$ and $y$ are free (global)
$p{:}_{\emptyset} A(x, y)$ — both $x$ and $y$ are bound (local)

*Proof terms* are generated from proof variables and constants by functional symbols for operations on proofs: $+$, $\cdot$ and !, borrowed from LP and unary $\text{gen}_x$ for each individual variable $x$.

*Formulas* are defined standardly with the only new clause: for a proof term $t$, a finite set of individual variables $X$ and a formula $F$, $t{:}_X F$ is a formula.

# The language of FOLP

$t{:}_X A \approx$ "$t$ is a proof of $A$ with the set of global parameters $X$."

All occurrences of variables from $X$ that are free in $A$ are also free in $t{:}_X A$. All other free variables of $A$ are considered local and hence bound in $t{:}_X A$.

**Example.** Let $A(x, y)$ be atomic. Then

| | |
|---|---|
| $p{:}_{\{x\}} A(x, y)$ | $x$ is free (global), $y$ is bound (local) |
| $p{:}_{\{x,y\}} A(x, y)$ | both $x$ and $y$ are free (global) |
| $p{:}_{\emptyset} A(x, y)$ | both $x$ and $y$ are bound (local) |

*Proof terms* are generated from proof variables and constants by functional symbols for operations on proofs: $+$, $\cdot$ and !, borrowed from LP and unary $\mathrm{gen}_x$ for each individual variable $x$.

*Formulas* are defined standardly with the only new clause: for a proof term $t$, a finite set of individual variables $X$ and a formula $F$, $t{:}_X F$ is a formula.

# Axioms and rules of FOLP

Axioms:

A1 classical axioms of first-order logic

A2 $t{:}_{Xy}A \to t{:}_X A, \quad y \notin FVar(A)$

A3 $t{:}_X A \to t{:}_{Xy}A$

B1 $t{:}_X A \to A$

B2 $s{:}_X(A \to B) \wedge t{:}_X A \to (s \cdot t){:}_X B$

B3 $t{:}_X A \to (t+s){:}_X A, \quad s{:}_X A \to (t+s){:}_X A$

B4 $t{:}_X A \to\ !t{:}_X t{:}_X A$

B5 $t{:}_X A \to \mathrm{gen}_x(t){:}_X \forall x A,\ x \notin X$

Inference rules:

R1 $\vdash A, A \to B \ \Rightarrow\ \vdash B$ *Modus Ponens*

R2 $\vdash A \ \Rightarrow\ \vdash \forall x A$ *generalization*

R3 $\vdash c{:}A$, where $A$ is an axiom, $c$ is a proof constant *axiom necessitation.*

# An explicit variant of the converse Barcan Formula

The converse Barcan Formula: $\Box\forall xA \to \forall x\Box A$.

In FOLP:

1. $\forall xA \to A$ logical axiom;
2. $c{:}(\forall xA \to A)$ axiom necessitation;
3. $c{:}_{\{x\}}(\forall xA \to A)$ from 2, by axiom A3;
4. $c{:}_{\{x\}}(\forall xA \to A) \to (u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A)$ axiom B2;
5. $u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A$ from 3, 4, by Modus Ponens;
6. $u{:}\forall xA \to u{:}_{\{x\}}\forall xA$ by axiom A3;
7. $u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A$ from 5, 6;
8. $\forall x[u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A]$ from 7, by generalization;
9. $u{:}\forall xA \to \forall x((c\cdot u){:}_{\{x\}}A)$ from 8.

# An explicit variant of the converse Barcan Formula

The converse Barcan Formula: $\Box\forall xA \to \forall x\Box A$.

In FOLP:

1. $\forall xA \to A$ logical axiom;
2. $c{:}(\forall xA \to A)$ axiom necessitation;
3. $c{:}_{\{x\}}(\forall xA \to A)$ from 2, by axiom A3;
4. $c{:}_{\{x\}}(\forall xA \to A) \to (u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A)$ axiom B2;
5. $u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A$ from 3, 4, by Modus Ponens;
6. $u{:}\forall xA \to u{:}_{\{x\}}\forall xA$ by axiom A3;
7. $u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A$ from 5, 6;
8. $\forall x[u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A]$ from 7, by generalization;
9. $u{:}\forall xA \to \forall x((c\cdot u){:}_{\{x\}}A)$ from 8.

# An explicit variant of the converse Barcan Formula

The converse Barcan Formula: $\Box\forall xA \to \forall x\Box A$.

In FOLP:

1. $\forall xA \to A$ — logical axiom;
2. $c{:}(\forall xA \to A)$ — axiom necessitation;
3. $c{:}_{\{x\}}(\forall xA \to A)$ — from 2, by axiom A3;
4. $c{:}_{\{x\}}(\forall xA \to A) \to (u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A)$ — axiom B2;
5. $u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A$ — from 3, 4, by Modus Ponens;
6. $u{:}\forall xA \to u{:}_{\{x\}}\forall xA$ — by axiom A3;
7. $u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A$ — from 5, 6;
8. $\forall x[u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A]$ — from 7, by generalization;
9. $u{:}\forall xA \to \forall x((c\cdot u){:}_{\{x\}}A)$ — from 8.

# An explicit variant of the converse Barcan Formula

The converse Barcan Formula: $\Box\forall x A \to \forall x \Box A$.

In FOLP:

1. $\forall x A \to A$ logical axiom;
2. $c{:}(\forall x A \to A)$ axiom necessitation;
3. $c{:}_{\{x\}}(\forall x A \to A)$ from 2, by axiom A3;
4. $c{:}_{\{x\}}(\forall x A \to A) \to (u{:}_{\{x\}}\forall x A \to (c \cdot u){:}_{\{x\}}A)$ axiom B2;
5. $u{:}_{\{x\}}\forall x A \to (c \cdot u){:}_{\{x\}}A$ from 3, 4, by Modus Ponens;
6. $u{:}\forall x A \to u{:}_{\{x\}}\forall x A$ by axiom A3;
7. $u{:}\forall x A \to (c \cdot u){:}_{\{x\}}A$ from 5, 6;
8. $\forall x[u{:}\forall x A \to (c \cdot u){:}_{\{x\}}A]$ from 7, by generalization;
9. $u{:}\forall x A \to \forall x((c \cdot u){:}_{\{x\}}A)$ from 8.

# An explicit variant of the converse Barcan Formula

The converse Barcan Formula: $\Box\forall x A \to \forall x \Box A$.

In FOLP:

1. $\forall x A \to A$ — logical axiom;
2. $c{:}(\forall x A \to A)$ — axiom necessitation;
3. $c{:}_{\{x\}}(\forall x A \to A)$ — from 2, by axiom A3;
4. $c{:}_{\{x\}}(\forall x A \to A) \to (u{:}_{\{x\}}\forall x A \to (c\cdot u){:}_{\{x\}}A)$ — axiom B2;
5. $u{:}_{\{x\}}\forall x A \to (c\cdot u){:}_{\{x\}}A$ — from 3, 4, by Modus Ponens;
6. $u{:}\forall x A \to u{:}_{\{x\}}\forall x A$ — by axiom A3;
7. $u{:}\forall x A \to (c\cdot u){:}_{\{x\}}A$ — from 5, 6;
8. $\forall x[u{:}\forall x A \to (c\cdot u){:}_{\{x\}}A]$ — from 7, by generalization;
9. $u{:}\forall x A \to \forall x((c\cdot u){:}_{\{x\}}A)$ — from 8.

# An explicit variant of the converse Barcan Formula

The converse Barcan Formula: $\Box\forall xA \rightarrow \forall x\Box A$.

In FOLP:

1. $\forall xA \rightarrow A$ logical axiom;
2. $c{:}(\forall xA \rightarrow A)$ axiom necessitation;
3. $c{:}_{\{x\}}(\forall xA \rightarrow A)$ from 2, by axiom A3;
4. $c{:}_{\{x\}}(\forall xA \rightarrow A) \rightarrow (u{:}_{\{x\}}\forall xA \rightarrow (c\cdot u){:}_{\{x\}}A)$ axiom B2;
5. $u{:}_{\{x\}}\forall xA \rightarrow (c\cdot u){:}_{\{x\}}A$ from 3, 4, by Modus Ponens;
6. $u{:}\forall xA \rightarrow u{:}_{\{x\}}\forall xA$ by axiom A3;
7. $u{:}\forall xA \rightarrow (c\cdot u){:}_{\{x\}}A$ from 5, 6;
8. $\forall x[u{:}\forall xA \rightarrow (c\cdot u){:}_{\{x\}}A]$ from 7, by generalization;
9. $u{:}\forall xA \rightarrow \forall x((c\cdot u){:}_{\{x\}}A)$ from 8.

# An explicit variant of the converse Barcan Formula

The converse Barcan Formula: $\Box\forall xA \to \forall x\Box A$.

In FOLP:

1. $\forall xA \to A$ — logical axiom;
2. $c{:}(\forall xA \to A)$ — axiom necessitation;
3. $c{:}_{\{x\}}(\forall xA \to A)$ — from 2, by axiom A3;
4. $c{:}_{\{x\}}(\forall xA \to A) \to (u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A)$ — axiom B2;
5. $u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A$ — from 3, 4, by Modus Ponens;
6. $u{:}\forall xA \to u{:}_{\{x\}}\forall xA$ — by axiom A3;
7. $u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A$ — from 5, 6;
8. $\forall x[u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A]$ — from 7, by generalization;
9. $u{:}\forall xA \to \forall x((c\cdot u){:}_{\{x\}}A)$ — from 8.

# An explicit variant of the converse Barcan Formula

The converse Barcan Formula: $\Box\forall xA \to \forall x\Box A$.

In FOLP:

1. $\forall xA \to A$ — logical axiom;
2. $c{:}(\forall xA \to A)$ — axiom necessitation;
3. $c{:}_{\{x\}}(\forall xA \to A)$ — from 2, by axiom A3;
4. $c{:}_{\{x\}}(\forall xA \to A) \to (u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A)$ — axiom B2;
5. $u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A$ — from 3, 4, by Modus Ponens;
6. $u{:}\forall xA \to u{:}_{\{x\}}\forall xA$ — by axiom A3;
7. $u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A$ — from 5, 6;
8. $\forall x[u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A]$ — from 7, by generalization;
9. $u{:}\forall xA \to \forall x((c\cdot u){:}_{\{x\}}A)$ — from 8.

# An explicit variant of the converse Barcan Formula

The converse Barcan Formula: $\Box\forall xA \to \forall x\Box A$.

In FOLP:

1. $\forall xA \to A$ logical axiom;
2. $c{:}(\forall xA \to A)$ axiom necessitation;
3. $c{:}_{\{x\}}(\forall xA \to A)$ from 2, by axiom A3;
4. $c{:}_{\{x\}}(\forall xA \to A) \to (u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A)$ axiom B2;
5. $u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A$ from 3, 4, by Modus Ponens;
6. $u{:}\forall xA \to u{:}_{\{x\}}\forall xA$ by axiom A3;
7. $u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A$ from 5, 6;
8. $\forall x[u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A]$ from 7, by generalization;
9. $u{:}\forall xA \to \forall x((c\cdot u){:}_{\{x\}}A)$ from 8.

# An explicit variant of the converse Barcan Formula

The converse Barcan Formula: $\Box\forall xA \to \forall x\Box A$.

In FOLP:

1. $\forall xA \to A$ — logical axiom;
2. $c{:}(\forall xA \to A)$ — axiom necessitation;
3. $c{:}_{\{x\}}(\forall xA \to A)$ — from 2, by axiom A3;
4. $c{:}_{\{x\}}(\forall xA \to A) \to (u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A)$ — axiom B2;
5. $u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A$ — from 3, 4, by Modus Ponens;
6. $u{:}\forall xA \to u{:}_{\{x\}}\forall xA$ — by axiom A3;
7. $u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A$ — from 5, 6;
8. $\forall x[u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A]$ — from 7, by generalization;
9. $u{:}\forall xA \to \forall x((c\cdot u){:}_{\{x\}}A)$ — from 8.

# An explicit variant of the converse Barcan Formula

The converse Barcan Formula: $\Box\forall xA \to \forall x\Box A$.

In FOLP:

1. $\forall xA \to A$ — logical axiom;
2. $c{:}(\forall xA \to A)$ — axiom necessitation;
3. $c{:}_{\{x\}}(\forall xA \to A)$ — from 2, by axiom A3;
4. $c{:}_{\{x\}}(\forall xA \to A) \to (u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A)$ — axiom B2;
5. $u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A$ — from 3, 4, by Modus Ponens;
6. $u{:}\forall xA \to u{:}_{\{x\}}\forall xA$ — by axiom A3;
7. $u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A$ — from 5, 6;
8. $\forall x[u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A]$ — from 7, by generalization;
9. $u{:}\forall xA \to \forall x((c\cdot u){:}_{\{x\}}A)$ — from 8.

# Realization of FOS4

Let $A$ be a first-order modal formula.

**Definition.** *Realization* of $A$ is a FOLP-formula $A^r$ obtained from $A$ by replacing each subformula of the form $\Box B$ by $t{:}_X B$ for some proof term $t$ and $X = FVar(B)$. A realization is *normal* if all negative occurrences of $\Box$ are assigned proof variables.

**Theorem.** If $\mathsf{FOS4} \vdash A$, then there is a normal realization $A^r$ such that $\mathsf{FOLP} \vdash A^r$.

**Example.** The converse Barcan Formula

$$\Box\forall x A \to \forall x \Box A \qquad \text{a modal formula}$$

$$u{:}\,\forall x A \to \forall x((c\cdot u){:}_{\{x\}} A) \qquad \text{its normal realization}$$

# Forgetful projection of FOLP.

In propositional case a forgetful projection $(\cdot)^0$ of an LP-formula to modal language replaces terms by $\Box$, that is, $(p{:}F)^0 = \Box F$.

**Theorem.** (S.A., 1995) $\mathrm{LP}^0 = \mathrm{S4}$.

In first-order case the definition of $(t{:}_X F)^0 = \Box F$ does not work.
**Example.** $(t{:}_{\{x\}} F(x,y))^0 = \Box F(x,y)$ changes the meaning of the formula:

$$\begin{aligned}(p{:}_{\{x\}} F(x,y) \to \mathrm{gen}_y(p){:}_{\{x\}} \forall y F(x,y))^0 &= \\ &= \Box F(x,y) \to \Box\forall y F(x,y)\end{aligned}$$

We have to bind $y$ in $(p{:}_{\{x\}} F(x,y))^0$. A reasonable variant is $\Box\forall y F(x,y)$.

**Definition.** Define $(\cdot)^0$ by induction: $F^0 = F$ for atomic formulas, $(\cdot)^0$ commutes with Boolean connectives and quantifiers, and

$$(t{:}_X F)^0 = \Box\forall y_0 \ldots \forall y_k F^0, \text{ where } \{y_0, \ldots, y_k\} = FVar(F) \setminus X.$$

**Theorem.** $\mathrm{FOLP} \vdash F$ iff $F^0$ is derivable in FOS4.

# Provability interpretation of FOLP: operations on proofs.

Let us fix a natural multi-conclusion Gödel proof predicate

$Proof(x, y) \rightleftharpoons$ *"x is the code of a finite set of tree-like* PA*-derivations, y is the code of a root formula of one of those derivations."*

$[E(\underline{X})]$ denotes a natural arithmetical term for $\lambda K \ulcorner E(K) \urcorner$ where $X$ is a set of individual variables, and $E$ is an arithmetical formula or a derivation.

We skip "[" and "]" whenever it is safe, e.g. for $\lambda n \ulcorner F(n, \ulcorner G(n) \urcorner) \urcorner$

- the full notation is $[F(\underline{x}, [G(\underline{\underline{x}})])]$,
- the simplified notation is $F(\underline{x}, G(\underline{\underline{x}}))$.

**Lemma.** There exist total recursive operations $\cdot$, $+$, $!$, $\text{gen}_x$ satisfying the axioms of FOLP, e.g. the following is provable in PA:

- $Proof(d(\underline{X}), F(\underline{X})) \to Proof(!d(\underline{X}), Proof(d(\underline{\underline{X}}), F(\underline{\underline{X}})))$;

  $t{:}_X A \to\ !t{:}_X t{:}_X A$

- $Proof(d(\underline{X}), F(\underline{X})) \to Proof(\text{gen}_x(d)(\underline{X}), \forall x F(\underline{X}))$, $x \notin X$.

  $t{:}_X A \to \text{gen}_x(t){:}_X \forall x A,\ x \notin X$

# Provability interpretation of FOLP: operations on proofs.

Let us fix a natural multi-conclusion Gödel proof predicate

$Proof(x, y) \rightleftharpoons$ *"x is the code of a finite set of tree-like* PA*-derivations, y is the code of a root formula of one of those derivations."*

$[E(\underline{X})]$ denotes a natural arithmetical term for $\lambda K \ulcorner E(K) \urcorner$ where $X$ is a set of individual variables, and $E$ is an arithmetical formula or a derivation.

We skip "[" and "]" whenever it is safe, e.g. for $\lambda n \ulcorner F(n, \ulcorner G(n) \urcorner) \urcorner$

- the full notation is $[F(\underline{x}, [G(\underline{\underline{x}})])]$,
- the simplified notation is $F(\underline{x}, G(\underline{\underline{x}}))$.

**Lemma.** There exist total recursive operations $\cdot$, $+$, $!$, $\mathrm{gen}_x$ satisfying the axioms of FOLP, e.g. the following is provable in PA:

- $Proof(d(\underline{X}), F(\underline{X})) \to Proof(!d(\underline{X}), Proof(d(\underline{\underline{X}}), F(\underline{\underline{X}})))$;
  $t{:}_X A \to\ !t{:}_X t{:}_X A$
- $Proof(d(\underline{X}), F(\underline{X})) \to Proof(\mathrm{gen}_x(d)(\underline{X}), \forall x F(\underline{X}))$, $x \notin X$.
  $t{:}_X A \to \mathrm{gen}_x(t){:}_X \forall x A$, $x \notin X$

# Parametric interpretation of FOLP.

*A parametric arithmetical interpretation* for FOLP is defined by operations $+$, $\cdot$, $!$, $\text{gen}_x$ as above and an evaluation $*$ that maps

- proof variables and constants to multi-conclusion arithmetical proofs;
- predicate symbols of arity $n$ to arithmetical formulas with $n$ free variables ($*$ commutes with the renaming of individual variables).

The interpretation $*$ of all FOLP-terms and formulas: $*$ commutes with the operations on proofs, Boolean connectives and quantifiers, and

$$(t{:}_X F)^* = Proof(t^*(\underline{X}), F^*(\underline{X})).$$

Each derivation in FOLP generates a *constant specification CS*, i.e., a (finite) set of formulas $c{:}A$ introduced by the axiom necessitation rule. We say that $*$ *respects CS*, if all formulas from $CS$ are true (provable in PA).

**Soundness Theorem.** If FOLP $\vdash F$ with a constant specification $CS$, then PA $\vdash F^*$ for every parametric arithmetical interpretation $*$ respecting $CS$.

# Soundness of the weak reflexivity axiom

**Lemma.** For a given derivation $d$, a formula $F$ and a set of parameters $X$

$$\mathrm{PA} \vdash \mathit{Proof}(d(\underline{X}), F(\underline{X})) \to F(X).$$

**Proof.** Reason in PA.

1. $d(X)$ proves all substitutional instances of formulas proven by the derivation $d$ and only them.

2. If $d(X)$ is a proof of $F(X)$, then $F(X)$ is a substitutional example of $F$ such that $d$ is a proof of $F$.

3. If $d$ is a proof of $F$, then $F$ by the standard parameter-free argument for propositional Logic of Proofs.

# Soundness of the remaining axioms

**Lemma.** For a given derivation $d$, a formula $F$ and a set of parameters $X$

1. $\mathsf{PA} \vdash \mathit{Proof}(d(\underline{Xy}), F(\underline{Xy})) \to \mathit{Proof}(d(\underline{X}), F(\underline{X}))$, $y \notin \mathit{FVar}(F)$;

2. $\mathsf{PA} \vdash \mathit{Proof}(d(\underline{X}), F(\underline{X})) \to \mathit{Proof}(d(\underline{Xy}), F(\underline{Xy}))$.

**Proof.** Reason in PA.

1. If $d$ is a derivation of a formula $F$ and $y$ is not free in $F$, then $y$ is not open in the tree-like proof of $F$ provided by $d$. Hence, removing $y$ from the set of global parameters of $d$ does not change the proof of $F$.

2. A substitution of any number $y$ for an open variable in a derivation $d$ of a formula $F$ is a legitimate derivation $d(y)$ of $F(y)$.

# Invariant parametric semantics of FOLP.

$$\neg u{:}(\neg u{:}\bot)$$

Intuitively, this is not a valid principle, but it holds in the parametric semantics as a result of the specific coding. If $(u{:}\neg u{:}\bot)^*$ were true, then $\ulcorner u^* \urcorner < \ulcorner (\neg u{:}\bot)^* \urcorner < \ulcorner u^* \urcorner$ - a contradiction.

*A proof predicate Prf(x,y)* is a $\Delta_1$-formula with the computable translators $\alpha$ and $\beta$ between *Prf*-proofs and *Proof*-proofs of the same theorems:

$\mathrm{PA} \vdash \forall x, y(Proof(x, y) \leftrightarrow Prf(\alpha(x), y))$, and

$\mathrm{PA} \vdash \forall x, y(Proof(\beta(x), y) \leftrightarrow Prf(x, y))$.

*Invariant interpretation of* FOLP consists of *Prf*, $\alpha$, $\beta$ and

- operations on proofs $\cdot$, $+$, $!$, $\mathrm{gen}_x$ expressed in terms of $\alpha$, $\beta$ and operations for *Proof*; they satisfy the axioms of FOLP
- a mapping $*$.

**Theorem.** If $\mathrm{FOLP} \vdash F$ with the constant specification *CS* then $\mathrm{PA} \vdash F^*$ for each invariant interpretation respecting *CS*.

# Invariant semantics: $\neg u{:}\neg u{:}\bot$ is not valid.

Find $Prf$ and $*$ such that $(u{:}\neg u{:}\bot)^*$ holds.

- Fix an injective numeration of the joint syntax of FOLP and PA.
- Define $Prf$ by a fixed point equation:
  $PA \vdash Prf(x,y) \leftrightarrow Proof(x,y) \vee (x = \ulcorner u \urcorner \wedge y = \ulcorner \neg Prf(\ulcorner u \urcorner, \ulcorner \bot \urcorner) \urcorner)$.
- Define $u^* = \ulcorner u \urcorner$.

One can check that $Prf$ is a proper proof predicate and $(u{:}\neg u{:}\bot)^*$ is provable in PA.

# Invariant semantics: Barcan formula is valid

$\mathsf{PA} \vdash (\forall y(t{:}_{X_y} A) \to t{:}_X A)^*$
for each parametric or invariant interpretation $*$.

**Proof.** For parametric semantics, we reason in PA.

• Let $A(x)$ and $B(x)$ be arithmetical formulas, and suppose for two distinct numerals $n_1$ and $n_2$, $A(n_i)$ syntactically coincides with $B(n_i)$. Then $A(x)$ coincides with $B(x)$.

• Let $p(x)$ be a derivation in PA, and $Q(x)$ an arithmetical formula. If for all $n = 0, 1, 2, \ldots$, $p(n)$ is a derivation for $Q(n)$, then $p$ is a derivation for $Q$ with $x$ as a local variable.

• Reason in PA. Suppose for all $y$, $t^*(X, y)$ is a proof of $A^*(X, y)$. Then $A^*(X, y)$ is in $t^*(X)$ where $y$ is a local variable. Therefore, $t^*(X)$ is a proof of $A^*(X)$.

# Proof functions: definition

A *proof function* is a pair $(p(X), \widehat{p})$, where $p(X)$ is an arithmetical term for a provably total recursive function and $\widehat{p} = (F_1, \ldots, F_n)$ is a finite set of arithmetical formulas (proven by this proof function), such that

1. PA "knows" that for all values of $X$, $p(X)$ proves substitutional examples of formulas from $\widehat{p}$ and only them, that is,
   $$\mathrm{PA} \vdash \mathit{Prf}(p(X), y) \leftrightarrow \bigvee_{i=1}^{n}(y = [F_i(\underline{X})]);$$
2. PA "knows" that each formula proven by $p(X)$ actually holds, i.e., for each formula $F \in \widehat{p}$,
   $$\mathrm{PA} \vdash \mathit{Prf}(p(X), F(\underline{X})) \to F.$$

**Example.** • For a given set of variables $X$ each $\mathit{Prf}$-proof $d$ produces a proof function with $\widehat{p}$ being the set of all formulas proven by $d$.

• There is a provably total recursive function $p(x)$ which for each $n$ finds the proof of $\neg \mathit{Proof}(n, \ulcorner \bot \urcorner)$; it is a proof function.

# Interpretation of proof terms: Proof forms.

*A proof form* $\mathcal{P}$ for a given finite set of variables $Y$ is a family of proof functions $\mathcal{P}_X(X)$, one for each $X \subseteq Y$ such that PA proves

- *Monotonicity*: $\widehat{\mathcal{P}_X} \subseteq \widehat{\mathcal{P}_{Xy}}$.
- *Coherence*: $\widehat{\mathcal{P}_{Xy}} \setminus \widehat{\mathcal{P}_X}$ consists only of formulas in which $y$ occurs free.

**Now we need operations on proof forms.**

**Lemma.** Suppose that $\mathcal{P} = \{p_X \mid X \subseteq Y\}$, $\mathcal{Q} = \{q_X \mid X \subseteq Y\}$ are proof forms. Then one can effectively find proof forms $\mathcal{P} \cdot \mathcal{Q}$, $\mathcal{P} + \mathcal{Q}$, $!\mathcal{P}$, $\mathrm{gen}_x\mathcal{P}$ such that for each $X \subseteq Y$ the following formulas are provable in PA:

- $Prf(\mathcal{P}_X(X), y) \vee Prf(\mathcal{Q}_X(X), y) \rightarrow Prf([\mathcal{P} + \mathcal{Q}]_X(X), y)$;
- $Prf(\mathcal{P}_X(X), y \dot{\rightarrow} z) \wedge Prf(\mathcal{Q}_X(X), y) \rightarrow Prf([\mathcal{P} \cdot \mathcal{Q}]_X(X), z)$;
- $Prf(\mathcal{P}_X(X), y) \rightarrow Prf([!\mathcal{P}]_X(X), Prf(\mathcal{P}_X(\underline{X}), \underline{y}))$;
- $Prf(\mathcal{P}_X(X), y) \rightarrow Prf([\mathrm{gen}_x\mathcal{P}]_X(X), \dot{\forall}_x y)$, $x \notin X$.

# Interpretation of proof terms: Proof forms.

*A proof form* $\mathcal{P}$ for a given finite set of variables $Y$ is a family of proof functions $\mathcal{P}_X(X)$, one for each $X \subseteq Y$ such that PA proves

- *Monotonicity*: $\widehat{\mathcal{P}_X} \subseteq \widehat{\mathcal{P}_{Xy}}$.
- *Coherence*: $\widehat{\mathcal{P}_{Xy}} \setminus \widehat{\mathcal{P}_X}$ consists only of formulas in which $y$ occurs free.

**Now we need operations on proof forms.**

**Lemma.** Suppose that $\mathcal{P} = \{p_X \mid X \subseteq Y\}$, $\mathcal{Q} = \{q_X \mid X \subseteq Y\}$ are proof forms. Then one can effectively find proof forms $\mathcal{P} \cdot \mathcal{Q}$, $\mathcal{P} + \mathcal{Q}$, $!\mathcal{P}$, $\mathrm{gen}_x\mathcal{P}$ such that for each $X \subseteq Y$ the following formulas are provable in PA:

- $Prf(\mathcal{P}_X(X), y) \vee Prf(\mathcal{Q}_X(X), y) \to Prf([\mathcal{P} + \mathcal{Q}]_X(X), y)$;
- $Prf(\mathcal{P}_X(X), y \dot{\to} z) \wedge Prf(\mathcal{Q}_X(X), y) \to Prf([\mathcal{P} \cdot \mathcal{Q}]_X(X), z)$;
- $Prf(\mathcal{P}_X(X), y) \to Prf([!\mathcal{P}]_X(X), Prf(\mathcal{P}_X(\underline{X}), \underline{y}))$;
- $Prf(\mathcal{P}_X(X), y) \to Prf([\mathrm{gen}_x\mathcal{P}]_X(X), \dot{\forall}_x y)$, $x \notin X$.

# Interpretation of proof terms: Proof forms.

*A proof form $\mathcal{P}$* for a given finite set of variables $Y$ is a family of proof functions $\mathcal{P}_X(X)$, one for each $X \subseteq Y$ such that PA proves

- *Monotonicity*: $\widehat{\mathcal{P}_X} \subseteq \widehat{\mathcal{P}_{Xy}}$.
- *Coherence*: $\widehat{\mathcal{P}_{Xy}} \setminus \widehat{\mathcal{P}_X}$ consists only of formulas in which $y$ occurs free.

**Now we need operations on proof forms.**

**Lemma.** Suppose that $\mathcal{P} = \{p_X \mid X \subseteq Y\}$, $\mathcal{Q} = \{q_X \mid X \subseteq Y\}$ are proof forms. Then one can effectively find proof forms $\mathcal{P} \cdot \mathcal{Q}$, $\mathcal{P} + \mathcal{Q}$, $!\mathcal{P}$, $\text{gen}_x\mathcal{P}$ such that for each $X \subseteq Y$ the following formulas are provable in PA:

- $Prf(\mathcal{P}_X(X), y) \vee Prf(\mathcal{Q}_X(X), y) \to Prf([\mathcal{P} + \mathcal{Q}]_X(X), y)$;
- $Prf(\mathcal{P}_X(X), y \dot{\to} z) \wedge Prf(\mathcal{Q}_X(X), y) \to Prf([\mathcal{P} \cdot \mathcal{Q}]_X(X), z)$;
- $Prf(\mathcal{P}_X(X), y) \to Prf([!\mathcal{P}]_X(X), Prf(\mathcal{P}_X(\underline{X}), \underline{y}))$;
- $Prf(\mathcal{P}_X(X), y) \to Prf([\text{gen}_x\mathcal{P}]_X(X), \dot{\forall}_x y)$, $x \notin X$.

# Interpretation of proof terms: Proof forms.

*A proof form $\mathcal{P}$* for a given finite set of variables $Y$ is a family of proof functions $\mathcal{P}_X(X)$, one for each $X \subseteq Y$ such that PA proves

- *Monotonicity*: $\widehat{\mathcal{P}_X} \subseteq \widehat{\mathcal{P}_{Xy}}$.
- *Coherence*: $\widehat{\mathcal{P}_{Xy}} \setminus \widehat{\mathcal{P}_X}$ consists only of formulas in which $y$ occurs free.

**Now we need operations on proof forms.**

**Lemma.** Suppose that $\mathcal{P} = \{p_X \mid X \subseteq Y\}$, $\mathcal{Q} = \{q_X \mid X \subseteq Y\}$ are proof forms. Then one can effectively find proof forms $\mathcal{P} \cdot \mathcal{Q}$, $\mathcal{P} + \mathcal{Q}$, $!\mathcal{P}$, $\text{gen}_x\mathcal{P}$ such that for each $X \subseteq Y$ the following formulas are provable in PA:

- $Prf(\mathcal{P}_X(X), y) \vee Prf(\mathcal{Q}_X(X), y) \to Prf([\mathcal{P} + \mathcal{Q}]_X(X), y)$;
- $Prf(\mathcal{P}_X(X), y \dot{\to} z) \wedge Prf(\mathcal{Q}_X(X), y) \to Prf([\mathcal{P} \cdot \mathcal{Q}]_X(X), z)$;
- $Prf(\mathcal{P}_X(X), y) \to Prf([!\mathcal{P}]_X(X), Prf(\mathcal{P}_X(\underline{X}), \underline{y}))$;
- $Prf(\mathcal{P}_X(X), y) \to Prf([\text{gen}_x\mathcal{P}]_X(X), \dot{\forall}_x y)$, $x \notin X$.

# Generic provability semantics for FOLP.

*A generic arithmetical interpretation* of the language FOLP is

- a proof predicate *Prf*, finite set of variables $Y$, and operations $\{+, \cdot, !, \mathrm{gen}_x\}$ on proof forms which satisfy axioms of FOLP;
- an evaluation $*$ that maps proof variables and constants to proof forms and predicate symbols of arity $n$ to arithmetical formulas with $n$ free variables. We suppose that $*$ commutes with renaming of individual variables.

Interpretation $*$ commutes with *Prf*-operations on proofs, the Boolean connectives, and quantifiers. For proof assertions,

$$(t{:}_X F)^* = Prf(t^*(X), F^*(\underline{X})).$$

**Soundness Theorem.** If FOLP $\vdash F$ with a constant specification $CS$ , then for every generic interpretation $*$ respecting $CS$ , PA $\vdash F^*$.

# Generic provability semantics for FOLP.

*A generic arithmetical interpretation* of the language FOLP is

- a proof predicate $Prf$, finite set of variables $Y$, and operations $\{+, \cdot, !, \mathrm{gen}_x\}$ on proof forms which satisfy axioms of FOLP;
- an evaluation $*$ that maps proof variables and constants to proof forms and predicate symbols of arity $n$ to arithmetical formulas with $n$ free variables. We suppose that $*$ commutes with renaming of individual variables.

Interpretation $*$ commutes with $Prf$-operations on proofs, the Boolean connectives, and quantifiers. For proof assertions,

$$(t{:}_X F)^* = Prf(t^*(X), F^*(\underline{X})).$$

**Soundness Theorem.** If FOLP $\vdash F$ with a constant specification $CS$, then for every generic interpretation $*$ respecting $CS$, PA $\vdash F^*$.

# Generic provability semantics for FOLP.

*A generic arithmetical interpretation* of the language FOLP is

- a proof predicate $Prf$, finite set of variables $Y$, and operations $\{+, \cdot, !, \mathrm{gen}_x\}$ on proof forms which satisfy axioms of FOLP;
- an evaluation $*$ that maps proof variables and constants to proof forms and predicate symbols of arity $n$ to arithmetical formulas with $n$ free variables. We suppose that $*$ commutes with renaming of individual variables.

Interpretation $*$ commutes with $Prf$-operations on proofs, the Boolean connectives, and quantifiers. For proof assertions,

$$(t{:}_X F)^* = Prf(t^*(X), F^*(\underline{X})).$$

**Soundness Theorem.** If FOLP $\vdash F$ with a constant specification $CS$, then for every generic interpretation $*$ respecting $CS$, PA $\vdash F^*$.

# Generic provability semantics for FOLP.

*A generic arithmetical interpretation* of the language FOLP is

- a proof predicate *Prf*, finite set of variables $Y$, and operations $\{+, \cdot, !, \text{gen}_x\}$ on proof forms which satisfy axioms of FOLP;
- an evaluation $*$ that maps proof variables and constants to proof forms and predicate symbols of arity $n$ to arithmetical formulas with $n$ free variables. We suppose that $*$ commutes with renaming of individual variables.

Interpretation $*$ commutes with *Prf*-operations on proofs, the Boolean connectives, and quantifiers. For proof assertions,

$$(t{:}_X F)^* = Prf(t^*(X), F^*(\underline{X})).$$

**Soundness Theorem.** If FOLP $\vdash F$ with a constant specification $CS$ , then for every generic interpretation $*$ respecting $CS$ , PA $\vdash F^*$.

# Barcan formula is not valid in generic semantics

$$\forall x(p{:}_{\{x\}}A(x)) \to f(p){:}\forall x A(x)$$

1. Fix $Y = \{x\}$ and define $A^*(x) = \neg Proof(x, \ulcorner\bot\urcorner)$.
2. There is a provably recursive term (in fact, a proof function) $g(x)$ such that
$$\mathrm{PA} \vdash Proof(g(x), A^*(\underline{x})) \leftrightarrow A^*(x).$$
Define a proof form $\mathcal{G} = \{\mathcal{G}_\emptyset = 0,\ \mathcal{G}_{\{x\}}(x) = g(x)\}$.
3. Put $p^* = \mathcal{G}$, and $u^* = 0$ for all other atomic proof term $u$. Take natural operations on proofs.
4. Under this interpretation $*$, the explicit Barcan formula is false. Indeed, its antecedent, $\forall x Proof(g(x), \neg Proof(\underline{x}, \ulcorner\bot\urcorner))$ is true, whereas its succedent, $Proof(f(p)^*, \ulcorner\forall x A^*\urcorner)$, is false since $\forall x A^*$ is equivalent to the consistency of PA.

# Barcan formula is not valid in generic semantics

$$\forall x(p{:}_{\{x\}}A(x)) \to f(p){:}\forall x A(x)$$

1. Fix $Y = \{x\}$ and define $A^*(x) = \neg Proof(x, \ulcorner\bot\urcorner)$.
2. There is a provably recursive term (in fact, a proof function) $g(x)$ such that

$$\text{PA} \vdash Proof(g(x), A^*(\underline{x})) \leftrightarrow A^*(x).$$

   Define a proof form $\mathcal{G} = \{\mathcal{G}_\emptyset = 0,\ \mathcal{G}_{\{x\}}(x) = g(x)\}$.
3. Put $p^* = \mathcal{G}$, and $u^* = 0$ for all other atomic proof term $u$. Take natural operations on proofs.
4. Under this interpretation $*$, the explicit Barcan formula is false. Indeed, its antecedent, $\forall x Proof(g(x), \neg Proof(\underline{x}, \ulcorner\bot\urcorner))$ is true, whereas its succedent, $Proof(f(p)^*, \ulcorner\forall x A^*\urcorner)$, is false since $\forall x A^*$ is equivalent to the consistency of PA.

# Barcan formula is not valid in generic semantics

$$\forall x(p{:}_{\{x\}}A(x)) \to f(p){:}\forall x A(x)$$

1. Fix $Y = \{x\}$ and define $A^*(x) = \neg Proof(x, \ulcorner\bot\urcorner)$.
2. There is a provably recursive term (in fact, a proof function) $g(x)$ such that

   $$\mathrm{PA} \vdash Proof(g(x), A^*(\underline{x})) \leftrightarrow A^*(x).$$

   Define a proof form $\mathcal{G} = \{\mathcal{G}_\emptyset = 0,\ \mathcal{G}_{\{x\}}(x) = g(x)\}$.
3. Put $p^* = \mathcal{G}$, and $u^* = 0$ for all other atomic proof term $u$. Take natural operations on proofs.
4. Under this interpretation $*$, the explicit Barcan formula is false. Indeed, its antecedent, $\forall x Proof(g(x), \neg Proof(\underline{x}, \ulcorner\bot\urcorner))$ is true, whereas its succedent, $Proof(f(p)^*, \ulcorner\forall x A^*\urcorner)$, is false since $\forall x A^*$ is equivalent to the consistency of PA.

# Barcan formula is not valid in generic semantics

$$\forall x(p{:}_{\{x\}}A(x)) \to f(p){:}\forall x A(x)$$

1. Fix $Y = \{x\}$ and define $A^*(x) = \neg Proof(x, \ulcorner\bot\urcorner)$.
2. There is a provably recursive term (in fact, a proof function) $g(x)$ such that

$$\mathrm{PA} \vdash Proof(g(x), A^*(\underline{x})) \leftrightarrow A^*(x).$$

   Define a proof form $\mathcal{G} = \{\mathcal{G}_\emptyset = 0,\ \mathcal{G}_{\{x\}}(x) = g(x)\}$.
3. Put $p^* = \mathcal{G}$, and $u^* = 0$ for all other atomic proof term $u$. Take natural operations on proofs.
4. Under this interpretation $*$, the explicit Barcan formula is false. Indeed, its antecedent, $\forall x Proof(g(x), \neg Proof(\underline{x}, \ulcorner\bot\urcorner))$ is true, whereas its succedent, $Proof(f(p)^*, \ulcorner\forall x A^*\urcorner)$, is false since $\forall x A^*$ is equivalent to the consistency of PA.

# Barcan formula is not valid in generic semantics

$$\forall x(p{:}_{\{x\}}A(x)) \to f(p){:}\forall x A(x)$$

1. Fix $Y = \{x\}$ and define $A^*(x) = \neg Proof(x, \ulcorner\bot\urcorner)$.
2. There is a provably recursive term (in fact, a proof function) $g(x)$ such that
$$\mathrm{PA} \vdash Proof(g(x), A^*(\underline{x})) \leftrightarrow A^*(x).$$
Define a proof form $\mathcal{G} = \{\mathcal{G}_\emptyset = 0,\ \mathcal{G}_{\{x\}}(x) = g(x)\}$.
3. Put $p^* = \mathcal{G}$, and $u^* = 0$ for all other atomic proof term $u$. Take natural operations on proofs.
4. Under this interpretation $*$, the explicit Barcan formula is false. Indeed, its antecedent, $\forall x Proof(g(x), \neg Proof(\underline{x}, \ulcorner\bot\urcorner))$ is true, whereas its succedent, $Proof(f(p)^*, \ulcorner\forall x A^*\urcorner)$, is false since $\forall x A^*$ is equivalent to the consistency of PA.

# Completeness is not attainable

Consider the languages of FOLP without proof constants and operations on proofs. Let PAR, INV, and GEN be sets of FOLP-formulas valid under the parametric, invariant parametric, and generic semantics correspondingly.

It is clear that

$$\mathrm{FOLP} \subsetneq \mathrm{GEN} \subsetneq \mathrm{INV} \subsetneq \mathrm{PAR}.$$

**Theorem.** Neither of GEN, PAR, or INV is recursively enumerable.

**Proof.**

1. The following set is not recursively enumerable:

$$\mathcal{F} = \{F \mid F \text{ is a first order formula and for all } * \ \mathrm{PA} \nvdash \neg F^*\}.$$

2. This set is reducible to PAR

$$\neg p{:}\neg F \in \mathrm{PAR} \text{ if and only if } F \in \mathcal{F}$$

**Thank you!**