Simulating modified realisability and A-translation with Gödel's Dialectica interpretation

Trifon Trifonov

Ludwig Maximilian Universität, München

ASL'08
Bern, 4 July 2008

Gödel's T

Types

$$\sigma, \rho \quad ::= \quad \mathrm{N} \mid \mathrm{B} \mid \sigma \Rightarrow \rho \mid \sigma \times \rho$$

Terms

Gödel's T

Types

$$\sigma, \rho \quad ::= \quad \mathrm{N} \mid \mathrm{B} \mid \sigma \Rightarrow \rho \mid \sigma \times \rho$$

Terms

$$t, s \quad ::= \quad x^{\rho} \mid ts \mid \lambda_x t \mid \langle t, s \rangle \mid t_{\llcorner} \mid t_{\lrcorner} \mid$$

Gödel's T

Types

$$\sigma, \rho \quad ::= \quad \mathrm{N} \mid \mathrm{B} \mid \sigma \Rightarrow \rho \mid \sigma \times \rho$$

Terms

$$\begin{aligned} t, s \quad ::= \quad & x^{\rho} \mid ts \mid \lambda_x t \mid \langle t, s \rangle \mid t_{\llcorner} \mid t_{\lrcorner} \mid \\ & \mathrm{tt}^{\mathrm{B}} \mid \mathrm{ff}^{\mathrm{B}} \mid 0^{\mathrm{N}} \mid \mathrm{S}^{\mathrm{N} \Rightarrow \mathrm{N}} \end{aligned}$$

Gödel's T

Types

$$\sigma, \rho \quad ::= \quad \mathrm{N} \mid \mathrm{B} \mid \sigma \Rightarrow \rho \mid \sigma \times \rho$$

Terms

$$\begin{aligned} t, s \quad ::= \quad & x^\rho \mid ts \mid \lambda_x t \mid \langle t, s\rangle \mid t_\llcorner \mid t_\lrcorner \mid \\ & \mathrm{tt}^{\mathrm{B}} \mid \mathrm{ff}^{\mathrm{B}} \mid 0^{\mathrm{N}} \mid \mathrm{S}^{\mathrm{N}\Rightarrow\mathrm{N}} \\ & \textbf{if } b^{\mathrm{B}} \textbf{ then } s^\tau \textbf{ else } t^\tau \mid \end{aligned}$$

Gödel's T

Types

$$\sigma, \rho \quad ::= \quad \mathrm{N} \mid \mathrm{B} \mid \sigma \Rightarrow \rho \mid \sigma \times \rho$$

Terms

$$\begin{aligned} t, s \quad ::= \quad & x^\rho \mid ts \mid \lambda_x t \mid \langle t, s \rangle \mid t_\llcorner \mid t_\lrcorner \mid \\ & \mathsf{tt}^\mathrm{B} \mid \mathsf{ff}^\mathrm{B} \mid 0^\mathrm{N} \mid \mathsf{S}^{\mathrm{N} \Rightarrow \mathrm{N}} \\ & \textbf{if } b^\mathrm{B} \textbf{ then } s^\tau \textbf{ else } t^\tau \mid \\ & \mathcal{R}^\tau_\mathrm{N} : \mathrm{N} \Rightarrow \tau \Rightarrow (\mathrm{N} \Rightarrow \tau \Rightarrow \tau) \Rightarrow \tau \end{aligned}$$

Notation shortcuts

We will use a special nulltype symbol ε and stipulate:

$$
\begin{array}{rcl}
\rho \times \varepsilon \rightsquigarrow \rho, & t_{\llcorner} \rightsquigarrow t, & \langle t, \varepsilon \rangle \rightsquigarrow t \\
\varepsilon \times \rho \rightsquigarrow \rho, & t_{\lrcorner} \rightsquigarrow t, & \langle \varepsilon, t \rangle \rightsquigarrow t \\
\rho \Rightarrow \varepsilon \rightsquigarrow \varepsilon, & \lambda_x \varepsilon \rightsquigarrow \varepsilon, & \varepsilon t \rightsquigarrow \varepsilon \\
\varepsilon \Rightarrow \rho \rightsquigarrow \rho, & \lambda_{x^\varepsilon} t \rightsquigarrow t, & t\varepsilon \rightsquigarrow t \\
 & \forall_{x^\varepsilon} A \rightsquigarrow A, & M\varepsilon \rightsquigarrow M
\end{array}
$$

We could have used a `unit` type, but then the equalities above become explicit isomorphisms.

Negative Arithmetic (NA^ω)

We consider the negative fragment of Heyting Arithmetic.

$$A, B \quad ::= \quad P(\vec{t}) \mid \mathrm{at}(b^{\mathbb{B}}) \mid A \to B \mid A \wedge B \mid \forall_x A \mid \exists_x A$$

We obtain HA^ω by adding the strong existential $\exists$.

Negative Arithmetic (NA^{ω})

We consider the negative fragment of Heyting Arithmetic.

$$
\begin{aligned}
A, B \quad &::= \quad P(\vec{t}) \mid \mathrm{at}(b^{\mathbb{B}}) \mid A \to B \mid A \wedge B \mid \forall_x A \mid \exists_x A \\
\neg A \quad &::= \quad A \to \bot
\end{aligned}
$$

We obtain HA^{ω} by adding the strong existential $\exists$.

Negative Arithmetic (NA^ω)

We consider the negative fragment of Heyting Arithmetic.

$$
\begin{aligned}
A, B &::= P(\vec{t}) \mid \mathrm{at}(b^{\mathbb{B}}) \mid A \to B \mid A \wedge B \mid \forall_x A \mid \exists_x A \\
\neg A &::= A \to \bot \\
\tilde{\exists}_x A &::= \neg \forall_x \neg A
\end{aligned}
$$

We obtain HA^ω by adding the strong existential $\exists$.

Heyting Arithmetic (HA^ω)

We consider the negative fragment of Heyting Arithmetic.

$$\begin{array}{rcl} A, B & ::= & P(\vec{t}) \mid \mathrm{at}(b^{\mathbb{B}}) \mid A \to B \mid A \wedge B \mid \forall_x A \mid \exists_x A \\ \neg A & ::= & A \to \bot \\ \tilde{\exists}_x A & ::= & \neg \forall_x \neg A \end{array}$$

We obtain HA^ω by adding the strong existential $\exists$.

Boolean falsity

Using a general predicate variable $\bot$ we work in a minimal logic setting. We denote the system as HA_0^ω.

However, if we use *decidable falsity* $\mathrm{F} := \mathrm{at}(\mathrm{ff})$, we are able to prove by induction on the definition of formulas

Lemma (ex falso quodlibet)

$\mathrm{HA}^\omega \vdash \mathrm{F} \to A$

Lemma (stability)

$\mathrm{NA}^\omega \vdash ((A \to \mathrm{F}) \to \mathrm{F}) \to A$

if A contains no predicate variables.

Boolean falsity

Using a general predicate variable $\bot$ we work in a minimal logic setting. We denote the system as HA_0^ω.
However, if we use *decidable falsity* $\mathrm{F} := \mathrm{at}(\mathrm{ff})$, we are able to prove by induction on the definition of formulas

Lemma (ex falso quodlibet)

$\mathrm{HA}^\omega \vdash \mathrm{F} \to A$

Lemma (stability)

$\mathrm{NA}^\omega \vdash ((A \to \mathrm{F}) \to \mathrm{F}) \to A$

if A contains no predicate variables.

Boolean falsity

Using a general predicate variable $\bot$ we work in a minimal logic setting. We denote the system as HA_0^ω.
However, if we use *decidable falsity* $\mathrm{F} := \mathrm{at}(\mathrm{ff})$, we are able to prove by induction on the definition of formulas

Lemma (ex falso quodlibet)

$\mathrm{HA}^\omega \vdash \mathrm{F} \to A$

Lemma (stability)

$\mathrm{NA}^\omega \vdash ((A \to \mathrm{F}) \to \mathrm{F}) \to A$

if A contains no predicate variables.

Boolean falsity

Using a general predicate variable $\bot$ we work in a minimal logic setting. We denote the system as HA_0^ω.
However, if we use *decidable falsity* $\mathrm{F} := \mathrm{at}(\mathrm{ff})$, we are able to prove by induction on the definition of formulas

Lemma (ex falso quodlibet)

$\mathrm{HA}^\omega \vdash \mathrm{F} \to A$

Lemma (stability)

$\mathrm{NA}^\omega \vdash ((A \to \mathrm{F}) \to \mathrm{F}) \to A$

if A contains no predicate variables.

Boolean falsity

Using a general predicate variable $\bot$ we work in a minimal logic setting. We denote the system as HA_0^ω.
However, if we use *decidable falsity* $\mathrm{F} := \mathrm{at}(\mathrm{ff})$, we are able to prove by induction on the definition of formulas

Lemma (ex falso quodlibet)

$\mathrm{HA}^\omega \vdash \mathrm{F} \to A$

Lemma (stability)

$\mathrm{NA}^\omega \vdash ((A \to \mathrm{F}) \to \mathrm{F}) \to A$

if A contains no predicate variables.

Realisability and Dialectica

Realisability

- Formulas A are problems
- They ask for solutions of type $\tau^\circ(A)$
- Translations $r \mathrm{mr} A$ verify if r is a solution to A

Dialectica

- Formulas A are problems
- They ask for solutions of type $\tau^+(A)$
- that are challenged by terms of type $\tau^-(A)$
- Translations $|A|^r_s$ verify if r solves A for a challenge s

Realisability and Dialectica

Realisability

- Formulas A are problems
- They ask for solutions of type $\tau^\circ(A)$
- Translations $r \mathrm{\ mr\ } A$ verify if r is a solution to A

Dialectica

- Formulas A are problems
- They ask for solutions of type $\tau^+(A)$
- that are challenged by terms of type $\tau^-(A)$
- Translations $|A|^r_s$ verify if r solves A for a challenge s

Computational type

C	$\tau^\circ(C)$	$\tau^+(C)$	$\tau^-(C)$
$P(\vec{t})$	τ°	τ^+	τ^-
$\mathtt{at}(b)$	ε	ε	ε
$A \wedge B$	$\tau^\circ(A) \times \tau^\circ(B)$	$\tau^+(A) \times \tau^+(B)$	$\tau^-(A) \times \tau^-(B)$
$A \to B$	$\tau^\circ(A) \Rightarrow \tau^\circ(B)$	$(\tau^+(A) \Rightarrow \tau^+(B)) \times (\tau^+(A) \Rightarrow \tau^-(B) \Rightarrow \tau^-(A))$	$\tau^+(A) \times \tau^-(B)$
$\forall_{x^\rho} A$	$\rho \Rightarrow \tau^\circ(A)$	$\rho \Rightarrow \tau^\circ(A)$	$\rho \times \tau^-(A)$
$\exists_{x^\rho} A$	$\rho \times \tau^\circ(A)$	$\rho \times \tau^+(A)$	$\tau^-(A)$

- Realisability needs predicates with computational content.
- Dialectica obtains content from negative occurrences of $\forall$

Computational type

C	$\tau^\circ(C)$	$\tau^+(C)$	$\tau^-(C)$
$P(\vec{t})$	τ°	τ^+	τ^-
$\mathtt{at}(b)$	ε	ε	ε
$A \wedge B$	$\tau^\circ(A) \times \tau^\circ(B)$	$\tau^+(A) \times \tau^+(B)$	$\tau^-(A) \times \tau^-(B)$
$A \to B$	$\tau^\circ(A) \Rightarrow \tau^\circ(B)$	$(\tau^+(A) \Rightarrow \tau^+(B)) \times$ $(\tau^+(A) \Rightarrow \tau^-(B) \Rightarrow \tau^-(A))$	$\tau^+(A) \times \tau^-(B)$
$\forall_{x^\rho} A$	$\rho \Rightarrow \tau^\circ(A)$	$\rho \Rightarrow \tau^\circ(A)$	$\rho \times \tau^-(A)$
$\exists_{x^\rho} A$	$\rho \times \tau^\circ(A)$	$\rho \times \tau^+(A)$	$\tau^-(A)$

- Realisability needs predicates with computational content.
- Dialectica obtains content from negative occurrences of $\forall$

Computational type

C	$\tau^\circ(C)$	$\tau^+(C)$	$\tau^-(C)$
$P(\vec{t})$	τ°	τ^+	τ^-
$\mathtt{at}(b)$	ε	ε	ε
$A \wedge B$	$\tau^\circ(A) \times \tau^\circ(B)$	$\tau^+(A) \times \tau^+(B)$	$\tau^-(A) \times \tau^-(B)$
$A \to B$	$\tau^\circ(A) \Rightarrow \tau^\circ(B)$	$(\tau^+(A) \Rightarrow \tau^+(B)) \times$ $(\tau^+(A) \Rightarrow \tau^-(B) \Rightarrow \tau^-(A))$	$\tau^+(A) \times \tau^-(B)$
$\forall_{x^\rho} A$	$\rho \Rightarrow \tau^\circ(A)$	$\rho \Rightarrow \tau^\circ(A)$	$\rho \times \tau^-(A)$
$\exists_{x^\rho} A$	$\rho \times \tau^\circ(A)$	$\rho \times \tau^+(A)$	$\tau^-(A)$

- Realisability needs predicates with computational content.
- Dialectica obtains content from negative occurrences of $\forall$

Translation

C	$r \text{ mr } C$	$\lvert C\rvert^r_s$
$P(\vec{t})$	$P^\circ(r^{\tau^\circ}, \vec{t})$	$P^*(r^{\tau^+}, s^{\tau^-}, \vec{t})$
$\mathtt{at}(b)$	$\mathtt{at}(b)$	$\mathtt{at}(b)$
$A \wedge B$	$r_\llcorner \text{ mr } A \wedge r_\lrcorner \text{ mr } B$	$\lvert A\rvert^{r_\llcorner}_{s_\llcorner} \wedge \lvert B\rvert^{r_\lrcorner}_{s_\lrcorner}$
$A \to B$	$\forall_x (x \text{ mr } A \to rx \text{ mr } B)$	$\lvert A\rvert^{s_\llcorner}_{(r_\lrcorner)(s_\llcorner)(s_\lrcorner)} \to \lvert A\rvert^{(r_\llcorner)(s_\llcorner)}_{s_\lrcorner}$
$\forall_x A(x)$	$\forall_x rx \text{ mr } A(x)$	$\lvert A(s_\llcorner)\rvert^{r(s_\llcorner)}_{s_\lrcorner}$
$\exists_x A(x)$	$r_\lrcorner \text{ mr } A(r_\llcorner)$	$\lvert A(r_\llcorner)\rvert^{r_\lrcorner}_{s}$

The Dialectica translation is always a quantifier-free formula.

Translation

C	$r \text{ mr } C$	$\lvert C\rvert^r_s$
$P(\vec{t})$	$P^\circ(r^{\tau^\circ}, \vec{t})$	$P^*(r^{\tau^+}, s^{\tau^-}, \vec{t})$
$\mathtt{at}(b)$	$\mathtt{at}(b)$	$\mathtt{at}(b)$
$A \wedge B$	$r_\llcorner \text{ mr } A \wedge r_\lrcorner \text{ mr } B$	$\lvert A\rvert^{r_\llcorner}_{s_\llcorner} \wedge \lvert B\rvert^{r_\lrcorner}_{s_\lrcorner}$
$A \to B$	$\forall_x (x \text{ mr } A \to rx \text{ mr } B)$	$\lvert A\rvert^{s_\llcorner}_{(r_\lrcorner)(s_\llcorner)(s_\lrcorner)} \to \lvert A\rvert^{(r_\llcorner)(s_\llcorner)}_{s_\lrcorner}$
$\forall_x A(x)$	$\forall_x rx \text{ mr } A(x)$	$\lvert A(s_\llcorner)\rvert^{r(s_\llcorner)}_{s_\lrcorner}$
$\exists_x A(x)$	$r_\lrcorner \text{ mr } A(r_\llcorner)$	$\lvert A(r_\llcorner)\rvert^{r_\lrcorner}_{s}$

The Dialectica translation is always a quantifier-free formula.

Soundness

Theorem (Realisability)

Let $\mathcal{P}$ be a proof of A from assumptions $u_i : C_i$. Then we can prove $[\![\mathcal{P}]\!]^\circ \ \mathrm{mr}\ A$ from assumptions $x_{u_i}\ \mathrm{mr}\ C_i$.

Theorem (Dialectica)

Let $\mathcal{P}$ be a proof of A from assumptions $u_i : C_i$. Then we can prove $|A|_{y_A}^{[\![\mathcal{P}]\!]^+}$ from assumptions $|C_i|_{[\![\mathcal{P}]\!]_i^-}^{x_{u_i}}$ and $y_A \notin \mathrm{FV}([\![\mathcal{P}]\!]^+)$.

Soundness

Theorem (Realisability)

Let $\mathcal{P}$ be a proof of A from assumptions $u_i : C_i$. Then we can prove $[\![\mathcal{P}]\!]^\circ \mathrm{\ mr\ } A$ from assumptions $x_{u_i} \mathrm{\ mr\ } C_i$.

Theorem (Dialectica)

Let $\mathcal{P}$ be a proof of A from assumptions $u_i : C_i$. Then we can prove $|A|_{y_A}^{[\![\mathcal{P}]\!]^+}$ from assumptions $|C_i|_{[\![\mathcal{P}]\!]_i^-}^{x_{u_i}}$ and $y_A \notin \mathrm{FV}([\![\mathcal{P}]\!]^+)$.

Dialectica and contractions

- Consider a binary rule:

$$\dfrac{\begin{array}{c} u : C \\ \mid M \\ B_1 \end{array} \qquad \begin{array}{c} u : C \\ \mid N \\ B_2 \end{array}}{A} \quad \rightsquigarrow \quad \dfrac{\begin{array}{c} u : |C|^x_{[\![M]\!]^-} \\ \mid \\ |B_1|^{[\![M]\!]^+}_y \end{array} \qquad \begin{array}{c} u : |C|^x_{[\![N]\!]^-} \\ \mid \\ |B_2|^{[\![N]\!]^+}_z \end{array}}{}$$

- Solution — case distinction on decidable kernel.

$$[\![P]\!]^- := [\![M]\!]^- \overset{u}{\bowtie} [\![N]\!]^- := \textbf{if } |C|^x_{[\![M]\!]^-} \textbf{ then } [\![N]\!]^- \textbf{ else } [\![M]\!]^-.$$

- Other approaches — finite set of solutions (Diller-Nahm, 1974), monotone Dialectica (Kohlenbach, 1993)

Dialectica and contractions

- Consider a binary rule:

$$\dfrac{\begin{array}{c} u : C \\ \mid M \\ B_1 \end{array} \qquad \begin{array}{c} u : C \\ \mid N \\ B_2 \end{array}}{A} \quad \rightsquigarrow \quad \dfrac{\begin{array}{c} u : |C|^{x}_{[\![P]\!]^-} \\ \mid \\ |B_1|^{[\![M]\!]^+}_{y} \end{array} \qquad \begin{array}{c} u : |C|^{x}_{[\![P]\!]^-} \\ \mid \\ |B_2|^{[\![N]\!]^+}_{z} \end{array}}{|A|^{[\![P]\!]^+}_{v}}$$

- Solution — case distinction on decidable kernel.

$$[\![P]\!]^- := [\![M]\!]^- \overset{u}{\bowtie} [\![N]\!]^- := \textbf{if } |C|^{x}_{[\![M]\!]^-} \textbf{ then } [\![N]\!]^- \textbf{ else } [\![M]\!]^-.$$

- Other approaches — finite set of solutions (Diller-Nahm, 1974), monotone Dialectica (Kohlenbach, 1993)

Dialectica and contractions

- Consider a binary rule:

$$\dfrac{\begin{array}{c} u : C \\ \mid M \\ B_1 \end{array} \qquad \begin{array}{c} u : C \\ \mid N \\ B_2 \end{array}}{A} \quad \rightsquigarrow \quad \dfrac{\begin{array}{c} u : |C|^{x}_{[\![P]\!]^-} \\ \mid \\ |B_1|^{[\![M]\!]^+}_{y} \end{array} \qquad \begin{array}{c} u : |C|^{x}_{[\![P]\!]^-} \\ \mid \\ |B_2|^{[\![N]\!]^+}_{z} \end{array}}{|A|^{[\![P]\!]^+}_{v}}$$

- Solution — case distinction on decidable kernel.

$$[\![P]\!]^- := [\![M]\!]^- \overset{u}{\bowtie} [\![N]\!]^- := \textbf{if } |C|^{x}_{[\![M]\!]^-} \textbf{ then } [\![N]\!]^- \textbf{ else } [\![M]\!]^- .$$

- Other approaches — finite set of solutions (Diller-Nahm, 1974), monotone Dialectica (Kohlenbach, 1993)

Dialectica and contractions

- Consider a binary rule:

$$\dfrac{\begin{array}{c} u : C \\ \mid M \\ B_1 \end{array} \qquad \begin{array}{c} u : C \\ \mid N \\ B_2 \end{array}}{A} \quad \rightsquigarrow \quad \dfrac{\begin{array}{c} u : |C|^{x}_{\color{red}{[\![P]\!]^-}} \\ \mid \\ |B_1|^{[\![M]\!]^+}_{y} \end{array} \qquad \begin{array}{c} u : |C|^{x}_{\color{red}{[\![P]\!]^-}} \\ \mid \\ |B_2|^{[\![N]\!]^+}_{z} \end{array}}{|A|^{[\![P]\!]^+}_{v}}$$

- Solution — case distinction on decidable kernel.

$$[\![P]\!]^- := [\![M]\!]^- \overset{u}{\bowtie} [\![N]\!]^- := \textbf{if } |C|^{x}_{[\![M]\!]^-} \textbf{ then } [\![N]\!]^- \textbf{ else } [\![M]\!]^- .$$

- Other approaches — finite set of solutions (Diller-Nahm, 1974), monotone Dialectica (Kohlenbach, 1993)

Uniform quantifiers

Consider the formula

$$\forall_n \forall_m \big(n = 2m \to \exists_k (n^2 = 4k)\big)$$

A realiser could be $k = m^2$ or $k = \mathrm{Quot}(n^2, 4)$. Hence, a quantified variable is not necessarily computationally relevant. If not, we quantify it *uniformly*: $\forall^{\mathsf{U}}_x A$ or $\exists^{\mathsf{U}}_x A$

- Introduced for realisability by Berger (2005)
- Adapted to Dialectica by Hernest (2007)

Uniform quantifiers

Consider the formula

$$\forall_n \forall_m \big(n = 2m \to \exists_k (n^2 = 4k)\big)$$

A realiser could be $k = m^2$ or $k = \mathrm{Quot}(n^2, 4)$. Hence, a quantified variable is not necessarily computationally relevant. If not, we quantify it *uniformly*: $\forall_x^{\mathrm{U}} A$ or $\exists_x^{\mathrm{U}} A$

- Introduced for realisability by Berger (2005)
- Adapted to Dialectica by Hernest (2007)

Uniform quantifiers

Consider the formula

$$\forall_n \forall_m \big(n = 2m \to \exists_k (n^2 = 4k)\big)$$

A realiser could be $k = m^2$ or $k = \mathrm{Quot}(n^2, 4)$. Hence, a quantified variable is not necessarily computationally relevant. If not, we quantify it *uniformly*: $\forall^{U}_{x} A$ or $\exists^{U}_{x} A$

- Introduced for realisability by Berger (2005)
- Adapted to Dialectica by Hernest (2007)

Uniform quantifiers

Consider the formula

$$\forall_n \forall_m (n = 2m \to \exists_k (n^2 = 4k))$$

A realiser could be $k = m^2$ or $k = \mathrm{Quot}(n^2, 4)$. Hence, a quantified variable is not necessarily computationally relevant. If not, we quantify it *uniformly*: $\forall^{\mathrm{U}}_x A$ or $\exists^{\mathrm{U}}_x A$

- Introduced for realisability by Berger (2005)
- Adapted to Dialectica by Hernest (2007)

Uniform quantifiers

Consider the formula

$$\forall_n \forall_m \big(n = 2m \to \exists_k (n^2 = 4k)\big)$$

A realiser could be $k = m^2$ or $k = \mathrm{Quot}(n^2, 4)$. Hence, a quantified variable is not necessarily computationally relevant. If not, we quantify it *uniformly*: $\forall_x^{\mathsf{U}} A$ or $\exists_x^{\mathsf{U}} A$

- Introduced for realisability by Berger (2005)
- Adapted to Dialectica by Hernest (2007)

Uniform quantifiers in realisability

- $$\dfrac{\begin{array}{c}\mid M\\ A\end{array}}{\forall^{\mathsf{U}}_x A}\; x \notin \mathrm{FV}(\llbracket M \rrbracket^\circ)$$
- $\exists^{\mathsf{U},-} : \exists^{\mathsf{U}}_x A \to \forall^{\mathsf{U}}_x (A \to B) \to B$
- $\tau^\circ(\forall^{\mathsf{U}}_x A) = \tau^\circ(\exists^{\mathsf{U}}_x A) = \tau^\circ(A)$
- $r \;\mathrm{mr}\; \forall^{\mathsf{U}}_x A = \forall_x r \;\mathrm{mr}\; A$
- $r \;\mathrm{mr}\; \exists^{\mathsf{U}}_x A = \exists_x r \;\mathrm{mr}\; A$
- $\llbracket \lambda^{\mathsf{U}}_x M \rrbracket^\circ = \llbracket M \rrbracket^\circ$

Uniform quantifiers in realisability

- $$\frac{\begin{array}{c} \mid M \\ A \end{array}}{\forall^{\mathsf{U}}_x A}\; x \notin \mathrm{FV}(\llbracket M \rrbracket^\circ)$$
- $\exists^{\mathsf{U},-} : \exists^{\mathsf{U}}_x A \to \forall^{\mathsf{U}}_x (A \to B) \to B$
- $\tau^\circ(\forall^{\mathsf{U}}_x A) = \tau^\circ(\exists^{\mathsf{U}}_x A) = \tau^\circ(A)$
- $r \;\mathrm{mr}\; \forall^{\mathsf{U}}_x A = \forall_x r \;\mathrm{mr}\; A$
- $r \;\mathrm{mr}\; \exists^{\mathsf{U}}_x A = \exists_x r \;\mathrm{mr}\; A$
- $\llbracket \lambda^{\mathsf{U}}_x M \rrbracket^\circ = \llbracket M \rrbracket^\circ$

Uniform quantifiers in realisability

- $$\frac{\begin{array}{c} \mid M \\ A \end{array}}{\forall^{\mathsf{U}}_x A}\ x \notin \mathrm{FV}(\llbracket M \rrbracket^\circ)$$
- $\exists^{\mathsf{U},-} : \exists^{\mathsf{U}}_x A \to \forall^{\mathsf{U}}_x (A \to B) \to B$
- $\tau^\circ(\forall^{\mathsf{U}}_x A) = \tau^\circ(\exists^{\mathsf{U}}_x A) = \tau^\circ(A)$
- $r \ \mathrm{mr}\ \forall^{\mathsf{U}}_x A = \forall_x r \ \mathrm{mr}\ A$
- $r \ \mathrm{mr}\ \exists^{\mathsf{U}}_x A = \exists_x r \ \mathrm{mr}\ A$
- $\llbracket \lambda^{\mathsf{U}}_x M \rrbracket^\circ = \llbracket M \rrbracket^\circ$

Uniform quantifiers in realisability

- $$\frac{\begin{array}{c} \mid M \\ A \end{array}}{\forall^{\mathsf{U}}_x A}\; x \notin \mathrm{FV}([\![M]\!]^\circ)$$
- $\exists^{\mathsf{U},-} : \exists^{\mathsf{U}}_x A \to \forall^{\mathsf{U}}_x (A \to B) \to B$
- $\tau^\circ(\forall^{\mathsf{U}}_x A) = \tau^\circ(\exists^{\mathsf{U}}_x A) = \tau^\circ(A)$
- $r \ \mathrm{mr}\ \forall^{\mathsf{U}}_x A = \forall_x r \ \mathrm{mr}\ A$
- $r \ \mathrm{mr}\ \exists^{\mathsf{U}}_x A = \exists_x r \ \mathrm{mr}\ A$
- $[\![\lambda^{\mathsf{U}}_x M]\!]^\circ = [\![M]\!]^\circ$

Uniform quantifiers in realisability

- $$\frac{\begin{array}{c} \mid M \\ A \end{array}}{\forall^{\mathsf{U}}_x A}\ x \notin \mathrm{FV}(\llbracket M \rrbracket^\circ)$$
- $\exists^{\mathsf{U},-} : \exists^{\mathsf{U}}_x A \to \forall^{\mathsf{U}}_x (A \to B) \to B$
- $\tau^\circ(\forall^{\mathsf{U}}_x A) = \tau^\circ(\exists^{\mathsf{U}}_x A) = \tau^\circ(A)$
- $r \ \mathrm{mr}\ \forall^{\mathsf{U}}_x A = \forall_x r \ \mathrm{mr}\ A$
- $r \ \mathrm{mr}\ \exists^{\mathsf{U}}_x A = \exists_x r \ \mathrm{mr}\ A$
- $\llbracket \lambda^{\mathsf{U}}_x M \rrbracket^\circ = \llbracket M \rrbracket^\circ$

Uniform quantifiers in realisability

- $$\dfrac{\begin{array}{c} \mid M \\ A \end{array}}{\forall^{\mathsf{U}}_x A}\; x \notin \mathrm{FV}(\llbracket M \rrbracket^\circ)$$
- $\exists^{\mathsf{U},-} : \exists^{\mathsf{U}}_x A \to \forall^{\mathsf{U}}_x (A \to B) \to B$
- $\tau^\circ(\forall^{\mathsf{U}}_x A) = \tau^\circ(\exists^{\mathsf{U}}_x A) = \tau^\circ(A)$
- $r \;\mathrm{mr}\; \forall^{\mathsf{U}}_x A = \forall_x r \;\mathrm{mr}\; A$
- $r \;\mathrm{mr}\; \exists^{\mathsf{U}}_x A = \exists_x r \;\mathrm{mr}\; A$
- $\llbracket \lambda^{\mathsf{U}}_x M \rrbracket^\circ = \llbracket M \rrbracket^\circ$

Uniform quantifiers in Dialectica

- $$\frac{\begin{array}{c}\mid M\\ A\end{array}}{\forall^{\mathsf{U}}_x A}\; x \notin \mathrm{FV}(\llbracket M\rrbracket^+) \cup \mathrm{FV}(\llbracket M\rrbracket^-_i)$$
- $\exists^{\mathsf{U},-} : \exists^{\mathsf{U}}_x A \to \forall^{\mathsf{U}}_x (A \to B) \to B$
- $\tau^{\pm}(\forall^{\mathsf{U}}_x A) = \tau^{\pm}(\exists^{\mathsf{U}}_x A) = \tau^{\pm}(A)$
- $\left|\forall^{\mathsf{U}}_x A\right|^r_s = \forall_x \left|A\right|^r_s$
- $\left|\exists^{\mathsf{U}}_x A\right|^r_s = \exists_x \left|A\right|^r_s$
- $|A|^x_y$ is no longer quantifier-free!
- Forbid uniform quantifiers in relevant contractions
- $\llbracket\lambda^{\mathsf{U}}_x M\rrbracket^+ = \llbracket M\rrbracket^+$
- Extends better to monotone Dialectica

Uniform quantifiers in Dialectica

- $$\dfrac{\begin{array}{c} \mid M \\ A \end{array}}{\forall^{\mathsf{U}}_x A}\; x \notin \mathrm{FV}([\![M]\!]^+) \cup \mathrm{FV}([\![M]\!]^-_i)$$
- $\exists^{\mathsf{U},-} : \exists^{\mathsf{U}}_x A \to \forall^{\mathsf{U}}_x (A \to B) \to B$
- $\tau^{\pm}(\forall^{\mathsf{U}}_x A) = \tau^{\pm}(\exists^{\mathsf{U}}_x A) = \tau^{\pm}(A)$
- $\left|\forall^{\mathsf{U}}_x A\right|^r_s = \forall_x \left|A\right|^r_s$
- $\left|\exists^{\mathsf{U}}_x A\right|^r_s = \exists_x \left|A\right|^r_s$
- $|A|^x_y$ is no longer quantifier-free!
- Forbid uniform quantifiers in relevant contractions
- $[\![\lambda^{\mathsf{U}}_x M]\!]^+ = [\![M]\!]^+$
- Extends better to monotone Dialectica

Uniform quantifiers in Dialectica

- $$\frac{\begin{array}{c} \mid M \\ A \end{array}}{\forall^{\mathsf{U}}_x A}\; x \notin \mathrm{FV}(\llbracket M \rrbracket^+) \cup \mathrm{FV}(\llbracket M \rrbracket^-_i)$$
- $\exists^{\mathsf{U},-} : \exists^{\mathsf{U}}_x A \to \forall^{\mathsf{U}}_x (A \to B) \to B$
- $\tau^{\pm}(\forall^{\mathsf{U}}_x A) = \tau^{\pm}(\exists^{\mathsf{U}}_x A) = \tau^{\pm}(A)$
- $\left|\forall^{\mathsf{U}}_x A\right|^r_s = \forall_x \left|A\right|^r_s$
- $\left|\exists^{\mathsf{U}}_x A\right|^r_s = \exists_x \left|A\right|^r_s$
- $|A|^x_y$ is no longer quantifier-free!
- Forbid uniform quantifiers in relevant contractions
- $\llbracket \lambda^{\mathsf{U}}_x M \rrbracket^+ = \llbracket M \rrbracket^+$
- Extends better to monotone Dialectica

Uniform quantifiers in Dialectica

- $$\frac{\begin{array}{c} \mid M \\ A \end{array}}{\forall^{\mathsf{U}}_x A}\ x \notin \mathrm{FV}(\llbracket M \rrbracket^+) \cup \mathrm{FV}(\llbracket M \rrbracket^-_i)$$
- $\exists^{\mathsf{U},-} : \exists^{\mathsf{U}}_x A \to \forall^{\mathsf{U}}_x (A \to B) \to B$
- $\tau^{\pm}(\forall^{\mathsf{U}}_x A) = \tau^{\pm}(\exists^{\mathsf{U}}_x A) = \tau^{\pm}(A)$
- $\left|\forall^{\mathsf{U}}_x A\right|^r_s = \forall_x \left|A\right|^r_s$
- $\left|\exists^{\mathsf{U}}_x A\right|^r_s = \exists_x \left|A\right|^r_s$
- $|A|^x_y$ is no longer quantifier-free!
- Forbid uniform quantifiers in relevant contractions
- $\llbracket \lambda^{\mathsf{U}}_x M \rrbracket^+ = \llbracket M \rrbracket^+$
- Extends better to monotone Dialectica

Uniform quantifiers in Dialectica

- $$\dfrac{\begin{array}{c} \mid M \\ A \end{array}}{\forall^{\mathsf{U}}_x A}\; x \notin \mathrm{FV}(\llbracket M \rrbracket^{+}) \cup \mathrm{FV}(\llbracket M \rrbracket^{-}_{i})$$
- $\exists^{\mathsf{U},-} : \exists^{\mathsf{U}}_x A \to \forall^{\mathsf{U}}_x (A \to B) \to B$
- $\tau^{\pm}(\forall^{\mathsf{U}}_x A) = \tau^{\pm}(\exists^{\mathsf{U}}_x A) = \tau^{\pm}(A)$
- $\left|\forall^{\mathsf{U}}_x A\right|^r_s = \forall_x \left|A\right|^r_s$
- $\left|\exists^{\mathsf{U}}_x A\right|^r_s = \exists_x \left|A\right|^r_s$
- $|A|^x_y$ is no longer quantifier-free!
- Forbid uniform quantifiers in relevant contractions
- $\llbracket \lambda^{\mathsf{U}}_x M \rrbracket^{+} = \llbracket M \rrbracket^{+}$
- Extends better to monotone Dialectica

Uniform quantifiers in Dialectica

- $$\dfrac{\begin{array}{c} \mid M \\ A \end{array}}{\forall^{\mathrm{U}}_x A}\; x \notin \mathrm{FV}(\llbracket M \rrbracket^+) \cup \mathrm{FV}(\llbracket M \rrbracket^-_i)$$
- $\exists^{\mathrm{U},-} : \exists^{\mathrm{U}}_x A \to \forall^{\mathrm{U}}_x (A \to B) \to B$
- $\tau^{\pm}(\forall^{\mathrm{U}}_x A) = \tau^{\pm}(\exists^{\mathrm{U}}_x A) = \tau^{\pm}(A)$
- $\left|\forall^{\mathrm{U}}_x A\right|^r_s = \forall_x \left|A\right|^r_s$
- $\left|\exists^{\mathrm{U}}_x A\right|^r_s = \exists_x \left|A\right|^r_s$
- $|A|^x_y$ is no longer quantifier-free!
- Forbid uniform quantifiers in relevant contractions
- $\llbracket \lambda^{\mathrm{U}}_x M \rrbracket^+ = \llbracket M \rrbracket^+$
- Extends better to monotone Dialectica

Uniform quantifiers in Dialectica

- $$\dfrac{\begin{array}{c}\mid M\\ A\end{array}}{\forall^{\mathsf{U}}_x A}\; x \notin \mathrm{FV}(\llbracket M\rrbracket^{+}) \cup \mathrm{FV}(\llbracket M\rrbracket^{-}_{i})$$
- $\exists^{\mathsf{U},-} : \exists^{\mathsf{U}}_x A \to \forall^{\mathsf{U}}_x (A \to B) \to B$
- $\tau^{\pm}(\forall^{\mathsf{U}}_x A) = \tau^{\pm}(\exists^{\mathsf{U}}_x A) = \tau^{\pm}(A)$
- $\left|\forall^{\mathsf{U}}_x A\right|^r_s = \forall_x \left|A\right|^r_s$
- $\left|\exists^{\mathsf{U}}_x A\right|^r_s = \exists_x \left|A\right|^r_s$
- $|A|^x_y$ is no longer quantifier-free!
- Forbid uniform quantifiers in relevant contractions
- $\llbracket \lambda^{\mathsf{U}}_x M\rrbracket^{+} = \llbracket M\rrbracket^{+}$
- Extends better to monotone Dialectica

Uniform quantifiers in Dialectica

- $$\frac{\begin{array}{c} \mid M \\ A \end{array}}{\forall^{\mathsf{U}}_x A} \; x \notin \mathrm{FV}(\llbracket M \rrbracket^+) \cup \mathrm{FV}(\llbracket M \rrbracket^-_i)$$
- $\exists^{\mathsf{U},-} : \exists^{\mathsf{U}}_x A \to \forall^{\mathsf{U}}_x (A \to B) \to B$
- $\tau^{\pm}(\forall^{\mathsf{U}}_x A) = \tau^{\pm}(\exists^{\mathsf{U}}_x A) = \tau^{\pm}(A)$
- $\left|\forall^{\mathsf{U}}_x A\right|^r_s = \forall_x \, |A|^r_s$
- $\left|\exists^{\mathsf{U}}_x A\right|^r_s = \exists_x \, |A|^r_s$
- $|A|^x_y$ is no longer quantifier-free!
- Forbid uniform quantifiers in relevant contractions
- $\llbracket \lambda^{\mathsf{U}}_x M \rrbracket^+ = \llbracket M \rrbracket^+$
- Extends better to monotone Dialectica

Uniform quantifiers in Dialectica

- $$\dfrac{\begin{array}{c} \mid M \\ A \end{array}}{\forall^{\mathsf{U}}_x A}\; x \notin \mathrm{FV}(\llbracket M \rrbracket^{+}) \cup \mathrm{FV}(\llbracket M \rrbracket^{-}_{i})$$
- $\exists^{\mathsf{U},-} : \exists^{\mathsf{U}}_x A \to \forall^{\mathsf{U}}_x (A \to B) \to B$
- $\tau^{\pm}(\forall^{\mathsf{U}}_x A) = \tau^{\pm}(\exists^{\mathsf{U}}_x A) = \tau^{\pm}(A)$
- $\left|\forall^{\mathsf{U}}_x A\right|^r_s = \forall_x \left|A\right|^r_s$
- $\left|\exists^{\mathsf{U}}_x A\right|^r_s = \exists_x \left|A\right|^r_s$
- $|A|^x_y$ is no longer quantifier-free!
- Forbid uniform quantifiers in relevant contractions
- $\llbracket \lambda^{\mathsf{U}}_x M \rrbracket^{+} = \llbracket M \rrbracket^{+}$
- Extends better to monotone Dialectica

Oliva's counterexample

Let $P(n)$ be an undecidable predicate.

$$\forall_n \exists_{m>n} P(m) \to \forall_d \exists_{n_1,n_2} (n_1 + d < n_2 \wedge P(n_1) \wedge P(n_2))$$

- The proof uses the premise twice (for 0 and $n_1 + d$)
- Realisability extracts witness $\lambda_{f,d} \langle f0, f(f0 + d) \rangle$
- Dialectica tries to extract counterexamples for the premise...
- ...but needs decidability of P!
- marking n as uniform turns f into a constant

Oliva's counterexample

Let $P(n)$ be an undecidable predicate.

$$\forall_n \exists_{m>n} P(m) \to \forall_d \exists_{n_1,n_2} (n_1 + d < n_2 \wedge P(n_1) \wedge P(n_2))$$

- The proof uses the premise twice (for 0 and $n_1 + d$)
- Realisability extracts witness $\lambda_{f,d} \langle f0, f(f0 + d) \rangle$
- Dialectica tries to extract counterexamples for the premise…
- …but needs decidability of P!
- marking n as uniform turns f into a constant

Oliva's counterexample

Let $P(n)$ be an undecidable predicate.

$$\forall_n \exists_{m>n} P(m) \to \forall_d \exists_{n_1,n_2} (n_1 + d < n_2 \wedge P(n_1) \wedge P(n_2))$$

- The proof uses the premise twice (for 0 and $n_1 + d$)
- Realisability extracts witness $\lambda_{f,d} \langle f0, f(f0 + d) \rangle$
- Dialectica tries to extract counterexamples for the premise...
- ...but needs decidability of P!
- marking n as uniform turns f into a constant

Oliva's counterexample

Let $P(n)$ be an undecidable predicate.

$$\forall_n \exists_{m>n} P(m) \to \forall_d \exists_{n_1,n_2} (n_1 + d < n_2 \wedge P(n_1) \wedge P(n_2))$$

- The proof uses the premise twice (for 0 and $n_1 + d$)
- Realisability extracts witness $\lambda_{f,d} \langle f0, f(f0 + d) \rangle$
- Dialectica tries to extract counterexamples for the premise...
- ...but needs decidability of P!
- marking n as uniform turns f into a constant

Oliva's counterexample

Let $P(n)$ be an undecidable predicate.

$$\forall_n^{\mathsf{U}} \exists_{m>n} P(m) \to \forall_d \exists_{n_1,n_2} (n_1 + d < n_2 \wedge P(n_1) \wedge P(n_2))$$

- The proof uses the premise twice (for 0 and $n_1 + d$)
- Realisability extracts witness $\lambda_{f,d} \langle f0, f(f0 + d) \rangle$
- Dialectica tries to extract counterexamples for the premise...
- ...but needs decidability of P!
- marking n as uniform turns f into a constant

Hybrid functional interpretations

Hernest and Oliva (2007):

- Can realisability and Dialectica live together?
- They differ in treatment of contractions
- Linear logic explicitates contractions
- Use two sorts of linear modalities:

$$|!_k A|^x = !\forall_y \, |A|^x_y \qquad ! \, \big| !_g A \big|^x_f = |A|^x_{fx}$$
$$|?_k A|_y = ?\exists_x \, |A|^x_y \qquad ? \, \big| ?_g A \big|^f_y = |A|^{fy}_y$$

- Solution for the counterexample:

$$!_k(\forall_n \exists_{m>n} P(m)) \multimap \forall_d \exists_{n_1,n_2}(n_1 + d < n_2 \wedge P(n_1) \wedge P(n_2))$$

Hybrid functional interpretations

Hernest and Oliva (2007):

- Can realisability and Dialectica live together?
- They differ in treatment of contractions
- Linear logic explicitates contractions
- Use two sorts of linear modalities:

$$|!_k A|^x = !\forall_y |A|^x_y \qquad ! \left| !_g A \right|^x_f = |A|^x_{fx}$$

$$|?_k A|_y = ?\exists_x |A|^x_y \qquad ? \left| ?_g A \right|^f_y = |A|^{fy}_y$$

- Solution for the counterexample:

$$!_k(\forall_n \exists_{m>n} P(m)) \multimap \forall_d \exists_{n_1,n_2}(n_1 + d < n_2 \wedge P(n_1) \wedge P(n_2))$$

Hybrid functional interpretations

Hernest and Oliva (2007):

- Can realisability and Dialectica live together?
- They differ in treatment of contractions
- Linear logic explicitates contractions
- Use two sorts of linear modalities:

$$|!_k A|^x = !\forall_y |A|^x_y \qquad ! \left| !_g A \right|^x_f = |A|^x_{fx}$$

$$|?_k A|_y = ?\exists_x |A|^x_y \qquad ? \left| ?_g A \right|^f_y = |A|^{fy}_y$$

- Solution for the counterexample:

$$!_k(\forall_n \exists_{m>n} P(m)) \multimap \forall_d \exists_{n_1,n_2}(n_1 + d < n_2 \wedge P(n_1) \wedge P(n_2))$$

Hybrid functional interpretations

Hernest and Oliva (2007):

- Can realisability and Dialectica live together?
- They differ in treatment of contractions
- Linear logic explicitates contractions
- Use two sorts of linear modalities:

$$|!_k A|^x = !\forall_y \, |A|^x_y \qquad ! \, \left|!_g A\right|^x_f = |A|^x_{fx}$$
$$|?_k A|_y = ?\exists_x \, |A|^x_y \qquad ? \, \left|?_g A\right|^f_y = |A|^{fy}_y$$

- Solution for the counterexample:

$$!_k(\forall_n \exists_{m>n} P(m)) \multimap \forall_d \exists_{n_1,n_2}(n_1 + d < n_2 \wedge P(n_1) \wedge P(n_2))$$

Hybrid functional interpretations

Hernest and Oliva (2007):

- Can realisability and Dialectica live together?
- They differ in treatment of contractions
- Linear logic explicitates contractions
- Use two sorts of linear modalities:

$$|!_k A|^x = !\forall_y |A|_y^x \qquad ! \left| !_g A \right|_f^x = |A|_{fx}^x$$
$$|?_k A|_y = ?\exists_x |A|_y^x \qquad ? \left| ?_g A \right|_y^f = |A|_y^{fy}$$

- Solution for the counterexample:

$$!_k(\forall_n \exists_{m>n} P(m)) \multimap \forall_d \exists_{n_1,n_2}(n_1 + d < n_2 \wedge P(n_1) \wedge P(n_2))$$

Refined uniform quantifiers

Hernest and T. (2008):

- Dialectica extends realisability with negative content
- Need to separate positive and negative computational use
- Use refined uniform annotations for quantifiers:

$$|\forall_{\emptyset x} A|^r_y = \forall_x |A|^r_y \qquad |\forall_{+x} A|^f_y = \forall_x |A|^{fx}_y$$
$$|\forall_{-x} A|^r_{x,y} = |A|^r_y \qquad |\forall_{\pm x} A|^f_{x,y} = |A|^{fx}_y$$

- Solution for the counterexample:

$$\forall_{+n} \exists_{m>n} P(m) \to \forall_d \exists_{n_1,n_2} (n_1 + d < n_2 \wedge P(n_1) \wedge P(n_2))$$

Refined uniform quantifiers

Hernest and T. (2008):

- Dialectica extends realisability with negative content
- Need to separate positive and negative computational use
- Use refined uniform annotations for quantifiers:

$$|\forall_{\emptyset x} A|^r_y = \forall_x |A|^r_y \qquad |\forall_{+x} A|^f_y = \forall_x |A|^{fx}_y$$
$$|\forall_{-x} A|^r_{x,y} = |A|^r_y \qquad |\forall_{\pm x} A|^f_{x,y} = |A|^{fx}_y$$

- Solution for the counterexample:

$$\forall_{+n} \exists_{m>n} P(m) \to \forall_d \exists_{n_1,n_2} (n_1 + d < n_2 \wedge P(n_1) \wedge P(n_2))$$

Refined uniform quantifiers

Hernest and T. (2008):

- Dialectica extends realisability with negative content
- Need to separate positive and negative computational use
- Use refined uniform annotations for quantifiers:

$$|\forall_{\emptyset x} A|^r_y = \forall_x |A|^r_y \qquad |\forall_{+x} A|^f_y = \forall_x |A|^{fx}_y$$

$$|\forall_{-x} A|^r_{x,y} = |A|^r_y \qquad |\forall_{\pm x} A|^f_{x,y} = |A|^{fx}_y$$

- Solution for the counterexample:

$$\forall_{+n} \exists_{m>n} P(m) \to \forall_d \exists_{n_1,n_2} (n_1 + d < n_2 \wedge P(n_1) \wedge P(n_2))$$

Refined uniform quantifiers

Hernest and T. (2008):

- Dialectica extends realisability with negative content
- Need to separate positive and negative computational use
- Use refined uniform annotations for quantifiers:

$$|\forall_{\emptyset x} A|^r_y = \forall_x\, |A|^r_y \qquad |\forall_{+x} A|^f_y = \forall_x\, |A|^{fx}_y$$
$$|\forall_{-x} A|^r_{x,y} = |A|^r_y \qquad |\forall_{\pm x} A|^f_{x,y} = |A|^{fx}_y$$

- Solution for the counterexample:

$$\forall_{+n} \exists_{m>n} P(m) \to \forall_d \exists_{n_1,n_2} (n_1 + d < n_2 \wedge P(n_1) \wedge P(n_2))$$

Another counterexample

$$\forall_n(\exists_m Q(n,m) \to \exists_m Q(Sn,m)) \to Q(0,0) \to \exists_m Q(2,m)$$

Solutions:

- with the hybrid interpretation

$$!_k(\forall_n(\exists_m Q(n,m) \multimap \exists_m Q(Sn,m))) \multimap Q(0,0) \multimap \exists_m Q(2,m)$$

- with the refined uniform quantifiers

$$\forall_{+n}(\exists_m Q(n,m) \to \exists_m Q(Sn,m)) \to Q(0,0) \to \exists_m Q(2,m)$$

- Still have negative content to witness!
- Using $\exists^U$ decreases generality

Another counterexample

$$\forall_n(\exists_m Q(n,m) \to \exists_m Q(Sn,m)) \to Q(0,0) \to \exists_m Q(2,m)$$

Solutions:

- with the hybrid interpretation

$$!_k(\forall_n(\exists_m Q(n,m) \multimap \exists_m Q(Sn,m))) \multimap Q(0,0) \multimap \exists_m Q(2,m)$$

- with the refined uniform quantifiers

$$\forall_{+n}(\exists_m Q(n,m) \to \exists_m Q(Sn,m)) \to Q(0,0) \to \exists_m Q(2,m)$$

- Still have negative content to witness!
- Using $\exists^{\mathrm{U}}$ decreases generality

Another counterexample

$$\forall_n(\exists_m Q(n,m) \to \exists_m Q(Sn,m)) \to Q(0,0) \to \exists_m Q(2,m)$$

Solutions:

- with the hybrid interpretation

$$!_k(\forall_n(\exists_m Q(n,m) \multimap \exists_m Q(Sn,m))) \multimap Q(0,0) \multimap \exists_m Q(2,m)$$

- with the refined uniform quantifiers

$$\forall_{+n}(\exists_m Q(n,m) \to \exists_m Q(Sn,m)) \to Q(0,0) \to \exists_m Q(2,m)$$

- Still have negative content to witness!
- Using $\exists^{\mathrm{U}}$ decreases generality

Another counterexample

$$\forall_n(\exists_m Q(n,m) \to \exists_m Q(Sn,m)) \to Q(0,0) \to \exists_m Q(2,m)$$

Solutions:

- with the hybrid interpretation

$$!_k(\forall_n(\exists_m Q(n,m) \multimap \exists_m Q(Sn,m))) \multimap Q(0,0) \multimap \exists_m Q(2,m)$$

- with the refined uniform quantifiers

$$\forall_{+n}(\exists_m Q(n,m) \to \exists_m Q(Sn,m)) \to Q(0,0) \to \exists_m Q(2,m)$$

- Still have negative content to witness!
- Using $\exists^{\mathrm{U}}$ decreases generality

Another counterexample

$$\forall_n(\exists_m Q(n,m) \to \exists_m Q(Sn,m)) \to Q(0,0) \to \exists_m Q(2,m)$$

Solutions:

- with the hybrid interpretation

$$!_k(\forall_n(\exists_m Q(n,m) \multimap \exists_m Q(Sn,m))) \multimap Q(0,0) \multimap \exists_m Q(2,m)$$

- with the refined uniform quantifiers

$$\forall_{+n}(\exists^{\mathsf{U}}_m Q(n,m) \to \exists_m Q(Sn,m)) \to Q(0,0) \to \exists_m Q(2,m)$$

- Still have negative content to witness!
- Using $\exists^{\mathsf{U}}$ decreases generality

Fine computational control

$$\tau^+(\forall_x A) = \underbrace{\rho}_{\overset{+}{\forall}} \Rightarrow \tau^+(A)$$

$$\tau^-(\forall_x A) = \underbrace{\rho}_{\overset{-}{\forall}} \times \tau^-(A)$$

Translation	$\left\lvert \overset{+}{\forall}_x A \right\rvert^r_{x,u} := \lvert A \rvert^r_u$
Restriction	$x \notin \mathrm{FV}(\llbracket M \rrbracket^+)$
Redundant if	

Fine computational control

$$\tau^{+}(\forall_x A) = \underbrace{\rho}_{\overset{+}{\forall}} \Rightarrow \tau^{+}(A)$$

$$\tau^{-}(\forall_x A) = \underbrace{\rho}_{\overset{-}{\forall}} \times \tau^{-}(A)$$

Translation	$\left\lvert \overset{-}{\forall}_x A \right\rvert^{f}_{u} := \forall_x \lvert A \rvert^{fx}_{u}$
Restriction	$x \notin \mathrm{FV}([\![M]\!]^{-}_{i})$
Redundant if	$\tau^{+}(A) = \varepsilon$

Fine computational control

$$\tau^+(A \to B) = (\underbrace{\tau^+(A)}_{\xrightarrow{\#}} \Rightarrow \tau^+(B)) \times (\underbrace{\tau^+(A)}_{\xrightarrow{\pm}} \Rightarrow \underbrace{\tau^-(B)}_{\xrightarrow{=}} \Rightarrow \tau^-(A))$$

$$\tau^-(A \to B) = \underbrace{\tau^+(A)}_{\underset{+}{\longrightarrow}} \times \underbrace{\tau^-(B)}_{\underset{-}{\longrightarrow}}$$

Translation	$\left\lvert A \xrightarrow{\#} B \right\rvert^{r,g}_{x,u} := \lvert A\rvert^{x}_{gxu} \to \lvert B\rvert^{r}_{u}$
Restriction	$x_u \notin \mathrm{FV}([\![M]\!]^+)$
Redundant if	$\tau^+(A) = \varepsilon$ or $\tau^+(B) = \varepsilon$

Fine computational control

$$\tau^+(A \to B) = (\underbrace{\tau^+(A)}_{\xrightarrow{\#}} \Rightarrow \tau^+(B)) \times (\underbrace{\tau^+(A)}_{\xrightarrow{\pm}} \Rightarrow \underbrace{\tau^-(B)}_{\xrightarrow{=}} \Rightarrow \tau^-(A))$$

$$\tau^-(A \to B) = \underbrace{\tau^+(A)}_{\xrightarrow[+]{}} \times \underbrace{\tau^-(B)}_{\xrightarrow[-]{}}$$

Translation	$\left\lvert A \xrightarrow{\pm} B \right\rvert^{f,g}_{x,u} := \lvert A\rvert^{x}_{gu} \to \lvert B\rvert^{fx}_{u}$
Restriction	$x_u \notin \mathrm{FV}(\llbracket M \rrbracket^-_A)$
Redundant if	$\tau^+(A) = \varepsilon$ or $\tau^-(A) = \varepsilon$

Fine computational control

$$\tau^+(A \to B) = (\underbrace{\tau^+(A)}_{\xrightarrow{\#}} \Rightarrow \tau^+(B)) \times (\underbrace{\tau^+(A)}_{\xrightarrow{\pm}} \Rightarrow \underbrace{\tau^-(B)}_{\xrightarrow{=}} \Rightarrow \tau^-(A))$$

$$\tau^-(A \to B) = \underbrace{\tau^+(A)}_{\xrightarrow[+]{}} \times \underbrace{\tau^-(B)}_{\xrightarrow[-]{}}$$

Translation	$\left\lvert A \xrightarrow{=} B \right\rvert^{f,g}_{x,u} := \lvert A \rvert^{x}_{gx} \to \lvert B \rvert^{fx}_{u}$
Restriction	$y_A \notin \mathrm{FV}([\![M]\!]^-_A)$
Redundant if	$\tau^-(B) = \varepsilon$ or $\tau^-(A) = \varepsilon$

Fine computational control

$$\tau^+(A \to B) = (\underbrace{\tau^+(A)}_{\overset{\#}{\longrightarrow}} \Rightarrow \tau^+(B)) \times (\underbrace{\tau^+(A)}_{\overset{\pm}{\longrightarrow}} \Rightarrow \underbrace{\tau^-(B)}_{\overset{=}{\longrightarrow}} \Rightarrow \tau^-(A))$$

$$\tau^-(A \to B) = \underbrace{\tau^+(A)}_{\underset{+}{\longrightarrow}} \times \underbrace{\tau^-(B)}_{\underset{-}{\longrightarrow}}$$

Translation	$\left\lvert A \underset{+}{\longrightarrow} B \right\rvert_u^{f,g} := \forall_x .\ \lvert A \rvert_{gxu}^x \to \lvert B \rvert_u^{fx}$
Restriction	$x_u \notin \mathrm{FV}([\![M]\!]_i^-)$
Redundant if	$\tau^+(A) = \varepsilon$

Fine computational control

$$\tau^+(A \to B) = (\underbrace{\tau^+(A)}_{\xrightarrow{\#}} \Rightarrow \tau^+(B)) \times (\underbrace{\tau^+(A)}_{\xrightarrow{\pm}} \Rightarrow \underbrace{\tau^-(B)}_{\xrightarrow{=}} \Rightarrow \tau^-(A))$$

$$\tau^-(A \to B) = \underbrace{\tau^+(A)}_{\xrightarrow[+]{}} \times \underbrace{\tau^-(B)}_{\xrightarrow[-]{}}$$

Translation	$\left\lvert A \underset{-}{\to} B \right\rvert^{f,g}_{x} := \forall_u .\ \lvert A\rvert^{x}_{gxu} \to \lvert B\rvert^{fx}_{u}$
Restriction	$y_A \notin \mathrm{FV}([\![M]\!]^-_i)$
Redundant if	$\tau^-(B) = \varepsilon$

A solution for the counterexample

$$\bar{\forall}_n(\exists_m Q(n,m) \underset{+}{\to} \exists_m Q(Sn,m)) \to Q(0,0) \to \exists_m Q(2,m)$$

Selectively removing content

Proposition

For every formula A there exist decorated variants $A^{\oplus}$ and $A^{\ominus}$ which remove positive and negative content of A, while preserving the opposite content. Formally:

$$\tau^{+}(A^{\oplus}) = \tau^{-}(A^{\ominus}) = \varepsilon, \qquad \tau^{-}(A^{\oplus}) = \tau^{-}(A), \qquad \tau^{+}(A^{\ominus}) = \tau^{+}(A)$$

Proof.

$$(P(\vec{t}))^{\oplus} := P^{\oplus}(\vec{t}) \qquad (P(\vec{t}))^{\ominus} := P^{\ominus}(\vec{t})$$

$$(\forall_x A)^{\oplus} := \forall_x A^{\oplus} \qquad (\forall_x A)^{\ominus} := \bar{\forall}_x A^{\ominus}$$

$$(A \to B)^{\oplus} := A^{\ominus} \to B^{\oplus} \qquad (A \to B)^{\ominus} := A \xrightarrow[+,-]{} B$$

$$(\exists_x A)^{\oplus} := \exists^{U}_x A^{\oplus} \qquad (\exists_x A)^{\ominus} := \exists_x A^{\ominus}$$

Selectively removing content

Proposition

For every formula A there exist decorated variants $A^{\oplus}$ and $A^{\ominus}$ which remove positive and negative content of A, while preserving the opposite content. Formally:

$$\tau^{+}(A^{\oplus}) = \tau^{-}(A^{\ominus}) = \varepsilon, \qquad \tau^{-}(A^{\oplus}) = \tau^{-}(A), \qquad \tau^{+}(A^{\ominus}) = \tau^{+}(A)$$

Proof.

$(P(\vec{t}))^{\oplus} := P^{\oplus}(\vec{t})$	$(P(\vec{t}))^{\ominus} := P^{\ominus}(\vec{t})$
$(\forall_x A)^{\oplus} := \forall_x A^{\oplus}$	$(\forall_x A)^{\ominus} := \bar{\forall}_x A^{\ominus}$
$(A \to B)^{\oplus} := A^{\ominus} \to B^{\oplus}$	$(A \to B)^{\ominus} := A \xrightarrow[+,-]{} B$
$(\exists_x A)^{\oplus} := \exists^{\mathsf{U}}_x A^{\oplus}$	$(\exists_x A)^{\ominus} := \exists_x A^{\ominus}$

Selectively removing content

Proposition

For every formula A there exist decorated variants $A^{\oplus}$ and $A^{\ominus}$ which remove positive and negative content of A, while preserving the opposite content. Formally:

$$\tau^{+}(A^{\oplus}) = \tau^{-}(A^{\ominus}) = \varepsilon, \qquad \tau^{-}(A^{\oplus}) = \tau^{-}(A), \qquad \tau^{+}(A^{\ominus}) = \tau^{+}(A)$$

Proof.

$$(P(\vec{t}))^{\oplus} := P^{\oplus}(\vec{t}) \qquad (P(\vec{t}))^{\ominus} := P^{\ominus}(\vec{t})$$

$$(\forall_x A)^{\oplus} := \forall_x A^{\oplus} \qquad (\forall_x A)^{\ominus} := \bar{\forall}_x A^{\ominus}$$

$$(A \to B)^{\oplus} := A^{\ominus} \to B^{\oplus} \qquad (A \to B)^{\ominus} := A \underset{+,-}{\longrightarrow} B$$

$$(\exists_x A)^{\oplus} := \exists_x^{\mathsf{U}} A^{\oplus} \qquad (\exists_x A)^{\ominus} := \exists_x A^{\ominus}$$

Redundant decorations

Corollary

The decorations $\xrightarrow[+]{\#,\pm}$ *and* $\xrightarrow[-]{=}$ *can be simulated.*

Proof.

$$\tau^*(A \xrightarrow[+]{\#,\pm} B) = \tau^*(A^{\oplus} \to B) \qquad \tau^*(A \xrightarrow[-]{=} B) = \tau^*(A \to B^{\ominus})$$

□

Also, $!_k A \multimap B$ can be simulated by $A^{\ominus} \to B$.

Redundant decorations

Corollary

The decorations $\xrightarrow[+]{\#,\pm}$ *and* $\xrightarrow[-]{=}$ *can be simulated.*

Proof.

$$\tau^*(A \xrightarrow[+]{\#,\pm} B) = \tau^*(A^{\oplus} \to B) \qquad \tau^*(A \xrightarrow[-]{=} B) = \tau^*(A \to B^{\ominus})$$

□

Also, $!_k A \multimap B$ can be simulated by $A^{\ominus} \to B$.

Simulating realisability

Theorem (Simulation of realisability)

Let A be a formula in HA^ω. Then there exists a decorated variant A°, such that the realisability interpretation of A coincides with the Dialectica interpretation of A°. Formally,

1. $\tau^+(A^\circ) = \tau^\circ(A)$ *and* $\tau^-(A^\circ) = \varepsilon$
2. $|A^\circ|^r \equiv r \mathrm{~mr~} A$
3. *For every proof $M : A$ from assumptions C_i, there exists an decorated variant $M^\circ : A^\circ$ from assumptions C_i°, such that $[\![M^\circ]\!]^+ = [\![M]\!]^\circ$.*

Simulating realisability

Theorem (Simulation of realisability)

Let A be a formula in HA^ω. Then there exists a decorated variant A°, such that the realisability interpretation of A coincides with the Dialectica interpretation of A°. Formally,

1. $\tau^+(A^\circ) = \tau^\circ(A)$ *and* $\tau^-(A^\circ) = \varepsilon$
2. $|A^\circ|^r \equiv r \;\mathrm{mr}\; A$
3. *For every proof $M : A$ from assumptions C_i, there exists an decorated variant $M^\circ : A^\circ$ from assumptions C_i°, such that $[\![M^\circ]\!]^+ = [\![M]\!]^\circ$.*

Simulating realisability

Proof.

$$
\begin{aligned}
(P(\vec{t}))^\circ &:= P^\oplus(\vec{t}) \\
(\mathrm{at}(t))^\circ &:= \mathrm{at}(t) \\
(A \to B)^\circ &:= A^\circ \underset{+}{\to} B^\circ \\
(\forall_x A)^\circ &:= \bar{\forall}_x A^\circ \\
(\exists_x A)^\circ &:= \exists_x A^\circ
\end{aligned}
$$

Since $\tau^-(A^\circ) = \varepsilon$, the uniformity restrictions are always satisfied. □

Simulating uniform annotations in realisability

- For the simulation we use only $\bar{\forall}$ and $\underset{+}{\longrightarrow}$
- Flags $\overset{\pm}{\longrightarrow}$, $\overset{=}{\longrightarrow}$ and $\underset{-}{\longrightarrow}$ become redundant
- The flag $\overset{+}{\forall}$ corresponds to Berger's $\forall^{\mathsf{U}}$
- The flag $\overset{\#}{\longrightarrow}$ corresponds to a uniform implication $\overset{\mathsf{U}}{\longrightarrow}$, suggested by Schwichtenberg
- However, $A \overset{\mathsf{U}}{\longrightarrow} B$ can be simulated by $A^{\oplus} \to B$.

Simulating uniform annotations in realisability

- For the simulation we use only $\bar{\forall}$ and $\underset{+}{\longrightarrow}$
- Flags $\overset{\pm}{\longrightarrow}$, $\overset{=}{\longrightarrow}$ and $\underset{-}{\longrightarrow}$ become redundant
- The flag $\overset{+}{\forall}$ corresponds to Berger's $\forall^{\mathsf{U}}$
- The flag $\overset{\#}{\longrightarrow}$ corresponds to a uniform implication $\overset{\mathsf{U}}{\longrightarrow}$, suggested by Schwichtenberg
- However, $A \overset{\mathsf{U}}{\longrightarrow} B$ can be simulated by $A^{\oplus} \to B$.

Simulating uniform annotations in realisability

- For the simulation we use only $\bar{\forall}$ and $\underset{+}{\longrightarrow}$
- Flags $\xrightarrow{\pm}$, $\xrightarrow{=}$ and $\underset{-}{\longrightarrow}$ become redundant
- The flag $\overset{+}{\forall}$ corresponds to Berger's $\forall^{\mathsf{U}}$
- The flag $\xrightarrow{\#}$ corresponds to a uniform implication $\xrightarrow{\mathsf{U}}$, suggested by Schwichtenberg
- However, $A \xrightarrow{\mathsf{U}} B$ can be simulated by $A^{\oplus} \rightarrow B$.

Simulating uniform annotations in realisability

- For the simulation we use only $\bar{\forall}$ and $\underset{+}{\longrightarrow}$
- Flags $\xrightarrow{\pm}$, $\xrightarrow{=}$ and $\underset{-}{\longrightarrow}$ become redundant
- The flag $\overset{+}{\forall}$ corresponds to Berger's $\forall^{\mathsf{U}}$
- The flag $\xrightarrow{\#}$ corresponds to a uniform implication $\xrightarrow{\mathsf{U}}$, suggested by Schwichtenberg
- However, $A \xrightarrow{\mathsf{U}} B$ can be simulated by $A^{\oplus} \longrightarrow B$.

Simulating uniform annotations in realisability

- For the simulation we use only $\bar{\forall}$ and $\underset{+}{\longrightarrow}$
- Flags $\xrightarrow{\pm}$, $\xrightarrow{=}$ and $\underset{-}{\longrightarrow}$ become redundant
- The flag $\overset{+}{\forall}$ corresponds to Berger's $\forall^{\mathsf{U}}$
- The flag $\xrightarrow{\#}$ corresponds to a uniform implication $\xrightarrow{\mathsf{U}}$, suggested by Schwichtenberg
- However, $A \xrightarrow{\mathsf{U}} B$ can be simulated by $A^{\oplus} \to B$.

A-translation

Idea: use $\bot$ to extract computational content of proofs in NA^ω.

Theorem (Extraction via A-translation)

Let M be a proof of

$$\mathrm{HA}_0^\omega \vdash D \to \tilde{\exists}_{y^\rho} G$$

with D, G not containing $\bot$. Then

$$\mathrm{HA}^\omega \vdash D \to \exists_y G$$

Idea.

Let $\tau^\circ(\bot) = \rho$ and $\bot$ be translated to a unary predicate variable $\mathcal{A}$. Substitute $\mathcal{A}$ with $G(y)$. □

A-translation

Idea: use $\bot$ to extract computational content of proofs in NA^ω.

Theorem (Extraction via A-translation)

Let M be a proof of

$$\mathrm{HA}_0^\omega \vdash D \to \forall_{y^\rho}(G \to \bot) \to \bot$$

with D, G not containing $\bot$. Then

$$\mathrm{HA}^\omega \vdash D \to \exists_y G$$

Idea.

Let $\tau^\circ(\bot) = \rho$ and $\bot$ be translated to a unary predicate variable $\mathcal{A}$. Substitute $\mathcal{A}$ with $G(y)$. □

A-translation

Idea: use $\bot$ to extract computational content of proofs in NA^ω.

Theorem (Extraction via A-translation)

Let M be a proof of

$$\mathrm{HA}_0^\omega \vdash D \to \forall_{y^\rho}(G \to \bot) \to \bot$$

with D, G not containing $\bot$. Then

$$\mathrm{HA}^\omega \vdash D \to \exists_y G$$

Idea.

Let $\tau^\circ(\bot) = \rho$ and $\bot$ be translated to a unary predicate variable $\mathcal{A}$. Substitute $\mathcal{A}$ with $G(y)$. □

A-translation

Idea: use $\bot$ to extract computational content of proofs in NA^ω.

Theorem (Extraction via A-translation)

Let M be a proof of

$$\mathrm{HA}_0^\omega \vdash D \to \forall_{y^\rho}(G \to \bot) \to \bot$$

with D, G not containing $\bot$. Then

$$\mathrm{HA}^\omega \vdash D \to \exists_y G$$

Idea.

Let $\tau^\circ(\bot) = \rho$ and $\bot$ be translated to a unary predicate variable $\mathcal{A}$. Substitute $\mathcal{A}$ with $G(y)$. □

A-translation

Idea: use $\bot$ to extract computational content of proofs in NA^ω.

Theorem (Extraction via A-translation)

Let M be a proof of

$$\mathrm{HA}_0^\omega \vdash D \to \forall_{y^\rho}(G \to \bot) \to \bot$$

with D, G *not containing* $\bot$*. Then*

$$\mathrm{HA}^\omega \vdash D \to \exists_y G$$

Idea.

Let $\tau^\circ(\bot) = \rho$ and $\bot$ be translated to a unary predicate variable $\mathcal{A}$. Substitute $\mathcal{A}$ with $G(y)$. □

A-translation

Idea: use $\bot$ to extract computational content of proofs in NA^ω.

Theorem (Extraction via A-translation)

Let M be a proof of

$$\mathrm{HA}_0^\omega \vdash D \to \forall_{y^\rho}(G \to \bot) \to \bot$$

with D, G *not containing* $\bot$*. Then*

$$\mathrm{NA}^\omega \vdash D \to G([\![M]\!](\lambda_y y)).$$

Idea.

Let $\tau^\circ(\bot) = \rho$ and $\bot$ be translated to a unary predicate variable $\mathcal{A}$. Substitute $\mathcal{A}$ with $G(y)$. □

Definite and goal formulas

What if $\bot$ appears in D or G?

Bucholz, Berger, Schwichtenberg (2000), Seisenberger (2008):

$$
\begin{aligned}
D \quad ::= \quad & P \mid G \to D \quad (\text{if } \tau^\circ(D) = \varepsilon \text{ then } \tau^\circ(G) = \varepsilon) \\
& \mid D_1 \wedge D_2 \quad (\text{if } \tau^\circ(D_1) \neq \varepsilon \text{ then } \tau^\circ(D_2) = \varepsilon) \\
& \mid \forall_x D
\end{aligned}
$$

$$
\begin{aligned}
G \quad ::= \quad & P \mid D \to G \quad (\text{if } \tau^\circ(G) \neq \varepsilon \text{ and } \tau^\circ(D) = \varepsilon \text{ then } D \text{ decidable}) \\
& \mid G_1 \wedge G_2 \\
& \mid \forall_x G \qquad (\text{if } \tau^\circ(G) = \varepsilon)
\end{aligned}
$$

Definite and goal formulas

What if $\perp$ appears in D or G?
Bucholz, Berger, Schwichtenberg (2000), Seisenberger (2008):

$$\begin{array}{lll} D & ::= & P \mid G \to D \quad (\text{if } \tau^\circ(D) = \varepsilon \text{ then } \tau^\circ(G) = \varepsilon) \\ & & \mid D_1 \wedge D_2 \quad (\text{if } \tau^\circ(D_1) \neq \varepsilon \text{ then } \tau^\circ(D_2) = \varepsilon) \\ & & \mid \forall_x D \end{array}$$

$$\begin{array}{lll} G & ::= & P \mid D \to G \quad (\text{if } \tau^\circ(G) \neq \varepsilon \text{ and } \tau^\circ(D) = \varepsilon \text{ then } D \text{ decidable}) \\ & & \mid G_1 \wedge G_2 \\ & & \mid \forall_x G \qquad (\text{if } \tau^\circ(G) = \varepsilon) \end{array}$$

Simulating refined A-translation

Refined A-translation in a nutshell:

- Express your statement as $\vec{D} \to \forall_y (G \to \bot) \to \bot$
- Prove in minimal logic (HA_0^ω)
- Consider $\bot$ a predicate variable with computational content
- Let $\vec{D}^{\mathrm{F}} := \vec{D}\,[\bot := F]$ and $G^{\mathrm{F}} := G\,[\bot := \mathrm{F}]$
- We can prove $D^{\mathrm{F}} \to D$ and $(G^{\mathrm{F}} \to \bot) \to (G \to \bot)$
- We obtain $(F \to \bot) \to \vec{D}^{\mathrm{F}} \to \forall_y (G^{\mathrm{F}} \to \bot) \to \bot$
- Apply realisability, translating $\bot$ to a predicate variable $\mathcal{A}$
- Substitute $\mathcal{A}$ with $\{y : \; G^{\mathrm{F}}\}$

- Consider $\bot$ a predicate variable with only positive content
- Simulate realisability, translating $\bot$ to $\mathcal{A}$

Simulating refined A-translation

Refined A-translation in a nutshell:

- Express your statement as $\vec{D} \to \forall_y (G \to \bot) \to \bot$
- Prove in minimal logic (HA_0^ω)
- Consider $\bot$ a predicate variable with computational content
- Let $\vec{D}^{\mathrm{F}} := \vec{D}\,[\bot := F]$ and $G^{\mathrm{F}} := G\,[\bot := \mathrm{F}]$
- We can prove $D^{\mathrm{F}} \to D$ and $(G^{\mathrm{F}} \to \bot) \to (G \to \bot)$
- We obtain $(F \to \bot) \to \vec{D}^{\mathrm{F}} \to \forall_y (G^{\mathrm{F}} \to \bot) \to \bot$
- Apply realisability, translating $\bot$ to a predicate variable $\mathcal{A}$
- Substitute $\mathcal{A}$ with $\{y : G^{\mathrm{F}}\}$

- Consider $\bot$ a predicate variable with only positive content
- Simulate realisability, translating $\bot$ to $\mathcal{A}$

Simulating refined A-translation

Refined *A*-translation in a nutshell:

- Express your statement as $\vec{D} \to \forall_y (G \to \bot) \to \bot$
- Prove in minimal logic (HA_0^ω)
- Consider $\bot$ a predicate variable with computational content
- Let $\vec{D}^{\mathrm{F}} := \vec{D}\,[\bot := F]$ and $G^{\mathrm{F}} := G\,[\bot := \mathrm{F}]$
- We can prove $D^{\mathrm{F}} \to D$ and $(G^{\mathrm{F}} \to \bot) \to (G \to \bot)$
- We obtain $(F \to \bot) \to \vec{D}^{\mathrm{F}} \to \forall_y (G^{\mathrm{F}} \to \bot) \to \bot$
- Apply realisability, translating $\bot$ to a predicate variable $\mathcal{A}$
- Substitute $\mathcal{A}$ with $\{y : \; G^{\mathrm{F}}\}$

- Consider $\bot$ a predicate variable with only positive content
- Simulate realisability, translating $\bot$ to $\mathcal{A}$

Simulating refined A-translation

Refined A-translation in a nutshell:

- Express your statement as $\vec{D} \to \forall_y (G \to \bot) \to \bot$
- Prove in minimal logic (HA_0^ω)
- Consider $\bot$ a predicate variable with computational content
- Let $\vec{D}^{\mathrm{F}} := \vec{D}\,[\bot := F]$ and $G^{\mathrm{F}} := G\,[\bot := \mathrm{F}]$
- We can prove $D^{\mathrm{F}} \to D$ and $(G^{\mathrm{F}} \to \bot) \to (G \to \bot)$
- We obtain $(F \to \bot) \to \vec{D}^{\mathrm{F}} \to \forall_y (G^{\mathrm{F}} \to \bot) \to \bot$
- Apply realisability, translating $\bot$ to a predicate variable $\mathcal{A}$
- Substitute $\mathcal{A}$ with $\{y : \; G^{\mathrm{F}}\}$

- Consider $\bot$ a predicate variable with only positive content
- Simulate realisability, translating $\bot$ to $\mathcal{A}$

Simulating refined A-translation

Refined A-translation in a nutshell:

- Express your statement as $\vec{D} \to \forall_y (G \to \bot) \to \bot$
- Prove in minimal logic (HA_0^ω)
- Consider $\bot$ a predicate variable with computational content
- Let $\vec{D}^{\mathrm{F}} := \vec{D}[\bot := F]$ and $G^{\mathrm{F}} := G[\bot := \mathrm{F}]$
- We can prove $D^{\mathrm{F}} \to D$ and $(G^{\mathrm{F}} \to \bot) \to (G \to \bot)$
- We obtain $(F \to \bot) \to \vec{D}^{\mathrm{F}} \to \forall_y (G^{\mathrm{F}} \to \bot) \to \bot$
- Apply realisability, translating $\bot$ to a predicate variable $\mathcal{A}$
- Substitute $\mathcal{A}$ with $\{y : G^{\mathrm{F}}\}$

- Consider $\bot$ a predicate variable with only positive content
- Simulate realisability, translating $\bot$ to $\mathcal{A}$

Simulating refined A-translation

Refined A-translation in a nutshell:

- Express your statement as $\vec{D} \to \forall_y (G \to \bot) \to \bot$
- Prove in minimal logic (HA_0^ω)
- Consider $\bot$ a predicate variable with computational content
- Let $\vec{D}^{\mathrm{F}} := \vec{D}\,[\bot := F]$ and $G^{\mathrm{F}} := G\,[\bot := \mathrm{F}]$
- We can prove $D^{\mathrm{F}} \to D$ and $(G^{\mathrm{F}} \to \bot) \to (G \to \bot)$
- We obtain $(F \to \bot) \to \vec{D}^{\mathrm{F}} \to \forall_y (G^{\mathrm{F}} \to \bot) \to \bot$
- Apply realisability, translating $\bot$ to a predicate variable $\mathcal{A}$
- Substitute $\mathcal{A}$ with $\{y : G^{\mathrm{F}}\}$

- Consider $\bot$ a predicate variable with only positive content
- Simulate realisability, translating $\bot$ to $\mathcal{A}$

Simulating refined A-translation

Refined *A*-translation in a nutshell:

- Express your statement as $\vec{D} \to \forall_y (G \to \bot) \to \bot$
- Prove in minimal logic (HA_0^ω)
- Consider $\bot$ a predicate variable with computational content
- Let $\vec{D}^{\mathrm{F}} := \vec{D}\,[\bot := F]$ and $G^{\mathrm{F}} := G\,[\bot := \mathrm{F}]$
- We can prove $D^{\mathrm{F}} \to D$ and $(G^{\mathrm{F}} \to \bot) \to (G \to \bot)$
- We obtain $(F \to \bot) \to \vec{D}^{\mathrm{F}} \to \forall_y (G^{\mathrm{F}} \to \bot) \to \bot$
- Apply realisability, translating $\bot$ to a predicate variable $\mathcal{A}$
- Substitute $\mathcal{A}$ with $\{y : G^{\mathrm{F}}\}$
- Consider $\bot$ a predicate variable with only positive content
- Simulate realisability, translating $\bot$ to $\mathcal{A}$

Simulating refined A-translation

Refined A-translation in a nutshell:

- Express your statement as $\vec{D} \to \forall_y (G \to \bot) \to \bot$
- Prove in minimal logic (HA_0^ω)
- Consider $\bot$ a predicate variable with computational content
- Let $\vec{D}^{\mathrm{F}} := \vec{D}\,[\bot := F]$ and $G^{\mathrm{F}} := G\,[\bot := \mathrm{F}]$
- We can prove $D^{\mathrm{F}} \to D$ and $(G^{\mathrm{F}} \to \bot) \to (G \to \bot)$
- We obtain $(F \to \bot) \to \vec{D}^{\mathrm{F}} \to \forall_y (G^{\mathrm{F}} \to \bot) \to \bot$
- Apply realisability, translating $\bot$ to a predicate variable $\mathcal{A}$
- Substitute $\mathcal{A}$ with $\{y : G^{\mathrm{F}}\}$

- Consider $\bot$ a predicate variable with only positive content
- Simulate realisability, translating $\bot$ to $\mathcal{A}$

Simulating refined A-translation

Refined *A*-translation in a nutshell:

- Express your statement as $\vec{D} \to \forall_y (G \to \bot) \to \bot$
- Prove in minimal logic (HA_0^ω)
- Consider $\bot$ a predicate variable with computational content
- Let $\vec{D}^{\mathrm{F}} := \vec{D}\,[\bot := F]$ and $G^{\mathrm{F}} := G\,[\bot := \mathrm{F}]$
- We can prove $D^{\mathrm{F}} \to D$ and $(G^{\mathrm{F}} \to \bot) \to (G \to \bot)$
- We obtain $(F \to \bot) \to \vec{D}^{\mathrm{F}} \to \forall_y (G^{\mathrm{F}} \to \bot) \to \bot$
- Apply realisability, translating $\bot$ to a predicate variable $\mathcal{A}$
- Substitute $\mathcal{A}$ with $\{y : \ G^{\mathrm{F}}\}$

To simulate we need to:

- Consider $\bot$ a predicate variable with only positive content
- Simulate realisability, translating $\bot$ to $\mathcal{A}$

Simulating refined A-translation

Refined A-translation in a nutshell:

- Express your statement as $\vec{D} \to \forall_y (G \to \bot) \to \bot$
- Prove in minimal logic (HA_0^ω)
- Consider $\bot$ a predicate variable with computational content
- Let $\vec{D}^{\mathrm{F}} := \vec{D}\,[\bot := F]$ and $G^{\mathrm{F}} := G\,[\bot := \mathrm{F}]$
- We can prove $D^{\mathrm{F}} \to D$ and $(G^{\mathrm{F}} \to \bot) \to (G \to \bot)$
- We obtain $(F \to \bot) \to \vec{D}^{\mathrm{F}} \to \forall_y (G^{\mathrm{F}} \to \bot) \to \bot$
- Apply realisability, translating $\bot$ to a predicate variable $\mathcal{A}$
- Substitute $\mathcal{A}$ with $\{y : G^{\mathrm{F}}\}$

To simulate we need to:

- Consider $\bot$ a predicate variable with only positive content
- Simulate realisability, translating $\bot$ to $\mathcal{A}$

Simulating refined A-translation

Refined A-translation in a nutshell:

- Express your statement as $\vec{D} \to \forall_y (G \to \bot) \to \bot$
- Prove in minimal logic (HA_0^ω)
- Consider $\bot$ a predicate variable with computational content
- Let $\vec{D}^{\mathrm{F}} := \vec{D}[\bot := F]$ and $G^{\mathrm{F}} := G[\bot := \mathrm{F}]$
- We can prove $D^{\mathrm{F}} \to D$ and $(G^{\mathrm{F}} \to \bot) \to (G \to \bot)$
- We obtain $(F \to \bot) \to \vec{D}^{\mathrm{F}} \to \forall_y (G^{\mathrm{F}} \to \bot) \to \bot$
- Apply realisability, translating $\bot$ to a predicate variable $\mathcal{A}$
- Substitute $\mathcal{A}$ with $\{y : G^{\mathrm{F}}\}$

To simulate we need to:

- Consider $\bot$ a predicate variable with only positive content
- Simulate realisability, translating $\bot$ to $\mathcal{A}$

Future work

- Explore further examples of optimising by decorating
- Find an algorithm for maximal decoration of a proof
- Study a modal functional interpretation

$$\Box A \approx A^{\ominus} \qquad \Diamond A \approx A^{\oplus}$$

- When is fine computational control really necessary?
- i.e., when are the hybrid/modal decorations not sufficient?

Future work

- Explore further examples of optimising by decorating
- Find an algorithm for maximal decoration of a proof
- Study a modal functional interpretation

$$\Box A \approx A^{\ominus} \qquad \Diamond A \approx A^{\oplus}$$

- When is fine computational control really necessary?
- i.e., when are the hybrid/modal decorations not sufficient?

Future work

- Explore further examples of optimising by decorating
- Find an algorithm for maximal decoration of a proof
- Study a modal functional interpretation

$$\Box A \approx A^{\ominus} \qquad \Diamond A \approx A^{\oplus}$$

- When is fine computational control really necessary?
- i.e., when are the hybrid/modal decorations not sufficient?

Future work

- Explore further examples of optimising by decorating
- Find an algorithm for maximal decoration of a proof
- Study a modal functional interpretation

$$\Box A \approx A^{\ominus} \qquad \Diamond A \approx A^{\oplus}$$

- When is fine computational control really necessary?
- i.e., when are the hybrid/modal decorations not sufficient?

Future work

- Explore further examples of optimising by decorating
- Find an algorithm for maximal decoration of a proof
- Study a modal functional interpretation

$$\Box A \approx A^{\ominus} \qquad \Diamond A \approx A^{\oplus}$$

- When is fine computational control really necessary?
- i.e., when are the hybrid/modal decorations not sufficient?

Thank you

Thank you for your attention!