Specifying Peirce's Law in Classical Realizability

Mauricio GUILLERMO & Alexandre MIQUEL

Chambéry 2011

June 16, 2011

Peirce's Law and Classical Realizability

- In 1990 Griffin discovered that `call/cc` could be given the type corresponding to Peirce's Law:

$$((A \Rightarrow B) \Rightarrow A) \Rightarrow A$$

- This discovery gave a direct computational interpretation of classical reasoning (as opposed to negative translations)
- Some classical λ-calculi:
  - Parigot's $\lambda\mu$-calculus.
  - Barbanera & Berardi's Symmetric λ calculus.
  - Curien & Herbelin's $\bar{\lambda}\mu$ calculus.
  - Krivine's λ_c calculus.

Peirce's Law and Classical Realizability

- In 1990 Griffin discovered that `call/cc` could be given the type corresponding to Peirce's Law:

$$((A \Rightarrow B) \Rightarrow A) \Rightarrow A$$

- This discovery gave a direct computational interpretation of classical reasoning (as opposed to negative translations)
- Some classical λ-calculi:
  - Parigot's $\lambda\mu$-calculus.
  - Barbanera & Berardi's Symmetric λ calculus.
  - Curien & Herbelin's $\bar{\lambda}\mu$ calculus.
  - Krivine's λ_c calculus.

Peirce's Law and Classical Realizability

- In 1990 Griffin discovered that `call/cc` could be given the type corresponding to Peirce's Law:

$$((A \Rightarrow B) \Rightarrow A) \Rightarrow A$$

- This discovery gave a direct computational interpretation of classical reasoning (as opposed to negative translations)
- Some classical λ-calculi:
    - Parigot's $\lambda\mu$-calculus.
    - Barbanera & Berardi's Symmetric λ calculus.
    - Curien & Herbelin's $\bar{\lambda}\mu$ calculus.
    - Krivine's λ_c calculus.

The language of classical realizers

The λ_c-calculus

Terms:	$\Lambda)$	t, u	$::=$	$x \mid \lambda xt \mid tu \mid \mathsf{cc} \mid k_\pi$
Stacks:	$\Pi)$	π	$::=$	$\alpha \mid t.\pi$
	$\Lambda \star \Pi)$	p, p'	$::=$	$t \star \pi$

Evaluation rules

(PUSH)	tu	$\star$	π	$\succ$	t	$\star$	$u.\pi$
(GRAB)	λxt	$\star$	$u.\pi$	$\succ$	$t\{x := u\}$	$\star$	π
(SAVE)	cc	$\star$	$t.\pi$	$\succ$	t	$\star$	$k_\pi.\pi$
(RESTORE)	k_π	$\star$	$t.\rho$	$\succ$	t	$\star$	π

The language of classical realizers

The λ_c-calculus

Terms:	$\Lambda)$	t, u	$::=$	$x \mid \lambda x t \mid tu \mid \mathsf{cc} \mid k_\pi$
Stacks:	$\Pi)$	π	$::=$	$\alpha \mid t.\pi$
	$\Lambda \star \Pi)$	p, p'	$::=$	$t \star \pi$

Evaluation rules

(Push)	tu	$\star$	π	$\succ$	t	$\star$	$u.\pi$
(Grab)	$\lambda x t$	$\star$	$u.\pi$	$\succ$	$t\{x := u\}$	$\star$	π
(Save)	cc	$\star$	$t.\pi$	$\succ$	t	$\star$	$k_\pi.\pi$
(Restore)	k_π	$\star$	$t.\rho$	$\succ$	t	$\star$	π

The language of classical realizers

The λ_c-calculus

Terms:	Λ)	t, u	$::=$	$x \mid \lambda xt \mid tu \mid \mathrm{cc} \mid k_\pi$
Stacks:	Π)	π	$::=$	$\alpha \mid t.\pi$
	$\Lambda \star \Pi$)	p, p'	$::=$	$t \star \pi$

Evaluation rules

(Push)	tu	$\star$	π	$\succ$	t	$\star$	$u.\pi$
(Grab)	λxt	$\star$	$u.\pi$	$\succ$	$t\{x := u\}$	$\star$	π
(Save)	cc	$\star$	$t.\pi$	$\succ$	t	$\star$	$k_\pi.\pi$
(Restore)	k_π	$\star$	$t.\rho$	$\succ$	t	$\star$	π

The language of classical realizers

The λ_c-calculus

Terms:	Λ)	t, u	$::=$	$x \mid \lambda x t \mid tu \mid \mathrm{cc} \mid k_\pi$
Stacks:	Π)	π	$::=$	$\alpha \mid t.\pi$
	$\Lambda \star \Pi$)	p, p'	$::=$	$t \star \pi$

Evaluation rules

(Push)	tu	$\star$	π	$\succ$	t	$\star$	$u.\pi$
(Grab)	$\lambda x t$	$\star$	$u.\pi$	$\succ$	$t\{x := u\}$	$\star$	π
(Save)	cc	$\star$	$t.\pi$	$\succ$	t	$\star$	$k_\pi.\pi$
(Restore)	k_π	$\star$	$t.\rho$	$\succ$	t	$\star$	π

The language of classical realizers

The λ_c-calculus

Terms:	$\Lambda)$	t, u	$::=$	$x \mid \lambda x t \mid tu \mid \mathrm{cc} \mid k_\pi$
Stacks:	$\Pi)$	π	$::=$	$\alpha \mid t.\pi$
	$\Lambda \star \Pi)$	p, p'	$::=$	$t \star \pi$

Evaluation rules

$$
\begin{array}{llll}
(\textsc{Push}) & tu \star \pi & \succ & t \star u.\pi \\
(\textsc{Grab}) & \lambda x t \star u.\pi & \succ & t\{x := u\} \star \pi \\
\\
(\textsc{Save}) & \mathrm{cc} \star t.\pi & \succ & t \star k_\pi.\pi \\
(\textsc{Restore}) & k_\pi \star t.\rho & \succ & t \star \pi
\end{array}
$$

The language of classical realizers

The λ_c-calculus

Terms:	$\Lambda)$	t, u	$::=$	$x \mid \lambda xt \mid tu \mid \mathrm{cc} \mid k_\pi$
Stacks:	$\Pi)$	π	$::=$	$\alpha \mid t.\pi$
	$\Lambda \star \Pi)$	p, p'	$::=$	$t \star \pi$

Evaluation rules

(Push)	tu	$\star$	π	$\succ$	t	$\star$	$u.\pi$
(Grab)	λxt	$\star$	$u.\pi$	$\succ$	$t\{x := u\}$	$\star$	π
(Save)	cc	$\star$	$t.\pi$	$\succ$	t	$\star$	$k_\pi.\pi$
(Restore)	k_π	$\star$	$t.\rho$	$\succ$	t	$\star$	π

The language of classical realizers

The λ_c-calculus

$$
\begin{array}{lrcl}
\text{Terms:} & \Lambda) \quad t, u & ::= & x \mid \lambda x t \mid tu \mid \mathtt{cc} \mid k_\pi \\
\text{Stacks:} & \Pi) \quad \pi & ::= & \alpha \mid t.\pi \\
& \Lambda \star \Pi) \quad p, p' & ::= & t \star \pi
\end{array}
$$

Evaluation rules

$$
\begin{array}{lccccccc}
(\text{PUSH}) & tu & \star & \pi & \succ & t & \star & u.\pi \\
(\text{GRAB}) & \lambda x t & \star & u.\pi & \succ & t\{x := u\} & \star & \pi \\
\\
(\text{SAVE}) & \mathtt{cc} & \star & t.\pi & \succ & t & \star & k_\pi.\pi \\
(\text{RESTORE}) & k_\pi & \star & t.\rho & \succ & t & \star & \pi
\end{array}
$$

The language of classical realizers

The λ_c-calculus

Terms:	$\Lambda)$	t, u	$::=$	$x \mid \lambda xt \mid tu \mid \mathrm{cc} \mid k_\pi$
Stacks:	$\Pi)$	π	$::=$	$\alpha \mid t.\pi$
	$\Lambda \star \Pi)$	p, p'	$::=$	$t \star \pi$

Evaluation rules

$$
\begin{array}{llll}
(\textsc{Push}) & tu \star \pi & \succ & t \star u.\pi \\
(\textsc{Grab}) & \lambda xt \star u.\pi & \succ & t\{x := u\} \star \pi \\
 & & & \\
(\textsc{Save}) & \mathrm{cc} \star t.\pi & \succ & t \star k_\pi.\pi \\
(\textsc{Restore}) & k_\pi \star t.\rho & \succ & t \star \pi
\end{array}
$$

The language of classical realizers

The λ_c-calculus

$$\begin{array}{lrrcl}
\text{Terms:} & \Lambda) & t, u & ::= & x \mid \lambda xt \mid tu \mid \mathrm{cc} \mid k_\pi \mid \ldots \\
\text{Stacks:} & \Pi) & \pi & ::= & \alpha \mid t.\pi \\
 & \Lambda \star \Pi) & p, p' & ::= & t \star \pi
\end{array}$$

Evaluation rules

$$\begin{array}{lrclcrcl}
(\textsc{Push}) & tu & \star & \pi & \succ & t & \star & u.\pi \\
(\textsc{Grab}) & \lambda xt & \star & u.\pi & \succ & t\{x := u\} & \star & \pi \\
 & & & & & & & \\
(\textsc{Save}) & \mathrm{cc} & \star & t.\pi & \succ & t & \star & k_\pi.\pi \\
(\textsc{Restore}) & k_\pi & \star & t.\rho & \succ & t & \star & \pi
\end{array}$$

$\vdots$

The language of classical realizers

The λ_c-calculus

Terms:	$\Lambda)$	t, u	$::= \; x \mid \lambda x t \mid tu \mid \mathtt{cc} \mid k_\pi \mid \mathtt{quote}$
Stacks:	$\Pi)$	π	$::= \; \alpha \mid t.\pi$
	$\Lambda \star \Pi)$	p, p'	$::= \; t \star \pi$

Evaluation rules

$$
\begin{array}{lrcl c rcl}
(\textsc{Push}) & tu & \star & \pi & \succ & t & \star & u.\pi \\
(\textsc{Grab}) & \lambda x t & \star & u.\pi & \succ & t\{x := u\} & \star & \pi \\
\\
(\textsc{Save}) & \mathtt{cc} & \star & t.\pi & \succ & t & \star & k_\pi.\pi \\
(\textsc{Restore}) & k_\pi & \star & t.\rho & \succ & t & \star & \pi \\
\\
(\textsc{Quote}) & \mathtt{quote} & \star & t.\pi & \succ & t & \star & \overline{n}_\pi.\pi
\end{array}
$$

The language of classical realizers (cont.)

Some extra instructions

$$(\textsc{Eq}) \quad \mathsf{eq} \star t_1.t_2.u.v.\pi \succ \begin{cases} u \star \pi & \textbf{If } t_1 \equiv t_2 \\ v \star \pi & \textbf{Otherwise} \end{cases}$$

$$(\textsc{Fork}) \quad \pitchfork \star t_1.t_2.\pi \succ \begin{cases} t_1 \star \pi \\ t_2 \star \pi \end{cases}$$

The language of classical realizers (cont.)

Some extra instructions

$$(\textsc{Eq}) \quad \mathsf{eq} \star t_1.t_2.u.v.\pi \succ \begin{cases} u \star \pi & \textbf{If } t_1 \equiv t_2 \\ v \star \pi & \textbf{Otherwise} \end{cases}$$

$$(\textsc{Fork}) \quad \pitchfork \star t_1.t_2.\pi \succ \begin{cases} t_1 \star \pi \\ t_2 \star \pi \end{cases}$$

The language of 2nd order arithmetic (PA2)

1. Language of *first order expressions* and *formulæ*:

$$e \quad ::= \quad x \mid f\ e_1 \ldots e_k$$

$$A, B \quad ::= \quad X\, e_1 \cdots e_k \mid A \Rightarrow B \mid \forall x\, A \mid \forall X\ A$$

2. Language of *parametrical formulæ*:

$$A, B \quad ::= \quad \cdots \mid \dot{F}\, e_1 \cdots e_k$$

for each falsity function $F : \mathbb{N}^k \to \mathcal{P}(\Pi)$

The language of 2nd order arithmetic (PA2)

1. Language of *first order expressions* and *formulæ*:

$$\begin{array}{rcl} e & ::= & x \mid f\ e_1 \ldots e_k \\ A, B & ::= & X\, e_1 \cdots e_k \mid A \Rightarrow B \mid \forall x\, A \mid \forall X\ A \end{array}$$

2. Language of *parametrical formulæ*:

$$A, B \quad ::= \quad \cdots \quad \mid \dot{F}\, e_1 \cdots e_k$$

for each falsity function $F : \mathbb{N}^k \to \mathcal{P}(\Pi)$

Typing rules:

$$\frac{}{\Gamma, x : A \vdash x : A}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x t : A \Rightarrow B} \qquad \frac{\Gamma \vdash t : A \Rightarrow B \quad \Gamma \vdash u : A}{\Gamma \vdash tu : B}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall x A}\; x \notin FV(\Gamma) \qquad \frac{\Gamma \vdash t : \forall x A}{\Gamma \vdash t : A\{x := e\}}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall X A}\; X \notin FV(\Gamma) \qquad \frac{\Gamma \vdash t : \forall X A}{\Gamma \vdash t : A\{X := \lambda x_1 \ldots \lambda x_k P\}}$$

Typing rules:

$$\frac{}{\Gamma, x : A \vdash x : A}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x t : A \Rightarrow B} \qquad \frac{\Gamma \vdash t : A \Rightarrow B \quad \Gamma \vdash u : A}{\Gamma \vdash tu : B}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall x A}\ x \notin FV(\Gamma) \qquad \frac{\Gamma \vdash t : \forall x A}{\Gamma \vdash t : A\{x := e\}}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall X A}\ X \notin FV(\Gamma) \qquad \frac{\Gamma \vdash t : \forall X A}{\Gamma \vdash t : A\{X := \lambda x_1 \ldots \lambda x_k P\}}$$

Typing rules:

$$\overline{\Gamma, x : A \vdash x : A}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x t : A \Rightarrow B} \qquad \frac{\Gamma \vdash t : A \Rightarrow B \quad \Gamma \vdash u : A}{\Gamma \vdash tu : B}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall x A} \; x \notin FV(\Gamma) \qquad \frac{\Gamma \vdash t : \forall x A}{\Gamma \vdash t : A\{x := e\}}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall X A} \; X \notin FV(\Gamma) \qquad \frac{\Gamma \vdash t : \forall X A}{\Gamma \vdash t : A\{X := \lambda x_1 \ldots \lambda x_k P\}}$$

Typing rules:

$$\frac{}{\Gamma, x : A \vdash x : A}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x t : A \Rightarrow B} \qquad \frac{\Gamma \vdash t : A \Rightarrow B \quad \Gamma \vdash u : A}{\Gamma \vdash tu : B}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall x A}\ x \notin FV(\Gamma) \qquad \frac{\Gamma \vdash t : \forall x A}{\Gamma \vdash t : A\{x := e\}}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall X A}\ X \notin FV(\Gamma) \qquad \frac{\Gamma \vdash t : \forall X A}{\Gamma \vdash t : A\{X := \lambda x_1 \ldots \lambda x_k P\}}$$

Typing rules:

$$\frac{}{\Gamma, x : A \vdash x : A}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x t : A \Rightarrow B} \qquad \frac{\Gamma \vdash t : A \Rightarrow B \quad \Gamma \vdash u : A}{\Gamma \vdash tu : B}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall x A}\ x \notin FV(\Gamma) \qquad \frac{\Gamma \vdash t : \forall x A}{\Gamma \vdash t : A\{x := e\}}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall X A}\ X \notin FV(\Gamma) \qquad \frac{\Gamma \vdash t : \forall X A}{\Gamma \vdash t : A\{X := \lambda x_1 \ldots \lambda x_k P\}}$$

(Intuitionistic) Typing rules:

$$\frac{}{\Gamma, x : A \vdash x : A}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x t : A \Rightarrow B} \qquad \frac{\Gamma \vdash t : A \Rightarrow B \quad \Gamma \vdash u : A}{\Gamma \vdash tu : B}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall x A}\ x \notin FV(\Gamma) \qquad \frac{\Gamma \vdash t : \forall x A}{\Gamma \vdash t : A\{x := e\}}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall X A}\ X \notin FV(\Gamma) \qquad \frac{\Gamma \vdash t : \forall X A}{\Gamma \vdash t : A\{X := \lambda x_1 \ldots \lambda x_k P\}}$$

Krivine's Realizability semantics

- Definition parameterized by a *saturated set of processes* $⊥\!\!⊥$.
- This set defines a contravariant function from sets of stacks to sets of terms:

$$S \mapsto S^{⊥\!\!⊥}$$

$$(\_)^{⊥\!\!⊥} : \mathcal{P}(\Pi) \to \mathcal{P}(\Lambda)$$

$$S^{⊥\!\!⊥} := \{t \in \Lambda \mid \forall \pi \in S \; t \star \pi \in ⊥\!\!⊥\}$$

Krivine's Realizability semantics

- Definition parameterized by a *saturated set of processes* $\perp\!\!\!\perp$.
- This set defines a contravariant function from sets of stacks to sets of terms:

$$S \mapsto S^{\perp\!\!\!\perp}$$

$$(\_)^{\perp\!\!\!\perp} : \mathcal{P}(\Pi) \to \mathcal{P}(\Lambda)$$

$$S^{\perp\!\!\!\perp} := \{t \in \Lambda \mid \forall \pi \in S \; t \star \pi \in \perp\!\!\!\perp\}$$

Krivine's Realizability semantics

- Definition parameterized by a *saturated set of processes* $\perp\!\!\!\perp$.
- This set defines a contravariant function from sets of stacks to sets of terms:

$$S \mapsto S^{\perp\!\!\!\perp}$$

$$(\_)^{\perp\!\!\!\perp} : \mathcal{P}(\Pi) \to \mathcal{P}(\Lambda)$$

$$S^{\perp\!\!\!\perp} := \{t \in \Lambda \mid \forall \pi \in S \; t \star \pi \in \perp\!\!\!\perp\}$$

Krivine's Realizability semantics

- Definition parameterized by a *saturated set of processes* $\perp\!\!\!\perp$.
- This set defines a contravariant function from sets of stacks to sets of terms:

$$S \mapsto S^{\perp\!\!\!\perp}$$

$$(\_)^{\perp\!\!\!\perp} : \mathcal{P}(\Pi) \to \mathcal{P}(\Lambda)$$

$$S^{\perp\!\!\!\perp} := \{t \in \Lambda \mid \forall \pi \in S \;\; t \star \pi \in \perp\!\!\!\perp\}$$

Krivine's Realizability semantics

- Definition parameterized by a *saturated set of processes* $\perp\!\!\!\perp$.
- This set defines a contravariant function from sets of stacks to sets of terms:

$$S \mapsto S^{\perp\!\!\!\perp}$$

$$(\_)^{\perp\!\!\!\perp} : \mathcal{P}(\Pi) \to \mathcal{P}(\Lambda)$$

$$S^{\perp\!\!\!\perp} := \{t \in \Lambda \mid \forall \pi \in S \ \ t \star \pi \in \perp\!\!\!\perp\}$$

Krivine's Realizability semantics (cont.)

There are two sets asociated to each *closed parametrical formula* A:

- *Truth value* $|A|$
- *Falsity value* $||A||$

Truth values and falsity values are related by: $|A| = ||A||^{\perp\!\!\!\perp}$.

Definition of the relation $\Vdash$:

$$t \Vdash A \qquad \text{iff} \qquad t \in |A|$$

Krivine's Realizability semantics (cont.)

There are two sets asociated to each *closed parametrical formula* A:

- *Truth value* $|A|$
- *Falsity value* $||A||$

Truth values and falsity values are related by: $|A| = ||A||^{\perp\!\!\!\perp}$.

Definition of the relation $\Vdash$:

$$t \Vdash A \qquad \text{iff} \qquad t \in |A|$$

Krivine's Realizability semantics (cont.)

There are two sets asociated to each *closed parametrical formula* A:

- *Truth value* $|A|$
- *Falsity value* $||A||$

Truth values and falsity values are related by: $|A| = ||A||^{\perp\!\!\!\perp}$.

Definition of the relation $\Vdash$:

$$t \Vdash A \qquad \text{iff} \qquad t \in |A|$$

Krivine's Realizability semantics (cont.)

There are two sets asociated to each *closed parametrical formula* A:

- *Truth value* $|A|$
- *Falsity value* $||A||$

Truth values and falsity values are related by: $|A| = ||A||^{\perp\!\!\!\perp}$.

Definition of the relation $\Vdash$:

$$t \Vdash A \qquad \text{iff} \qquad t \in |A|$$

Krivine's Realizability semantics (cont.)

The falsity values:

- Expressions are interpreted as natural numbers (as in model theory).
- k-ary second-order variables are interpreted as k-ary second-order parameters. Therefore atomic formulæ are interpreted as falsity values.
- First and second order $\forall$ are interpreted as unions:

$$\begin{array}{rcll} \forall x\ A(x) & = & \bigcup_{n\in\mathbb{N}} & \|A(n)\| \\ \forall X\ A(X) & = & \bigcup_{F\in\mathcal{P}(\Pi)^{\mathbb{N}^k}} & \|A(\dot{F})\| \end{array}$$

- For implication we use ortogonality:

$$\|A \Rightarrow B\| = |A|.\|B\|$$

Krivine's Realizability semantics (cont.)

The falsity values:

- Expressions are interpreted as natural numbers (as in model theory).
- k-ary second-order variables are interpreted as k-ary second-order parameters. Therefore atomic formulæ are interpreted as falsity values.
- First and second order $\forall$ are interpreted as unions:

$$\begin{aligned} \forall x\ A(x) &= \bigcup_{n\in\mathbb{N}} \|A(n)\| \\ \forall X\ A(X) &= \bigcup_{F\in\mathcal{P}(\Pi)^{\mathbb{N}^k}} \|A(\dot{F})\| \end{aligned}$$

- For implication we use ortogonality:

$$\|A \Rightarrow B\| = |A|.\|B\|$$

Krivine's Realizability semantics (cont.)

The falsity values:

- Expressions are interpreted as natural numbers (as in model theory).
- k-ary second-order variables are interpreted as k-ary second-order parameters. Therefore atomic formulæ are interpreted as falsity values.
- First and second order $\forall$ are interpreted as unions:

$$\begin{array}{rcll} \forall x\ A(x) & = & \bigcup_{n\in\mathbb{N}} & \|A(n)\| \\ \forall X\ A(X) & = & \bigcup_{F\in\mathcal{P}(\Pi)^{\mathbb{N}^k}} & \|A(\dot{F})\| \end{array}$$

- For implication we use ortogonality:

$$\|A \Rightarrow B\| = |A|.\|B\|$$

Krivine's Realizability semantics (cont.)

The falsity values:

- Expressions are interpreted as natural numbers (as in model theory).
- k-ary second-order variables are interpreted as k-ary second-order parameters. Therefore atomic formulæ are interpreted as falsity values.
- First and second order $\forall$ are interpreted as unions:

$$\begin{array}{rcll} \forall x\ A(x) & = & \bigcup_{n\in\mathbb{N}} & \|A(n)\| \\ \forall X\ A(X) & = & \bigcup_{F\in\mathcal{P}(\Pi)^{\mathbb{N}^k}} & \|A(\dot{F})\| \end{array}$$

- For implication we use ortogonality:

$$\|A \Rightarrow B\| = |A|.\|B\|$$

Krivine's Realizability semantics (cont.)

The falsity values:

- Expressions are interpreted as natural numbers (as in model theory).
- k-ary second-order variables are interpreted as k-ary second-order parameters. Therefore atomic formulæ are interpreted as falsity values.
- First and second order $\forall$ are interpreted as unions:

$$\begin{array}{rcll} \forall x\ A(x) & = & \bigcup_{n\in\mathbb{N}} & \|A(n)\| \\ \forall X\ A(X) & = & \bigcup_{F\in\mathcal{P}(\Pi)^{\mathbb{N}^k}} & \|A(\dot{F})\| \end{array}$$

- For implication we use ortogonality:

$$\|A \Rightarrow B\| = |A|.\|B\|$$

Krivine's Realizability semantics (cont.)

Universal Realizers

A *universal realizer* for a parametrical formula A is a proof-like term t (i.e.: a term without k_π) which realizes the formula A for all $\perp\!\!\!\perp$.

Lemma

Soundness

If $\vdash t : A$ *is provable in the type system* **then** $t \Vdash A$

Krivine's Realizability semantics (cont.)

Universal Realizers

A *universal realizer* for a parametrical formula A is a proof-like term t (i.e.: a term without k_π) which realizes the formula A for all $\perp\!\!\!\perp$.

We write $t \Vdash A$

Lemma

Soundness

If $\vdash t : A$ *is provable in the type system* **then** $t \Vdash A$

Krivine's Realizability semantics (cont.)

Universal Realizers

A *universal realizer* for a parametrical formula A is a proof-like term t (i.e.: a term without k_π) which realizes the formula A for all $\perp\!\!\!\perp$.

We write $t \Vdash A$

Lemma

Soundness

If $\vdash t : A$ *is provable in the type system* **then** $t \Vdash A$

Krivine's Realizability semantics (cont.)

Remark

(LJ)+(Peirce's Law) **iff** (LK)

Proposition

$cc \Vdash \forall X \forall Y ((X \Rightarrow Y) \Rightarrow X) \Rightarrow X$

Consequence

Adding the following typing rule:

$$\frac{}{\Gamma \vdash cc : \forall X \forall Y ((X \Rightarrow Y) \Rightarrow X) \Rightarrow X}$$

the typing system obtained is the *classical second order logic* and it satisfies the *soundness lemma*.

Krivine's Realizability semantics (cont.)

Remark

(LJ)+(Peirce's Law) **iff** (LK)

Proposition

$\mathsf{cc} \Vdash \forall X \forall Y ((X \Rightarrow Y) \Rightarrow X) \Rightarrow X$

Consequence

Adding the following typing rule:

$$\frac{}{\Gamma \vdash \mathsf{cc} : \forall X \forall Y ((X \Rightarrow Y) \Rightarrow X) \Rightarrow X}$$

the typing system obtained is the *classical second order logic* and it satisfies the *soundness lemma*.

Krivine's Realizability semantics (cont.)

Remark

(LJ)+(Peirce's Law) **iff** (LK)

Proposition

$\mathrm{cc} \Vdash \forall X \forall Y((X \Rightarrow Y) \Rightarrow X) \Rightarrow X$

Consequence

Adding the following typing rule:

$$\frac{}{\Gamma \vdash \mathrm{cc} : \forall X \forall Y((X \Rightarrow Y) \Rightarrow X) \Rightarrow X}$$

the typing system obtained is the *classical second order logic* and it satisfies the *soundness lemma*.

Remark

$$||\forall X \forall Y((X \Rightarrow Y) \Rightarrow X) \Rightarrow X|| = ||\forall X((X \Rightarrow \bot) \Rightarrow X) \Rightarrow X||$$

The Specification Problem

Main question

"Can we characterize the universal realizers of a given formula A from their computational behavior?"

Such a computational behavior is called *the Specification of A*.

Why (and where) the Specification problem is interesting?

In Intuitionistic Realizability, we can infer such a specification from the Realizability definition...

A familiar example

Consider a Σ^0_1-formula $\exists x\ (f(x) = 0)$

Why (and where) the Specification problem is interesting?

In Intuitionistic Realizability, we can infer such a specification from the Realizability definition...

A familiar example

Consider a Σ^0_1-formula $\exists x\ (f(x) = 0)$

Intuitionistic

$t \Vdash \exists^{\mathrm{Nat}} x (f(x) = 0)$ **iff** $t \succ \langle n, \mathbf{I} \rangle$
where $\mathbb{N} \models f(n) = 0$

Why (and where) the Specification problem is interesting?

In Intuitionistic Realizability, we can infer such a specification from the Realizability definition...

A familiar example

Consider a Σ^0_1-formula $\exists x\,(f(x) = 0)$

Intuitionistic

$t \Vdash \exists^{\mathrm{Nat}} x (f(x) = 0)$ **iff** $t \succ \langle n, \mathbf{I} \rangle$ where $\mathbb{N} \models f(n) = 0$

Classical

$t \Vvdash \exists^{\mathrm{Nat}} x (f(x) = 0)$ **iff** t computes a *winning strategy* for a game with backtracking.

The identity Type

Theorem

The following statements are equivalent:

1. $t \Vdash \forall X(X \Rightarrow X)$
2. *For all* $u \in \Lambda$ *and* $\pi \in \Pi \quad t \star u.\pi \succ u \star \pi$

The identity Type

Theorem

The following statements are equivalent:

1. $t \Vdash \forall X(X \Rightarrow X)$
2. *For all* $u \in \Lambda$ *and* $\pi \in \Pi$ $\quad t \star u.\pi \succ u \star \pi$

Proof.

$\square$

The identity Type

Theorem

The following statements are equivalent:

1. $t \Vdash \forall X(X \Rightarrow X)$
2. *For all* $u \in \Lambda$ *and* $\pi \in \Pi \quad t \star u.\pi \succ u \star \pi$

Proof.

$\Uparrow$ Consider $\mathbb{X} \subseteq \Pi$, a term $u \Vdash \mathbb{X}$ and a stack $\pi \in \mathbb{X}$. Since $u \star \pi \in \perp\!\!\!\perp$, by antievaluation $t \star u.\pi \in \perp\!\!\!\perp$.

□

The identity Type

Theorem

The following statements are equivalent:

1. $t \Vdash \forall X(X \Rightarrow X)$
2. *For all* $u \in \Lambda$ *and* $\pi \in \Pi$ $\quad t \star u.\pi \succ u \star \pi$

Proof.

$\Uparrow$ Consider $\mathbb{X} \subseteq \Pi$, a term $u \Vdash \mathbb{X}$ and a stack $\pi \in \mathbb{X}$. Since $u \star \pi \in \perp\!\!\!\perp$, by antievaluation $t \star u.\pi \in \perp\!\!\!\perp$.
$\Downarrow$ Consider $\perp\!\!\!\perp := \{p \mid p \succ u \star \pi\}$. Hence $u \Vdash \{\pi\}$ and we have the result because $t \Vdash \{\pi\} \Rightarrow \{\pi\}$. □

Interaction Constants (def.)

Definition

A constant K is:

Interaction Constants (def.)

Definition

A constant K is:

- *inert* **iff** for all stacks π $\quad K \star \pi \not\succ$

Interaction Constants (def.)

Definition

A constant K is:

- *inert* **iff** for all stacks π $\quad K \star \pi \not\succ$
- *substitutive* **iff** for all processes p, p' and for all terms u, $p \succ_1 p'$ implies $p\{K := u\} \succ_1 p'\{K := u\}$.

Interaction Constants (def.)

Definition

A constant K is:

- *inert* **iff** for all stacks π $\quad K \star \pi \not\succ$
- *substitutive* **iff** for all processes p, p' and for all terms u, $p \succ_1 p'$ implies $p\{K := u\} \succ_1 p'\{K := u\}$.
- *non generative* **iff** whenever $p \succ_1 p'$, the constant K does not occur in p' unless it already occurs in p.

Interaction Constants (def.)

Definition

A constant K is:

- *inert* **iff** for all stacks π $\quad K \star \pi \not\succ$
- *substitutive* **iff** for all processes p, p' and for all terms u, $p \succ_1 p'$ implies $p\{K := u\} \succ_1 p'\{K := u\}$.
- *non generative* **iff** whenever $p \succ_1 p'$, the constant K does not occur in p' unless it already occurs in p.

An *interaction constant* is an inert, substitutive and non generative constant.

Interaction Constants (def.)

Definition

A constant K is:

- *inert* **iff** for all stacks π $\quad K \star \pi \not\succ$
- *substitutive* **iff** for all processes p, p' and for all terms u, $p \succ_1 p'$ implies $p\{K := u\} \succ_1 p'\{K := u\}$.
- *non generative* **iff** whenever $p \succ_1 p'$, the constant K does not occur in p' unless it already occurs in p.

An *interaction constant* is an inert, substitutive and non generative constant.
Similar definitions are given for *substitutive* and *non generative* stack constants.

Remark

- Substitutive constants are compatible with the *basic rules*

$$(\text{PUSH}),\ (\text{GRAB}),\ (\text{SAVE}),\ (\text{RESTORE})$$

- On the other hand, substitutive constants are incompatible with the rules

$$(\text{QUOTE}),\ (\text{EQ})$$

Remark

- Substitutive constants are compatible with the *basic rules*

$$(\text{PUSH}),\ (\text{GRAB}),\ (\text{SAVE}),\ (\text{RESTORE})$$

- On the other hand, substitutive constants are incompatible with the rules

$$(\text{QUOTE}),\ (\text{EQ})$$

eq is incompatible with substitutive constants

A process containing eq and an inert constant K

$$\mathsf{eq} \star K \,.\, \mathbf{I} \,.\, \delta\delta\mathbf{0} \,.\, \delta\delta\mathbf{1} \,.\, \pi \;\succ\; \delta \star \delta \,.\, \mathbf{1} \,.\, \pi$$

Applying $\{K := \mathbf{I}\}$, we obtain

$$\mathsf{eq} \star \mathbf{I} \,.\, \mathbf{I} \,.\, \delta\delta\mathbf{0} \,.\, \delta\delta\mathbf{1} \,.\, \pi \;\succ\; \delta \star \delta \,.\, \mathbf{0} \,.\, \pi$$

eq is incompatible with substitutive constants

A process containing eq and an inert constant K

$\mathsf{eq} \star K \,.\, \mathbf{I} \,.\, \delta\delta\mathbf{0} \,.\, \delta\delta\mathbf{1} \,.\, \pi \;\succ\; \delta \star \delta \,.\, \mathbf{1} \,.\, \pi$

Applying $\{K := \mathbf{I}\}$, we obtain

$\mathsf{eq} \star \mathbf{I} \,.\, \mathbf{I} \,.\, \delta\delta\mathbf{0} \,.\, \delta\delta\mathbf{1} \,.\, \pi \;\succ\; \delta \star \delta \,.\, \mathbf{0} \,.\, \pi$

Consequence

quote is also incompatible with substitutive constants since eq can be programed in terms of quote.

The realizers $C_{n,p}$

The realizers $C_{n,p}$

Consider n, p such that $n \geq p \geq 1$ and a λ-term $C_{n,p}$ with the following behavior:

The realizers $C_{n,p}$

Consider n, p such that $n \geq p \geq 1$ and a λ-term $C_{n,p}$ with the following behavior:

Given a stack $u_0.\pi_0$:

The realizers $C_{n,p}$

Consider n, p such that $n \geq p \geq 1$ and a λ-term $C_{n,p}$ with the following behavior:

Given a stack $u_0.\pi_0$:

$$C_{n,p} \star u_0.\pi_0 \succ u_0 \star K^1_{n,p}[u_0, \pi_0].\pi_0$$

The realizers $C_{n,p}$

Consider n, p such that $n \geq p \geq 1$ and a λ-term $C_{n,p}$ with the following behavior:

Given a stack $u_0.\pi_0$:

$$C_{n,p} \quad \star \quad u_0.\pi_0 \quad \succ \quad u_0 \quad \star \quad K^1_{n,p}[u_0, \pi_0].\pi_0$$

Where for all stack $u_1.\pi_1$, $K^1_{n,p}$ satisfies:

The realizers $C_{n,p}$

Consider n, p such that $n \geq p \geq 1$ and a λ-term $C_{n,p}$ with the following behavior:

Given a stack $u_0.\pi_0$:

$$C_{n,p} \quad \star \quad u_0.\pi_0 \quad \succ \quad u_0 \quad \star \quad K^1_{n,p}[u_0, \pi_0].\pi_0$$

Where for all stack $u_1.\pi_1$, $K^1_{n,p}$ satisfies:

$$K^1_{n,p} \quad \star \quad u_1.\pi_1 \quad \succ \quad u_0 \quad \star \quad K^2_{n,p}[\ u_0, \pi_0, u_1, \pi_1].\pi_0$$

The realizers $C_{n,p}$

Consider n, p such that $n \geq p \geq 1$ and a λ-term $C_{n,p}$ with the following behavior:

Given a stack $u_0.\pi_0$:

$$C_{n,p} \quad \star \quad u_0.\pi_0 \quad \succ \quad u_0 \quad \star \quad K^1_{n,p}[u_0, \pi_0].\pi_0$$

Where for all stack $u_1.\pi_1$, $K^1_{n,p}$ satisfies:

$$K^1_{n,p} \quad \star \quad u_1.\pi_1 \quad \succ \quad u_0 \quad \star \quad K^2_{n,p}[\, u_0, \pi_0, u_1, \pi_1].\pi_0$$

$$\vdots$$

The realizers $C_{n,p}$

Consider n, p such that $n \geq p \geq 1$ and a λ-term $C_{n,p}$ with the following behavior:

Given a stack $u_0.\pi_0$:

$$C_{n,p} \quad \star \quad u_0.\pi_0 \quad \succ \quad u_0 \quad \star \quad K^1_{n,p}[u_0, \pi_0].\pi_0$$

Where for all stack $u_1.\pi_1$, $K^1_{n,p}$ satisfies:

$$K^1_{n,p} \quad \star \quad u_1.\pi_1 \quad \succ \quad u_0 \quad \star \quad K^2_{n,p}[\, u_0, \pi_0, u_1, \pi_1].\pi_0$$

$$\vdots$$

Where for all stack $u_n.\pi_n$, $K^n_{n,p}$ satisfies:

The realizers $C_{n,p}$

Consider n, p such that $n \geq p \geq 1$ and a λ-term $C_{n,p}$ with the following behavior:

Given a stack $u_0.\pi_0$:

$$C_{n,p} \quad \star \quad u_0.\pi_0 \quad \succ \quad u_0 \quad \star \quad K^1_{n,p}[u_0,\pi_0].\pi_0$$

Where for all stack $u_1.\pi_1$, $K^1_{n,p}$ satisfies:

$$K^1_{n,p} \quad \star \quad u_1.\pi_1 \quad \succ \quad u_0 \quad \star \quad K^2_{n,p}[\ u_0,\pi_0,u_1,\pi_1].\pi_0$$

$$\vdots$$

Where for all stack $u_n.\pi_n$, $K^n_{n,p}$ satisfies:

$$K^n_{n,p} \quad \star \quad u_n.\pi_n \quad \succ \quad u_p \quad \star \quad \pi_0$$

The realizers $C_{n,p}$

Consider n, p such that $n \geq p \geq 1$ and a λ-term $C_{n,p}$ with the following behavior:

Given a stack $u_0.\pi_0$:

$$C_{n,p} \star u_0.\pi_0 \succ u_0 \star K^1_{n,p}[u_0, \pi_0].\pi_0$$

Where for all stack $u_1.\pi_1$, $K^1_{n,p}$ satisfies:

$$K^1_{n,p} \star u_1.\pi_1 \succ u_0 \star K^2_{n,p}[u_0, \pi_0, u_1, \pi_1].\pi_0$$

$$\vdots$$

Where for all stack $u_n.\pi_n$, $K^n_{n,p}$ satisfies:

$$K^n_{n,p} \star u_n.\pi_n \succ u_p \star \pi_0$$

$K^n_{n,p}$ puts the p-th u_i in head position and restores the initial stack π_0.

The realizers $C_{n,p}$ (cont.)

Proposition

The terms $C_{n,p}$ are universal realizers of Peirce's Law.

The realizers $C_{n,p}$ (cont.)

Proposition

The terms $C_{n,p}$ are universal realizers of Peirce's Law.

Question:

Are there other universal realizers for Peirce's Law?

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$	$\forall$

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$	$\forall$
	t_0	

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$	$\forall$
	t_0	$u_0 . \pi_0$

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$	$\forall$
(initialisation phase) $\emptyset$	t_0	$u_0 . \pi_0$

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$

The game $\mathbb{G}_0$

$$\ell \text{ (}\forall\text{ played moves)} \quad \Big| \quad \exists \qquad \forall$$

$$\text{(initialisation phase)} \quad \emptyset \quad \Big| \quad t_0 \star u_0 . \pi_0 \succ$$

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$			
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0 . \pi_0$	$\succ$	u_0	π_0

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$				
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$	$\succ$	u_0	$\star$	$t_1.\ \pi_0$

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$				
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0 . \pi_0$	$\succ$	u_0	$\star$	$t_1 . \pi_0$
	t_1						

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$				
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$	$\succ$	u_0	$\star$	$t_1.\ \pi_0$
	t_1		$u_1.\pi_1$				

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$				
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$	$\succ$	u_0	$\star$	$t_1.\ \pi_0$
$\{u_1.\pi_1\}$	t_1		$u_1.\pi_1$				

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$				
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$	$\succ$	u_0	$\star$	$t_1.\ \pi_0$
$\{u_1.\pi_1\}$	t_1	$\star$	$u_1.\pi_1$				

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$					
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$	$\succ$	u_0	$\star$	$t_1.$	π_0
$\{u_1.\pi_1\}$	t_1	$\star$	$u_1.\pi_1$	$\succ$				

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$					
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$	$\succ$	u_0	$\star$	$t_1.$	π_0
$\{u_1.\pi_1\}$	t_1	$\star$	$u_1.\pi_1$	$\succ$	u_0			π_0

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$				
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$	$\succ$	u_0	$\star$	$t_1 . \pi_0$
$\{u_1.\pi_1\}$	t_1	$\star$	$u_1.\pi_1$	$\succ$	u_0	$\star$	$t_2 . \pi_0$
$\vdots$				$\vdots$			

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$					
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$	$\succ$	u_0	$\star$	$t_1.$	π_0
$\{u_1.\pi_1\}$	t_1	$\star$	$u_1.\pi_1$	$\succ$	u_0	$\star$	$t_2.$	π_0
$\vdots$				$\vdots$				
$\{u_1.\pi_1, \ldots,$	t_n							

The game $\mathbb{G}_0$

ℓ (∀ played moves)	∃		∀				
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$	$\succ$	u_0	$\star$	$t_1.\ \pi_0$
$\{u_1.\pi_1\}$	t_1	$\star$	$u_1.\pi_1$	$\succ$	u_0	$\star$	$t_2.\ \pi_0$
$\vdots$				$\vdots$			
$\{u_1.\pi_1, \ldots,$	t_n		$u_n.\pi_n$				

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$				
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$	$\succ$	u_0	$\star$	$t_1.\ \pi_0$
$\{u_1.\pi_1\}$	t_1	$\star$	$u_1.\pi_1$	$\succ$	u_0	$\star$	$t_2.\ \pi_0$
$\vdots$				$\vdots$			
$\{u_1.\pi_1, \ldots, u_n.\pi_n\}$	t_n		$u_n.\pi_n$				

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$				
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$	$\succ$	u_0	$\star$	$t_1.\ \pi_0$
$\{u_1.\pi_1\}$	t_1	$\star$	$u_1.\pi_1$	$\succ$	u_0	$\star$	$t_2.\ \pi_0$
$\vdots$				$\vdots$			
$\{u_1.\pi_1, \ldots, u_n.\pi_n\}$	t_n	$\star$	$u_n.\pi_n$				

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$				
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$	$\succ$	u_0	$\star$	$t_1.\ \pi_0$
$\{u_1.\pi_1\}$	t_1	$\star$	$u_1.\pi_1$	$\succ$	u_0	$\star$	$t_2.\ \pi_0$
$\vdots$				$\vdots$			
$\{u_1.\pi_1, \ldots, u_n.\pi_n\}$	t_n	$\star$	$u_n.\pi_n$	$\succ$			

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$				
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$	$\succ$	u_0	$\star$	$t_1.\ \pi_0$
$\{u_1.\pi_1\}$	t_1	$\star$	$u_1.\pi_1$	$\succ$	u_0	$\star$	$t_2.\ \pi_0$
$\vdots$				$\vdots$			
$\{u_1.\pi_1, \ldots, u_n.\pi_n\}$	t_n	$\star$	$u_n.\pi_n$	$\succ$			

$\exists$ wins **iff** at any time, for some $u_p.\pi_p$ previous move, the process $u_p \star \pi_0$ arrives on execution.

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$				
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$	$\succ$	u_0	$\star$	$t_1.\ \pi_0$
$\{u_1.\pi_1\}$	t_1	$\star$	$u_1.\pi_1$	$\succ$	u_0	$\star$	$t_2.\ \pi_0$
$\vdots$				$\vdots$			
$\{u_1.\pi_1, \ldots, u_n.\pi_n\}$	t_n	$\star$	$u_n.\pi_n$	$\succ$	u_p	$\star$	π_0

$\exists$ wins **iff** at any time, for some $u_p.\pi_p$ previous move, the process $u_p \star \pi_0$ arrives on execution.

The game $\mathbb{G}_0$

ℓ ($\forall$ played moves)	$\exists$		$\forall$				
(initialisation phase) $\emptyset$	t_0	$\star$	$u_0.\pi_0$	$\succ$	u_0	$\star$	$t_1.\ \pi_0$
$\{u_1.\pi_1\}$	t_1	$\star$	$u_1.\pi_1$	$\succ$	u_0	$\star$	$t_2.\ \pi_0$
$\vdots$				$\vdots$			
$\{u_1.\pi_1, \ldots, u_n.\pi_n\}$	t_n	$\star$	$u_n.\pi_n$	$\succ$	u_p	$\star$	π_0

$\exists$ wins **iff** at any time, for some $u_p.\pi_p$ previous move, the process $u_p \star \pi_0$ arrives on execution. **Otherwise** $\forall$ wins.

The game $\mathbb{G}_0$ (cont.)

$$\begin{array}{lclclcl}
C_{n,p} & \star & u_0.\pi_0 & \succ & u_0 & \star & K^1_{n,p}[u_0,\pi_0].\pi_0 \\
K^1_{n,p} & \star & u_1.\pi_1 & \succ & u_0 & \star & K^2_{n,p}[u_0,\pi_0,u_1,\pi_1].\pi_0 \\
 & & & \vdots & & & \\
K^n_{n,p} & \star & u_n.\pi_n & \succ & u_p & \star & \pi_0
\end{array}$$

Remark

The terms $C_{n,p}$ are *uniform* winning strategies for $\mathbb{G}_0$, in the sense that all plays have the very same structure:

- They all have the same length ($2n+1$ moves)
- In all of they $\exists$ wins using the p-th move of $\forall$.

The game $\mathbb{G}_0$ (cont.)

$$\begin{array}{lclclcl} C_{n,p} & \star & u_0.\pi_0 & \succ & u_0 & \star & K^1_{n,p}[u_0,\pi_0].\pi_0 \\ K^1_{n,p} & \star & u_1.\pi_1 & \succ & u_0 & \star & K^2_{n,p}[u_0,\pi_0,u_1,\pi_1].\pi_0 \\ & & & \vdots & & & \\ K^n_{n,p} & \star & u_n.\pi_n & \succ & u_p & \star & \pi_0 \end{array}$$

Remark

The terms $C_{n,p}$ are *uniform* winning strategies for $\mathbb{G}_0$, in the sense that all plays have the very same structure:

- They all have the same length ($2n+1$ moves)
- In all of they $\exists$ wins using the p-th move of $\forall$.

The game $\mathbb{G}_0$ (cont.)

$$
\begin{array}{ccccccc}
C_{n,p} & \star & u_0.\pi_0 & \succ & u_0 & \star & K^1_{n,p}[u_0,\pi_0].\pi_0 \\
K^1_{n,p} & \star & u_1.\pi_1 & \succ & u_0 & \star & K^2_{n,p}[u_0,\pi_0,u_1,\pi_1].\pi_0 \\
 & & & \vdots & & & \\
K^n_{n,p} & \star & u_n.\pi_n & \succ & u_p & \star & \pi_0
\end{array}
$$

Remark

The terms $C_{n,p}$ are *uniform* winning strategies for $\mathbb{G}_0$, in the sense that all plays have the very same structure:

- They all have the same length ($2n+1$ moves)
- In all of they $\exists$ wins using the p-th move of $\forall$.

The game $\mathbb{G}_0$ (cont.)

Syntactic definition of $\mathbb{G}_0$

The game $\mathbb{G}_0$ (cont.)

Syntactic definition of $\mathbb{G}_0$

We describe the states of $\mathbb{G}_0$ by pairs $\langle p, \ell \rangle$ where $p \in \Lambda \star \Pi$ is the head of the current thread and $\ell \subseteq \Pi$ is the set of $\forall$-moves.

The game $\mathbb{G}_0$ (cont.)

Syntactic definition of $\mathbb{G}_0$

We describe the states of $\mathbb{G}_0$ by pairs $\langle p, \ell \rangle$ where $p \in \Lambda \star \Pi$ is the head of the current thread and $\ell \subseteq \Pi$ is the set of $\forall$-moves.

Given a stack $u_0.\pi_0$, we define the set $W_{u_0.\pi_0}$ of *winning states* as follows:

The game $\mathbb{G}_0$ (cont.)

Syntactic definition of $\mathbb{G}_0$

We describe the states of $\mathbb{G}_0$ by pairs $\langle p, \ell \rangle$ where $p \in \Lambda \star \Pi$ is the head of the current thread and $\ell \subseteq \Pi$ is the set of $\forall$-moves.

Given a stack $u_0.\pi_0$, we define the set $W_{u_0.\pi_0}$ of *winning states* as follows:

$$\frac{}{\langle p, \ell \rangle \in W_{u_0.\pi_0}} \quad (\text{if } p \succ u \star \pi_0 \text{ for some } u.\pi \in \ell)$$

The game $\mathbb{G}_0$ (cont.)

Syntactic definition of $\mathbb{G}_0$

We describe the states of $\mathbb{G}_0$ by pairs $\langle p, \ell \rangle$ where $p \in \Lambda \star \Pi$ is the head of the current thread and $\ell \subseteq \Pi$ is the set of $\forall$-moves.

Given a stack $u_0.\pi_0$, we define the set $W_{u_0.\pi_0}$ of *winning states* as follows:

$$\frac{}{\langle p, \ell \rangle \in W_{u_0.\pi_0}} \quad (\text{if } p \succ u \star \pi_0 \text{ for some } u.\pi \in \ell)$$

$$\frac{\langle t \star u.\pi, \ell \cup \{u.\pi\} \rangle \in W_{u_0.\pi_0} \quad \text{for all } u.\pi \in \Pi}{\langle p, \ell \rangle \in W_{u_0.\pi_0}} \quad (\text{if } p \succ u_0 \star t.\pi_0)$$

The game $\mathbb{G}_0$ (cont.)

Definition

A closed λ-term t_0 is a *winning strategy for* $\mathbb{G}_0$ **iff** $\langle t_0 \star u_0.\pi_0, \emptyset\rangle \in W_{u_0.\pi_0}$ for all stack $u_0.\pi_0$

Adequacy

Definition

A closed λ-term t_0 is a *winning strategy for* $\mathbb{G}_0$ **iff** $\langle t_0 \star u_0.\pi_0, \emptyset\rangle \in W_{u_0.\pi_0}$ for all stack $u_0.\pi_0$

Proposition: Adequacy of $\Vdash$ w.r.t. $\mathbb{G}_0$

If t_0 is a winning strategy for $\mathbb{G}_0$ **then** $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$.

Adequacy

Definition

A closed λ-term t_0 is a *winning strategy for* $\mathbb{G}_0$ **iff** $\langle t_0 \star u_0.\pi_0, \emptyset\rangle \in W_{u_0.\pi_0}$ for all stack $u_0.\pi_0$

Proposition: Adequacy of $\Vdash$ w.r.t. $\mathbb{G}_0$

If t_0 is a winning strategy for $\mathbb{G}_0$ **then** $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$.

Proof.

Consider a pole $\perp\!\!\!\perp$ and a falsity value $\mathbb{X}$. We must prove:

$$t_0 \Vdash (\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}$$

□

Adequacy (cont.)

Proof.

... Take $u_0.\pi_0 \in \|(\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}\|$

□

Adequacy (cont.)

Proof.

... Take $u_0.\pi_0 \in \|(\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}\|$

Lemma

If $\langle p, \ell \rangle \in W_{u_0.\pi_0}$ *and* $\ell \subseteq \neg\mathbb{X}$ **then** $p \in \perp\!\!\!\perp$.

□

Adequacy (cont.)

Proof.

... Take $u_0.\pi_0 \in \|(\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}\|$

Lemma

If $\langle p, \ell\rangle \in W_{u_0.\pi_0}$ *and* $\ell \subseteq \neg\mathbb{X}$ **then** $p \in \perp\!\!\!\perp$.

Proof.

Induction on the derivation of $\langle p, \ell\rangle \in W_{u_0.\pi_0}$ □

□

Adequacy (cont.)

Proof.

... Take $u_0.\pi_0 \in ||(\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}||$

Lemma

If $\langle p, \ell\rangle \in W_{u_0.\pi_0}$ *and* $\ell \subseteq \neg\mathbb{X}$ **then** $p \in \perp\!\!\!\perp$.

Proof.

Induction on the derivation of $\langle p, \ell\rangle \in W_{u_0.\pi_0}$ □

Since $\langle t_0 \star u_0.\pi_0, \emptyset\rangle \in W_{u_0.\pi_0}$ we have the result. □

Completeness

Proposition: Completeness of $\Vdash$ w.r.t. $\mathbb{G}_0$

Suppose the calculus of realizers contains:

- infinitely many interaction constants
- infinitely many substitutive and non generative stack constants

Completeness

Proposition: Completeness of $\Vdash$ w.r.t. $\mathbb{G}_0$

Suppose the calculus of realizers contains:

- infinitely many interaction constants
- infinitely many substitutive and non generative stack constants

Completeness

Proposition: Completeness of $\Vdash$ w.r.t. $\mathbb{G}_0$

Suppose the calculus of realizers contains:

- infinitely many interaction constants
- infinitely many substitutive and non generative stack constants

Completeness

Proposition: Completeness of $\Vdash$ w.r.t. $\mathbb{G}_0$

Suppose the calculus of realizers contains:

- infinitely many interaction constants
- infinitely many substitutive and non generative stack constants

If $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$ **then** t_0 is a winning strategy for the game $\mathbb{G}_0$

Completeness

Proposition: Completeness of $\Vdash$ w.r.t. $\mathbb{G}_0$

Suppose the calculus of realizers contains:

- infinitely many interaction constants
- infinitely many substitutive and non generative stack constants

If $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$ **then** t_0 is a winning strategy for the game $\mathbb{G}_0$

Proof.

Consider $(K_i.\alpha_i)_{i \in \mathbb{N}}$ like in the red hypothesis and 'fresh' for t_0.

□

Completeness

Proposition: Completeness of $\Vdash$ w.r.t. $\mathbb{G}_0$

Suppose the calculus of realizers contains:

- infinitely many interaction constants
- infinitely many substitutive and non generative stack constants

If $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$ **then** t_0 is a winning strategy for the game $\mathbb{G}_0$

Proof.

Consider $(K_i.\alpha_i)_{i\in\mathbb{N}}$ like in the red hypothesis and 'fresh' for t_0.

Lemma

t_0 wins the game $\mathbb{G}_0$ against $\forall$ playing $(k_i.\alpha_i)_{i\in\mathbb{N}}$

□

Completeness (cont.)

Proof.

. . .

Proof.

By threads method:

□

. . .

□

Completeness (cont.)

Proof.

...

Proof.

By threads method:

Consider the sequence of the threads of the play

$$\begin{array}{lccccccc} \mathcal{Q}_0) & t_0 & \star & K_0.\alpha_0 & \succ & K_0 & \star & t_1.\alpha_0 \\ \mathcal{Q}_1) & t_1 & \star & K_1.\alpha_1 & \succ & K_0 & \star & t_2.\alpha_0 \\ & & & & \vdots & & & \end{array}$$

□

...

□

Completeness (cont.)

Proof.

. . .

Proof.

By threads method:

Consider the sequence of the threads of the play

$$
\begin{array}{rccccccc}
\mathcal{Q}_0) & t_0 & \star & K_0.\alpha_0 & \succ & K_0 & \star & t_1.\alpha_0 \\
\mathcal{Q}_1) & t_1 & \star & K_1.\alpha_1 & \succ & K_0 & \star & t_2.\alpha_0 \\
 & & & & \vdots & & & \\
\mathcal{Q}_i) & t_i & \star & K_i.\alpha_i & \succ & \ldots & & \\
\mathcal{Q}_{i+1}) & \emptyset & & & & & &
\end{array}
$$

□

. . .

□

Completeness (cont.)

Proof.

. . .

Proof.

By threads method:

Consider the sequence of the threads of the play

$$
\begin{array}{rlllllll}
\mathcal{Q}_0) & t_0 & \star & K_0.\alpha_0 & \succ & K_0 & \star & t_1.\alpha_0 \\
\mathcal{Q}_1) & t_1 & \star & K_1.\alpha_1 & \succ & K_0 & \star & t_2.\alpha_0 \\
 & & & & \vdots & & & \\
\mathcal{Q}_i) & t_i & \star & K_i.\alpha_i & \succ & K_0 & \star & t_{i+1}.\alpha_0 \\
\mathcal{Q}_{i+1}) & & & & & & &
\end{array}
$$

□

. . .

□

Completeness (cont.)

Proof.

...

Proof.

By threads method:

Consider the sequence of the threads of the play

$$\begin{array}{rcccccc}
\mathcal{Q}_0) & t_0 & \star & K_0.\alpha_0 & \succ & K_0 & \star & t_1.\alpha_0 \\
\mathcal{Q}_1) & t_1 & \star & K_1.\alpha_1 & \succ & K_0 & \star & t_2.\alpha_0 \\
 & & & & \vdots & & & \\
\mathcal{Q}_i) & t_i & \star & K_i.\alpha_i & \succ & K_0 & \star & t_{i+1}.\alpha_0 \\
\mathcal{Q}_{i+1}) & t_{i+1} & \star & K_{i+1}.\alpha_{i+1} & \dots & & &
\end{array}$$

□

...

□

Completeness (cont.)

Proof.

...

Proof.

By threads method:

Consider the sequence of the threads of the play

$$\begin{array}{rcccccccc}
\mathcal{Q}_0) & t_0 & \star & K_0.\alpha_0 & \succ & K_0 & \star & t_1.\alpha_0 \\
\mathcal{Q}_1) & t_1 & \star & K_1.\alpha_1 & \succ & K_0 & \star & t_2.\alpha_0 \\
 & & & & \vdots & & & \\
\mathcal{Q}_i) & t_i & \star & K_i.\alpha_i & \succ & K_0 & \star & t_{i+1}.\alpha_0 \\
\mathcal{Q}_{i+1}) & t_{i+1} & \star & K_{i+1}.\alpha_{i+1} & \ldots & & &
\end{array}$$

Define $\perp\!\!\!\perp := (\bigcup_{i \in \mathbb{N}} \mathcal{Q}_i)^c$

...

□

...

□

Completeness (cont.)

Proof.

...

Proof.

...

- Consider $\mathbb{X} := \{\alpha_0\}$.
- $K_0 \not\Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ since $t_0 \Vdash (\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}$ and $t_0 \star K_0.\alpha_0 \notin \perp\!\!\!\perp$.
- Then there is a term $t \Vdash \neg\mathbb{X}$ s.t. $K_0 \star t.\alpha_0$ belongs to a thread, namely $\mathcal{Q}_{p-1}$.
- Therefore, $\mathcal{Q}_p = \mathrm{th}(t \star K_p.\alpha_p)$.
- Since $t \Vdash \mathbb{X} \Rightarrow \bot$, $K_p \not\Vdash \mathbb{X}$, $K_p \star \alpha_0 \notin \perp\!\!\!\perp$

... □

... □

Completeness (cont.)

Proof.

...

Proof.

...

- Consider $\mathbb{X} := \{\alpha_0\}$.
- $K_0 \not\Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ since $t_0 \Vdash (\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}$ and $t_0 \star K_0.\alpha_0 \notin \perp\!\!\!\perp$.
- Then there is a term $t \Vdash \neg\mathbb{X}$ s.t. $K_0 \star t.\alpha_0$ belongs to a thread, namely $\mathcal{Q}_{p-1}$.
- Therefore, $\mathcal{Q}_p = \text{th}(t \star K_p.\alpha_p)$.
- Since $t \Vdash \mathbb{X} \Rightarrow \perp$, $K_p \not\Vdash \mathbb{X}$, $K_p \star \alpha_0 \notin \perp\!\!\!\perp$

...

□

...

□

Completeness (cont.)

Proof.

...

Proof.

...

- Consider $\mathbb{X} := \{\alpha_0\}$.
- $K_0 \not\Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ since $t_0 \Vdash (\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}$ and $t_0 \star K_0.\alpha_0 \notin \perp\!\!\!\perp$.
- Then there is a term $t \Vdash \neg\mathbb{X}$ s.t. $K_0 \star t.\alpha_0$ belongs to a thread, namely $\mathcal{Q}_{p-1}$.
- Therefore, $\mathcal{Q}_p = \mathrm{th}(t \star K_p.\alpha_p)$.
- Since $t \Vdash \mathbb{X} \Rightarrow \perp$, $K_p \not\Vdash \mathbb{X}$, $K_p \star \alpha_0 \notin \perp\!\!\!\perp$

... □

... □

Completeness (cont.)

Proof.

...

Proof.

...

- Consider $\mathbb{X} := \{\alpha_0\}$.
- $K_0 \not\Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ since $t_0 \Vdash (\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}$ and $t_0 \star K_0.\alpha_0 \notin \perp\!\!\!\perp$.
- Then there is a term $t \Vdash \neg\mathbb{X}$ s.t. $K_0 \star t.\alpha_0$ belongs to a thread, namely $\mathcal{Q}_{p-1}$.
- Therefore, $\mathcal{Q}_p = \text{th}(t \star K_p.\alpha_p)$.
- Since $t \Vdash \mathbb{X} \Rightarrow \perp$, $K_p \not\Vdash \mathbb{X}$, $K_p \star \alpha_0 \notin \perp\!\!\!\perp$

... □

... □

Completeness (cont.)

Proof.

...

Proof.

...

- Consider $\mathbb{X} := \{\alpha_0\}$.
- $K_0 \not\Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ since $t_0 \Vdash (\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}$ and $t_0 \star K_0.\alpha_0 \notin \perp\!\!\!\perp$.
- Then there is a term $t \Vdash \neg\mathbb{X}$ s.t. $K_0 \star t.\alpha_0$ belongs to a thread, namely $\mathcal{Q}_{p-1}$.
- Therefore, $\mathcal{Q}_p = \mathrm{th}(t \star K_p.\alpha_p)$.
- Since $t \Vdash \mathbb{X} \Rightarrow \bot$, $\quad K_p \not\Vdash \mathbb{X}$, $\quad K_p \star \alpha_0 \notin \perp\!\!\!\perp$

... □

... □

Proof.

...

Proof.

...

- Then, there is an n such that $K_p \star \alpha_0 \in \mathcal{Q}_n$.
- We have $p \leq n$ since K_i's are *non generative* and does *not occur in* t_0.
- Moreover, K_i's are *inert* and then the threads of this play are the following:

□

Proof.

...

Proof.

...

- Then, there is an n such that $K_p \star \alpha_0 \in \mathcal{Q}_n$.
- We have $p \leq n$ since K_i's are *non generative* and does *not occur in* t_0.
- Moreover, K_i's are *inert* and then the threads of this play are the following:

Proof.

...

Proof.

...

- Then, there is an n such that $K_p \star \alpha_0 \in \mathcal{Q}_n$.
- We have $p \leq n$ since K_i's are *non generative* and does *not occur in* t_0.
- Moreover, K_i's are *inert* and then the threads of this play are the following:

Proof.

...

Proof.

...

- Then, there is an n such that $K_p \star \alpha_0 \in \mathcal{Q}_n$.
- We have $p \leq n$ since K_i's are *non generative* and does *not occur in* t_0.
- Moreover, K_i's are *inert* and then the threads of this play are the following:

$$
\begin{array}{lccccccc}
\mathcal{Q}_0) & t_0 & \star & K_0.\alpha_0 & \succ & K_0 & \star & t_1.\alpha_0 \\
 & & & & \vdots & & & \\
\mathcal{Q}_{p-1}) & t_{i-1} & \star & K_{p-1}.\alpha_{p-1} & \succ & K_0 & \star & t_p.\alpha_0 \\
 & & & & \vdots & & & \\
\mathcal{Q}_n) & t_n & \star & K_n.\alpha_n & \succ & K_p & \star & \alpha_0
\end{array}
$$

□

Completeness (cont.)

Proof.

... Know, since K_i's and α_i's are substitutives, we can reason by (dynamic) substitution, thus proving the result. □

Completeness (cont.)

Proof.

... Know, since K_i's and α_i's are substitutives, we can reason by (dynamic) substitution, thus proving the result. □

Remark

All plays played by t_0 have the same lenght ($n+1$ threads) and chooses the p-th $\forall$-move to win.

Proof.

… Know, since K_i's and α_i's are substitutives, we can reason by (dynamic) substitution, thus proving the result. □

Remark

All plays played by t_0 have the same lenght ($n + 1$ threads) and chooses the p-th $\forall$-move to win.

Conclusion

For a Krivine machine compatible with interaction instructions, all realizers of Peirce's Law are uniform w.s. for $\mathbb{G}_0$, i.e.: they have the same behaviour than a suitable $C_{n,p}$.

Consider a term t such that for any stack $u.\pi$:

$$t \star u.\pi \succ u \star k.\pi$$

Consider a term t such that for any stack $u.\pi$:

$$t \star u.\pi \succ u' \star \pi \succ u \star k.\pi$$

suppose k satisfies for any stack $u''.\pi''$:

$$k \star u''.\pi'' \succ \begin{cases} k_\pi \star u''.\pi'' & \textbf{If}\, u'' \not\equiv u' \\ \text{any process} & \textbf{otherwise} \end{cases}$$

Consider a term t such that for any stack $u.\pi$:

$$t \star u.\pi \succ u' \star \pi \succ u \star k.\pi$$

suppose k satisfies for any stack $u''.\pi''$:

$$k \star u''.\pi'' \succ \begin{cases} k_\pi \star u''.\pi'' & \textbf{If}\, u'' \not\equiv u' \\ \text{or maybe...} & \textbf{otherwise} \end{cases}$$

Consider a term t such that for any stack $u.\pi$:

$$t \star u.\pi \succ u' \star \pi \succ u \star k.\pi$$

suppose k satisfies for any stack $u''.\pi''$:

$$k \star u''.\pi'' \succ \begin{cases} k_\pi \star u''.\pi'' & \textbf{If}\, u'' \not\equiv u' \\ \text{BANG!} & \textbf{otherwise} \end{cases}$$

Consider a term t such that for any stack $u.\pi$:

$$t \star u.\pi \succ u' \star \pi \succ u \star k.\pi$$

suppose k satisfies for any stack $u''.\pi''$:

$$k \star u''.\pi'' \succ \begin{cases} k_\pi \star u''.\pi'' & \textbf{If}\, u'' \not\equiv u' \\ \text{BANG!} & \textbf{otherwise} \end{cases}$$

Proposition

$$t \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$$

Consider a term t such that for any stack $u.\pi$:

$$t \star u.\pi \succ u' \star \pi \succ u \star k.\pi$$

suppose k satisfies for any stack $u''.\pi''$:

$$k \star u''.\pi'' \succ \begin{cases} k_\pi \star u''.\pi'' & \textbf{If}\, u'' \not\equiv u' \\ \text{BANG!} & \textbf{otherwise} \end{cases}$$

Proposition

$$t \Vvdash \forall X(\neg X \Rightarrow X) \Rightarrow X$$

Proof.

- Consider a pole $\perp\!\!\!\perp$ and a falsity value $\mathbb{X}$.
- Pick a term $u \Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ and a stack $\pi \in \mathbb{X}$.
- We must prove $t \star u.\pi \in \perp\!\!\!\perp$.
- By antireduction, **if** $u' \Vdash \mathbb{X}$ we are done...

Consider a term t such that for any stack $u.\pi$:

$$t \star u.\pi \succ u' \star \pi \succ u \star k.\pi$$

suppose k satisfies for any stack $u''.\pi''$:

$$k \star u''.\pi'' \succ \begin{cases} k_\pi \star u''.\pi'' & \textbf{If}\, u'' \not\equiv u' \\ \text{BANG!} & \textbf{otherwise} \end{cases}$$

Proposition

$$t \Vvdash \forall X(\neg X \Rightarrow X) \Rightarrow X$$

Proof.

- Consider a pole $\perp\!\!\!\perp$ and a falsity value $\mathbb{X}$.
- Pick a term $u \Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ and a stack $\pi \in \mathbb{X}$.
- We must prove $t \star u.\pi \in \perp\!\!\!\perp$.
- By antireduction, **if** $u' \Vdash \mathbb{X}$ we are done...

Consider a term t such that for any stack $u.\pi$:

$$t \star u.\pi \succ u' \star \pi \succ u \star k.\pi$$

suppose k satisfies for any stack $u''.\pi''$:

$$k \star u''.\pi'' \succ \begin{cases} k_\pi \star u''.\pi'' & \textbf{If}\, u'' \not\equiv u' \\ \text{BANG!} & \textbf{otherwise} \end{cases}$$

Proposition

$$t \Vvdash \forall X(\neg X \Rightarrow X) \Rightarrow X$$

Proof.

- Consider a pole $\perp\!\!\!\perp$ and a falsity value $\mathbb{X}$.
- Pick a term $u \Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ and a stack $\pi \in \mathbb{X}$.
- We must prove $t \star u.\pi \in \perp\!\!\!\perp$.
- By antireduction, **if** $u' \Vdash \mathbb{X}$ we are done...

Consider a term t such that for any stack $u.\pi$:

$$t \star u.\pi \succ u' \star \pi \succ u \star k.\pi$$

suppose k satisfies for any stack $u''.\pi''$:

$$k \star u''.\pi'' \succ \begin{cases} k_\pi \star u''.\pi'' & \textbf{If}\, u'' \not\equiv u' \\ \text{BANG!} & \textbf{otherwise} \end{cases}$$

Proposition

$$t \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$$

Proof.

- Consider a pole $\perp\!\!\!\perp$ and a falsity value $\mathbb{X}$.
- Pick a term $u \Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ and a stack $\pi \in \mathbb{X}$.
- We must prove $t \star u.\pi \in \perp\!\!\!\perp$.
- By antireduction, **if** $u' \Vdash \mathbb{X}$ we are done...

Consider a term t such that for any stack $u.\pi$:

$$t \star u.\pi \succ u' \star \pi \succ u \star k.\pi$$

suppose k satisfies for any stack $u''.\pi''$:

$$k \star u''.\pi'' \succ \begin{cases} k_\pi \star u''.\pi'' & \textbf{If}\, u'' \not\equiv u' \\ \text{BANG!} & \textbf{otherwise} \end{cases}$$

Proposition

$$t \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$$

Proof.

- Consider a pole $\perp\!\!\!\perp$ and a falsity value $\mathbb{X}$.
- Pick a term $u \Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ and a stack $\pi \in \mathbb{X}$.
- We must prove $t \star u.\pi \in \perp\!\!\!\perp$.
- By antireduction, **if** $u' \Vdash \mathbb{X}$ we are done...

$$t \star u.\pi \succ u' \star \pi \succ u \star k.\pi$$

k satisfies for any stack $u''.\pi''$:

$$k \star u''.\pi'' \succ \begin{cases} k_\pi \star u''.\pi'' & \textbf{If } u'' \not\equiv u' \\ \text{BANG!} & \textbf{otherwise} \end{cases}$$

Proof.

...

Otherwise, it suffices to prove: $u \star k.\pi \in \perp\!\!\!\perp$ and for that, it suffices to prove $k \Vdash \neg\mathbb{X}$.

Consider $u'' \Vdash \mathbb{X}$. By assumption, $u'' \not\equiv u'$ and hence $k \star u''.\pi'' \succ u \star \pi$ which is in $\perp\!\!\!\perp$ because $u \Vdash \mathbb{X}$ and $\pi \in \mathbb{X}$. □

How works a wild realizer?

How works a wild realizer?

Imagine you are arguing against someone....

How works a wild realizer?

Imagine you are arguing against someone....
You want to be sure your opponent is wrong, so you stand for a wrong argument...

How works a wild realizer?

Imagine you are arguing against someone....
You want to be sure your opponent is wrong, so you stand for a wrong argument...
Throughout the discussion, you imagine your opponent's possible arguments, to anticipate...

How works a wild realizer?

Imagine you are arguing against someone....
You want to be sure your opponent is wrong, so you stand for a wrong argument...
Throughout the discussion, you imagine your opponent's possible arguments, to anticipate...
Hence, if your opponent gives an argument that you know how to refute, you can go conviced your opponent is wrong!

How works a wild realizer?

Imagine you are arguing against someone....
You want to be sure your opponent is wrong, so you stand for a wrong argument...
Throughout the discussion, you imagine your opponent's possible arguments, to anticipate...
Hence, if your opponent gives an argument that you know how to refute, you can go conviced your opponent is wrong!
Imagine the arguments of your opponent are of the form:

> *The process* $u.\pi_0$ *never arrives on execution in our discussion*

How works a wild realizer?

Imagine you are arguing against someone....
You want to be sure your opponent is wrong, so you stand for a wrong argument...
Throughout the discussion, you imagine your opponent's possible arguments, to anticipate...
Hence, if your opponent gives an argument that you know how to refute, you can go conviced your opponent is wrong!
Imagine the arguments of your opponent are of the form:

> *The process $u.\pi_0$ never arrives on execution in our discussion*

If $u \star \pi_0$ was yet on execution, you know he's wrong!

Are there wild realizers?

In order to obtain a wild realizer, you can write a term cc′ satisfying:

cc′ evaluation

Are there wild realizers?

In order to obtain a wild realizer, you can write a term cc′ satisfying:

cc′ evaluation

for all stacks $u.\pi$

$$\mathrm{cc}' \star u.\pi \quad \succ \quad T[u] \star \pi \quad \succ \quad u \star K[u,\pi].\pi$$

Are there wild realizers?

In order to obtain a wild realizer, you can write a term cc′ satisfying:

cc′ evaluation

for all stacks $u.\pi$

$$\mathrm{cc}' \star u.\pi \quad \succ \quad T[u] \star \pi \quad \succ \quad u \star K[u,\pi].\pi$$

where this evaluation uses only the rules of the basic machine

Are there wild realizers?

In order to obtain a wild realizer, you can write a term cc′ satisfying:

cc′ evaluation

for all stacks $u.\pi$

$$\mathrm{cc}' \star u.\pi \quad \succ \quad T[u] \star \pi \quad \succ \quad u \star K[u,\pi].\pi$$

where this evaluation uses only the rules of the basic machine and $T[u] \star \pi$ is the sole process with the stack π in this thread

Are there wild realizers?

In order to obtain a wild realizer, you can write a term cc′ satisfying:

cc′ evaluation

for all stacks $u.\pi$

$$\mathrm{cc}' \star u.\pi \quad \succ \quad T[u] \star \pi \quad \succ \quad u \star K[u,\pi].\pi$$

where this evaluation uses only the rules of the basic machine and $T[u] \star \pi$ is the sole process with the stack π in this thread

$$K[u,\pi] \star u'.\pi' \quad \succ \quad \begin{cases} k_\pi \star u'.\pi' & u' \not\equiv T[u] \\ \mathbf{H} \star \pi' & \textbf{Otherwise} \end{cases}$$

Are there wild realizers?

In order to obtain a wild realizer, you can write a term cc' satisfying:

cc' evaluation

for all stacks $u.\pi$

$$\mathrm{cc}' \star u.\pi \quad \succ \quad T[u] \star \pi \quad \succ \quad u \star K[u,\pi].\pi$$

where this evaluation uses only the rules of the basic machine and $T[u] \star \pi$ is the sole process with the stack π in this thread

$$K[u,\pi] \star u'.\pi' \quad \succ \quad \begin{cases} k_\pi \star u'.\pi' & u' \not\equiv T[u] \\ \mathbf{H} \star \pi' & \textbf{Otherwise} \end{cases}$$

where $\mathbf{H}$ is an inert constant and K is written using (eq)

Proposition

$$\mathsf{cc}' \quad \Vdash \quad \forall X(\neg X \Rightarrow X) \Rightarrow X$$

Proposition

$$\mathrm{cc}' \quad \Vdash \quad \forall X(\neg X \Rightarrow X) \Rightarrow X$$

Proposition

cc' is not a winning strategy for the game $\mathbb{G}_0$

Proposition

$$\mathrm{cc}' \quad \Vdash \quad \forall X(\neg X \Rightarrow X) \Rightarrow X$$

Proposition

cc′ is not a winning strategy for the game $\mathbb{G}_0$

Proof.

We describe a winning strategy for $\forall$:

- $\forall$ initializes the game with the stack $\mathbf{H}.\alpha_0$
- The first thread finishes on the process $\mathbf{H} \star K[\mathbf{H}, \alpha_0].\alpha_0$ and hence $\forall$ must play $K[\mathbf{H}, \alpha_0]$

. . .

□

Proposition

$$\mathrm{cc}' \quad \Vdash \quad \forall X(\neg X \Rightarrow X) \Rightarrow X$$

Proposition

cc' is not a winning strategy for the game $\mathbb{G}_0$

Proof.

We describe a winning strategy for $\forall$:

- $\forall$ initializes the game with the stack $\mathbf{H}.\alpha_0$
- The first thread finishes on the process $\mathbf{H} \star K[\mathbf{H}, \alpha_0].\alpha_0$ and hence $\forall$ must play $K[\mathbf{H}, \alpha_0]$

. . .

□

Proof.

...

- $\forall$ answers $T[\mathbf{H}].\alpha_0$. Then, the second thread finishes on the process $\mathbf{H} \star \alpha_0$.
- Since the term $\mathbf{H}$ was not played before by $\forall$ and $\exists$ cannot play again, $\forall$ has win.

□

The game $\mathbb{G}_1$

Syntactic definition of $\mathbb{G}_1$

We represent the $\exists$-current position by the set of the heads of all threads currently played. The set of $\forall$ played moves is $\ell \subseteq \Pi$.

$$\frac{}{\langle P, \ell \rangle \in W'_{u_0.\pi_0}} \text{(If } p \succ u \star \pi_0 \text{ for some } p \in P \text{ and } u.\pi \in \ell)$$

$$\frac{\langle P \cup \{t \star u.\pi\}, \ell \cup \{u.\pi\} \rangle \in W'_{u_0.\pi_0} \quad \text{for all } u.\pi \in \Pi}{\langle P, \ell \rangle \in W'_{u_0.\pi_0}}$$

$$\text{(if } p \succ u_0 \star t.\pi_0;\ p \in P)$$

The game $\mathbb{G}_1$

Syntactic definition of $\mathbb{G}_1$

We represent the $\exists$-current position by the set of the heads of all threads currently played. The set of $\forall$ played moves is $\ell \subseteq \Pi$.

$$\frac{}{\langle P, \ell \rangle \in W'_{u_0.\pi_0}} \quad (\text{If } p \succ u \star \pi_0 \text{ for some } p \in P \text{ and } u.\pi \in \ell)$$

$$\frac{\langle P \cup \{t \star u.\pi\}, \ell \cup \{u.\pi\} \rangle \in W'_{u_0.\pi_0} \quad \text{for all } u.\pi \in \Pi}{\langle P, \ell \rangle \in W'_{u_0.\pi_0}} \quad (\text{if } p \succ u_0 \star t.\pi_0;\ p \in P)$$

The game $\mathbb{G}_1$

Syntactic definition of $\mathbb{G}_1$

We represent the $\exists$-current position by the set of the heads of all threads currently played. The set of $\forall$ played moves is $\ell \subseteq \Pi$.

$$\frac{}{\langle P, \ell \rangle \in W'_{u_0 . \pi_0}} \quad (\text{If } p \succ u \star \pi_0 \text{ for some } p \in P \text{ and } u.\pi \in \ell)$$

$$\frac{\langle P \cup \{t \star u.\pi\}, \ell \cup \{u.\pi\} \rangle \in W'_{u_0 . \pi_0} \quad \text{for all } u.\pi \in \Pi}{\langle P, \ell \rangle \in W'_{u_0 . \pi_0}}$$

$$(\text{if } p \succ u_0 \star t.\pi_0;\ p \in P)$$

Definition

A closed λ-term t_0 is a *winning strategy for* $\mathbb{G}_1$ **iff** $\langle\{t_0 \star u_0.\pi_0\}, \emptyset\rangle \in W'_{u_0.\pi_0}$ for all stack $u_0.\pi_0$

Definition

A closed λ-term t_0 is a *winning strategy for* $\mathbb{G}_1$ **iff** $\langle\{t_0 \star u_0.\pi_0\}, \emptyset\rangle \in W'_{u_0.\pi_0}$ for all stack $u_0.\pi_0$

Proposition: Adequacy of $\Vdash$ w.r.t. $\mathbb{G}_1$

If t_0 is a winning strategy for $\mathbb{G}_1$ **then** $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$.

Definition

A closed λ-term t_0 is a *winning strategy for* $\mathbb{G}_1$ **iff** $\langle\{t_0 \star u_0.\pi_0\}, \emptyset\rangle \in W'_{u_0.\pi_0}$ for all stack $u_0.\pi_0$

Proposition: Adequacy of $\Vdash$ w.r.t. $\mathbb{G}_1$

If t_0 is a winning strategy for $\mathbb{G}_1$ **then** $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$.

Proof.

□

Definition

A closed λ-term t_0 is a *winning strategy for* $\mathbb{G}_1$ **iff** $\langle\{t_0 \star u_0.\pi_0\}, \emptyset\rangle \in W'_{u_0.\pi_0}$ for all stack $u_0.\pi_0$

Proposition: Adequacy of $\Vdash$ w.r.t. $\mathbb{G}_1$

If t_0 is a winning strategy for $\mathbb{G}_1$ **then** $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$.

Proof.

Lemma

If $\langle P, \ell\rangle \in W'_{u_0.\pi_0}$ *and* $\ell \subseteq \neg\mathbb{X}$, **then** $P \cap \perp\!\!\!\perp \neq \emptyset$.

□

Definition

A closed λ-term t_0 is a *winning strategy for* $\mathbb{G}_1$ **iff** $\langle\{t_0 \star u_0.\pi_0\}, \emptyset\rangle \in W'_{u_0.\pi_0}$ for all stack $u_0.\pi_0$

Proposition: Adequacy of $\Vdash$ w.r.t. $\mathbb{G}_1$

If t_0 is a winning strategy for $\mathbb{G}_1$ **then** $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$.

Proof.

Lemma

If $\langle P, \ell\rangle \in W'_{u_0.\pi_0}$ *and* $\ell \subseteq \neg\mathbb{X}$, **then** $P \cap \perp\!\!\!\perp \neq \emptyset$.

And proceed as for the Adequacy of $\mathbb{G}_0$ □

Completeness

Proposition: Completeness of $\Vdash$ w.r.t. $\mathbb{G}_1$

If $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$ **then** t_0 is a winning strategy for the game $\mathbb{G}_1$

Proof.

- Suppose $\langle t_0 \star u_0.\pi_0, \emptyset \rangle \notin W'_{u_0.\pi_0}$ for some $u_0.\pi_0$.
- Consider a surjection $\phi : \mathbb{N} \to \Lambda$ s.t. $\phi^{-1}(t)$ is infinite for all t.
- Define $\langle P_0, \ell_0 \rangle := \langle \{t_0 \star u_0.\pi_0\}, \emptyset \rangle \notin W'_{u_0.\pi_0}$
- Suppose defined $\langle P_i, \ell_i \rangle$.
    - If $p \succ u_0 \star \phi(i).\pi_0$ for some $p \in P_i$, by assumption there is a stack $u.\pi$ s. t. $\langle P_{i+1}, \ell_{i+1} \rangle := \langle P_i \cup \{\phi(i) \star u.\pi\}, \ell_i \cup \{u.\pi\} \rangle$ is not a winning state.
    - **Otherwise** $\langle P_{i+1}, \ell_{i+1} \rangle := \langle P_i, \ell_i \rangle$

Completeness

Proposition: Completeness of $\Vdash$ w.r.t. $\mathbb{G}_1$

If $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$ **then** t_0 is a winning strategy for the game $\mathbb{G}_1$

Proof.

- Suppose $\langle t_0 \star u_0.\pi_0, \emptyset\rangle \notin W'_{u_0.\pi_0}$ for some $u_0.\pi_0$.
- Consider a surjection $\phi : \mathbb{N} \to \Lambda$ s.t. $\phi^{-1}(t)$ is infinite for all t.
- Define $\langle P_0, \ell_0\rangle := \langle\{t_0 \star u_0.\pi_0\}, \emptyset\rangle \notin W'_{u_0.\pi_0}$
- Suppose defined $\langle P_i, \ell_i\rangle$.
  - **If** $p \succ u_0 \star \phi(i).\pi_0$ for some $p \in P_i$, by assumption there is a stack $u.\pi$ s. t. $\langle P_{i+1}, \ell_{i+1}\rangle := \langle P_i \cup \{\phi(i) \star u.\pi\}, \ell_i \cup \{u.\pi\}\rangle$ is not a winning state.
  - **Otherwise** $\langle P_{i+1}, \ell_{i+1}\rangle := \langle P_i, \ell_i\rangle$

Completeness

Proposition: Completeness of $\Vdash$ w.r.t. $\mathbb{G}_1$

If $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$ **then** t_0 is a winning strategy for the game $\mathbb{G}_1$

Proof.

- Suppose $\langle t_0 \star u_0.\pi_0, \emptyset\rangle \notin W'_{u_0.\pi_0}$ for some $u_0.\pi_0$.
- Consider a surjection $\phi : \mathbb{N} \to \Lambda$ s.t. $\phi^{-1}(t)$ is infinite for all t.
- Define $\langle P_0, \ell_0\rangle := \langle\{t_0 \star u_0.\pi_0\}, \emptyset\rangle \notin W'_{u_0.\pi_0}$
- Suppose defined $\langle P_i, \ell_i\rangle$.
  - If $p \succ u_0 \star \phi(i).\pi_0$ for some $p \in P_i$, by assumption there is a stack $u.\pi$ s. t. $\langle P_{i+1}, \ell_{i+1}\rangle := \langle P_i \cup \{\phi(i) \star u.\pi\}, \ell_i \cup \{u.\pi\}\rangle$ is not a winning state.
  - **Otherwise** $\langle P_{i+1}, \ell_{i+1}\rangle := \langle P_i, \ell_i\rangle$

Completeness

Proposition: Completeness of $\Vdash$ w.r.t. $\mathbb{G}_1$

If $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$ **then** t_0 is a winning strategy for the game $\mathbb{G}_1$

Proof.

- Suppose $\langle t_0 \star u_0.\pi_0, \emptyset\rangle \notin W'_{u_0.\pi_0}$ for some $u_0.\pi_0$.
- Consider a surjection $\phi : \mathbb{N} \to \Lambda$ s.t. $\phi^{-1}(t)$ is infinite for all t.
- Define $\langle P_0, \ell_0\rangle := \langle\{t_0 \star u_0.\pi_0\}, \emptyset\rangle \notin W'_{u_0.\pi_0}$
- Suppose defined $\langle P_i, \ell_i\rangle$.
  - **If** $p \succ u_0 \star \phi(i).\pi_0$ for some $p \in P_i$, by assumption there is a stack $u.\pi$ s. t. $\langle P_{i+1}, \ell_{i+1}\rangle := \langle P_i \cup \{\phi(i) \star u.\pi\}, \ell_i \cup \{u.\pi\}\rangle$ is not a winning state.
  - **Otherwise** $\langle P_{i+1}, \ell_{i+1}\rangle := \langle P_i, \ell_i\rangle$

Completeness

Proposition: Completeness of $\Vdash$ w.r.t. $\mathbb{G}_1$

If $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$ **then** t_0 is a winning strategy for the game $\mathbb{G}_1$

Proof.

- Suppose $\langle t_0 \star u_0.\pi_0, \emptyset\rangle \notin W'_{u_0.\pi_0}$ for some $u_0.\pi_0$.
- Consider a surjection $\phi : \mathbb{N} \to \Lambda$ s.t. $\phi^{-1}(t)$ is infinite for all t.
- Define $\langle P_0, \ell_0\rangle := \langle\{t_0 \star u_0.\pi_0\}, \emptyset\rangle \notin W'_{u_0.\pi_0}$
- Suppose defined $\langle P_i, \ell_i\rangle$.
  - **If** $p \succ u_0 \star \phi(i).\pi_0$ for some $p \in P_i$, by assumption there is a stack $u.\pi$ s. t. $\langle P_{i+1}, \ell_{i+1}\rangle := \langle P_i \cup \{\phi(i) \star u.\pi\}, \ell_i \cup \{u.\pi\}\rangle$ is not a winning state.
  - **Otherwise** $\langle P_{i+1}, \ell_{i+1}\rangle := \langle P_i, \ell_i\rangle$

Completeness

Proposition: Completeness of $\Vdash$ w.r.t. $\mathbb{G}_1$

If $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$ **then** t_0 is a winning strategy for the game $\mathbb{G}_1$

Proof.

- Suppose $\langle t_0 \star u_0.\pi_0, \emptyset\rangle \notin W'_{u_0.\pi_0}$ for some $u_0.\pi_0$.
- Consider a surjection $\phi : \mathbb{N} \to \Lambda$ s.t. $\phi^{-1}(t)$ is infinite for all t.
- Define $\langle P_0, \ell_0\rangle := \langle\{t_0 \star u_0.\pi_0\}, \emptyset\rangle \notin W'_{u_0.\pi_0}$
- Suppose defined $\langle P_i, \ell_i\rangle$.
  - **If** $p \succ u_0 \star \phi(i).\pi_0$ for some $p \in P_i$, by assumption there is a stack $u.\pi$ s. t. $\langle P_{i+1}, \ell_{i+1}\rangle := \langle P_i \cup \{\phi(i) \star u.\pi\}, \ell_i \cup \{u.\pi\}\rangle$ is not a winning state.
  - **Otherwise** $\langle P_{i+1}, \ell_{i+1}\rangle := \langle P_i, \ell_i\rangle$

Completeness

Proposition: Completeness of $\Vdash$ w.r.t. $\mathbb{G}_1$

If $t_0 \Vdash \forall X(\neg X \Rightarrow X) \Rightarrow X$ **then** t_0 is a winning strategy for the game $\mathbb{G}_1$

Proof.

- Suppose $\langle t_0 \star u_0.\pi_0, \emptyset\rangle \notin W'_{u_0.\pi_0}$ for some $u_0.\pi_0$.
- Consider a surjection $\phi : \mathbb{N} \to \Lambda$ s.t. $\phi^{-1}(t)$ is infinite for all t.
- Define $\langle P_0, \ell_0\rangle := \langle\{t_0 \star u_0.\pi_0\}, \emptyset\rangle \notin W'_{u_0.\pi_0}$
- Suppose defined $\langle P_i, \ell_i\rangle$.
    - **If** $p \succ u_0 \star \phi(i).\pi_0$ for some $p \in P_i$, by assumption there is a stack $u.\pi$ s. t. $\langle P_{i+1}, \ell_{i+1}\rangle := \langle P_i \cup \{\phi(i) \star u.\pi\}, \ell_i \cup \{u.\pi\}\rangle$ is not a winning state.
    - **Otherwise** $\langle P_{i+1}, \ell_{i+1}\rangle := \langle P_i, \ell_i\rangle$

Completeness (cont.)

Proof.

...

- Define $P_\omega := \bigcup_{i\in\mathbb{N}} P_i$ and $\perp\!\!\!\perp^c := \bigcup_{p\in P_\omega} \mathbf{th}(p)$
- $u_0 \not\Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ since $t_0 \Vdash (\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}$.
- Thus $u_0 \star t.\pi_0 \in \mathbf{th}(p)$ for some $t \Vdash \neg\mathbb{X}$, $p \in P_n$ and $n \in \mathbb{N}$.
- Pick $n' \geq n$ s.t. $\phi(n') = t$. We have $p \in P_{n'}$ because $P_{n'} \supseteq P_n$.
- By definition, $P_{n'+1} = P_n \cup \{\phi(n') \star u.\pi\}$ for a suitable stack $u.\pi$.
- Then $t \star u.\pi \notin \perp\!\!\!\perp$, $u \not\Vdash \mathbb{X}$ and hence $u \star \pi_0 \in \perp\!\!\!\perp$.
- Taking $m \geq n' + 1$, $\langle P_m, \ell_m\rangle \in W'_{u_0.\pi_0}$, which leads a contradiction.

□

Completeness (cont.)

Proof.

...

- Define $P_\omega := \bigcup_{i\in\mathbb{N}} P_i$ and $\perp\!\!\!\perp^c := \bigcup_{p\in P_\omega} \mathbf{th}(p)$
- $u_0 \not\Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ since $t_0 \Vdash (\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}$.
- Thus $u_0 \star t.\pi_0 \in \mathbf{th}(p)$ for some $t \Vdash \neg\mathbb{X}$, $p \in P_n$ and $n \in \mathbb{N}$.
- Pick $n' \geq n$ s.t. $\phi(n') = t$. We have $p \in P_{n'}$ because $P_{n'} \supseteq P_n$.
- By definition, $P_{n'+1} = P_n \cup \{\phi(n') \star u.\pi\}$ for a suitable stack $u.\pi$.
- Then $t \star u.\pi \notin \perp\!\!\!\perp$, $u \not\Vdash \mathbb{X}$ and hence $u \star \pi_0 \in \perp\!\!\!\perp$.
- Taking $m \geq n' + 1$, $\langle P_m, \ell_m\rangle \in W'_{u_0.\pi_0}$, which leads a contradiction.

□

Completeness (cont.)

Proof.

...

- Define $P_\omega := \bigcup_{i\in\mathbb{N}} P_i$ and $\perp\!\!\!\perp^c := \bigcup_{p\in P_\omega} \mathbf{th}(p)$
- $u_0 \not\Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ since $t_0 \Vdash (\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}$.
- Thus $u_0 \star t.\pi_0 \in \mathbf{th}(p)$ for some $t \Vdash \neg\mathbb{X}$, $p \in P_n$ and $n \in \mathbb{N}$.
- Pick $n' \geq n$ s.t. $\phi(n') = t$. We have $p \in P_{n'}$ because $P_{n'} \supseteq P_n$.
- By definition, $P_{n'+1} = P_n \cup \{\phi(n') \star u.\pi\}$ for a suitable stack $u.\pi$.
- Then $t \star u.\pi \notin \perp\!\!\!\perp$, $u \not\Vdash \mathbb{X}$ and hence $u \star \pi_0 \in \perp\!\!\!\perp$.
- Taking $m \geq n' + 1$, $\langle P_m, \ell_m \rangle \in W'_{u_0.\pi_0}$, which leads a contradiction.

□

Completeness (cont.)

Proof.

...

- Define $P_\omega := \bigcup_{i\in\mathbb{N}} P_i$ and $\perp\!\!\!\perp^c := \bigcup_{p\in P_\omega} \mathbf{th}(p)$
- $u_0 \not\Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ since $t_0 \Vdash (\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}$.
- Thus $u_0 \star t.\pi_0 \in \mathbf{th}(p)$ for some $t \Vdash \neg\mathbb{X}$, $p \in P_n$ and $n \in \mathbb{N}$.
- Pick $n' \geq n$ s.t. $\phi(n') = t$. We have $p \in P_{n'}$ because $P_{n'} \supseteq P_n$.
- By definition, $P_{n'+1} = P_n \cup \{\phi(n') \star u.\pi\}$ for a suitable stack $u.\pi$.
- Then $t \star u.\pi \notin \perp\!\!\!\perp$, $u \not\Vdash \mathbb{X}$ and hence $u \star \pi_0 \in \perp\!\!\!\perp$.
- Taking $m \geq n' + 1$, $\langle P_m, \ell_m\rangle \in W'_{u_0.\pi_0}$, which leads a contradiction.

□

Completeness (cont.)

Proof.

...

- Define $P_\omega := \bigcup_{i\in\mathbb{N}} P_i$ and $\perp\!\!\!\perp^c := \bigcup_{p\in P_\omega} \mathbf{th}(p)$
- $u_0 \not\Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ since $t_0 \Vdash (\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}$.
- Thus $u_0 \star t.\pi_0 \in \mathbf{th}(p)$ for some $t \Vdash \neg\mathbb{X}$, $p \in P_n$ and $n \in \mathbb{N}$.
- Pick $n' \geq n$ s.t. $\phi(n') = t$. We have $p \in P_{n'}$ because $P_{n'} \supseteq P_n$.
- By definition, $P_{n'+1} = P_n \cup \{\phi(n') \star u.\pi\}$ for a suitable stack $u.\pi$.
- Then $t \star u.\pi \notin \perp\!\!\!\perp$, $u \not\Vdash \mathbb{X}$ and hence $u \star \pi_0 \in \perp\!\!\!\perp$.
- Taking $m \geq n' + 1$, $\langle P_m, \ell_m\rangle \in W'_{u_0.\pi_0}$, which leads a contradiction.

□

Completeness (cont.)

Proof.

...

- Define $P_\omega := \bigcup_{i\in\mathbb{N}} P_i$ and $\perp\!\!\!\perp^c := \bigcup_{p\in P_\omega} \mathbf{th}(p)$
- $u_0 \not\Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ since $t_0 \Vdash (\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}$.
- Thus $u_0 \star t.\pi_0 \in \mathbf{th}(p)$ for some $t \Vdash \neg\mathbb{X}$, $p \in P_n$ and $n \in \mathbb{N}$.
- Pick $n' \geq n$ s.t. $\phi(n') = t$. We have $p \in P_{n'}$ because $P_{n'} \supseteq P_n$.
- By definition, $P_{n'+1} = P_n \cup \{\phi(n') \star u.\pi\}$ for a suitable stack $u.\pi$.
- Then $t \star u.\pi \notin \perp\!\!\!\perp$, $u \not\Vdash \mathbb{X}$ and hence $u \star \pi_0 \in \perp\!\!\!\perp$.
- Taking $m \geq n' + 1$, $\langle P_m, \ell_m\rangle \in W'_{u_0.\pi_0}$, which leads a contradiction.

□

Completeness (cont.)

Proof.

...

- Define $P_\omega := \bigcup_{i \in \mathbb{N}} P_i$ and $\perp\!\!\!\perp^c := \bigcup_{p \in P_\omega} \mathbf{th}(p)$
- $u_0 \not\Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ since $t_0 \Vdash (\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}$.
- Thus $u_0 \star t.\pi_0 \in \mathbf{th}(p)$ for some $t \Vdash \neg\mathbb{X}$, $p \in P_n$ and $n \in \mathbb{N}$.
- Pick $n' \geq n$ s.t. $\phi(n') = t$. We have $p \in P_{n'}$ because $P_{n'} \supseteq P_n$.
- By definition, $P_{n'+1} = P_n \cup \{\phi(n') \star u.\pi\}$ for a suitable stack $u.\pi$.
- Then $t \star u.\pi \notin \perp\!\!\!\perp$, $u \not\Vdash \mathbb{X}$ and hence $u \star \pi_0 \in \perp\!\!\!\perp$.
- Taking $m \geq n' + 1$, $\langle P_m, \ell_m \rangle \in W'_{u_0.\pi_0}$, which leads a contradiction.

□

Completeness (cont.)

Proof.

...

- Define $P_\omega := \bigcup_{i\in\mathbb{N}} P_i$ and $\perp\!\!\!\perp^c := \bigcup_{p\in P_\omega} \mathbf{th}(p)$
- $u_0 \not\Vdash \neg\mathbb{X} \Rightarrow \mathbb{X}$ since $t_0 \Vdash (\neg\mathbb{X} \Rightarrow \mathbb{X}) \Rightarrow \mathbb{X}$.
- Thus $u_0 \star t.\pi_0 \in \mathbf{th}(p)$ for some $t \Vdash \neg\mathbb{X}$, $p \in P_n$ and $n \in \mathbb{N}$.
- Pick $n' \geq n$ s.t. $\phi(n') = t$. We have $p \in P_{n'}$ because $P_{n'} \supseteq P_n$.
- By definition, $P_{n'+1} = P_n \cup \{\phi(n') \star u.\pi\}$ for a suitable stack $u.\pi$.
- Then $t \star u.\pi \notin \perp\!\!\!\perp$, $u \not\Vdash \mathbb{X}$ and hence $u \star \pi_0 \in \perp\!\!\!\perp$.
- Taking $m \geq n' + 1$, $\langle P_m, \ell_m\rangle \in W'_{u_0.\pi_0}$, which leads a contradiction.

$\square$

conclusion

We have characterized the realizers of Peirce's Law, whenever the evaluation is compatible with substitutive constants, as the *uniform winning strategies* of a game $\mathbb{G}_0$.

The goal for $\exists$ in $\mathbb{G}_0$ is to put on execution a term u yet played by the opponent and restore the initial stack.

The uniformity means that, for each realizer, all the plays have the same length and finishes restoring the initial stack on the p-th *forall* move.

Conclusion (cont.)

We have characterized the realizers of Peirce's Law as the winning strategies of a game $\mathbb{G}_1$.

The goal for $\exists$ in $\mathbb{G}_1$ is to certify that one term u played by the opponent arrives at any time on execution, together with the initial stack π_0.

Here the uniformity is broken and $\exists$ can restore the stack of some moves before they are played.