Semantics of Functional Programming

Lecture IV: Computational Adequacy

Chen, Liang-Ting
lxc@iis.sinica.edu.tw

Institute of Information Science, Academia Sinica

Formosan Summer School on Logic, Language, and Computation 2014

Overview

So far we have given two kinds of semantics for **PCF**. For a program M of type σ,

- one gives how the program M is evaluated to a closed value V via the reduction relation $\mathrm{M} \Downarrow \mathrm{V}$;
- the other defines what the denotation $[\![\mathrm{M}]\!]$ of M is in a domain D_σ.

In this lecture, we will compare these two approaches and discuss some issues arising from them:

Correctness	$\mathrm{M} \Downarrow \mathrm{V}$ implies $[\![\mathrm{M}]\!] = [\![\mathrm{V}]\!]$.
Completeness	$[\![\mathrm{M}]\!] = n$ implies $\mathrm{M} \Downarrow \underline{n}$
Computational adequacy	Both of correctness and completeness hold.

Closed values of $\mathtt{nat}$ do not diverge

The bottom element $\perp$ models the divergence of computation. A closed value of $\mathtt{nat}$ is meant to be some natural number, so it shouldn't diverge.

Lemma 1

For every closed value V *of type* $\mathtt{nat}$*, the denotation* $[\![\mathsf{V}]\!]$ *is an element of* $\mathbb{N}$*. In particular,* $[\![\mathsf{V}]\!] \neq \perp$*.*

Proof.

By structural induction on closed values. For the following cases

$$\frac{}{\mathtt{zero}\ \ \mathbf{val}} \qquad \frac{\mathsf{M}\ \ \mathbf{val}}{\mathtt{suc}\ \mathsf{M}\ \ \mathbf{val}} \qquad \frac{}{\lambda x.\, \mathsf{M}\ \ \mathbf{val}}$$

it is easy to see that $[\![\mathtt{zero}]\!]$ and $[\![\mathtt{suc}\ \mathsf{M}]\!]$, if $[\![\mathsf{M}]\!] \in \mathbb{N}$, are elements of $\mathbb{N}$ by the definition of $[\![-]\!]$. On the other hand, $\lambda x.\, \mathsf{M}$ cannot be of type $\mathtt{nat}$, so this case holds vacuously. $\square$

Now we show that denotational semantics is correct with respect to denotational semantics:

Theorem 2

For every two programs M *and* V, $\mathsf{M} \Downarrow \mathsf{V}$ *implies* $[\![\mathsf{M}]\!] = [\![\mathsf{V}]\!]$.

A sanity check: By Preservation Theorem, it is known that $[\![\vdash \mathsf{M} : \tau]\!]$ and $[\![\vdash \mathsf{V} : \sigma]\!]$ are of the same type if $\mathsf{M} \Downarrow \mathsf{V}$, so their range $[\![\tau]\!]$ and $[\![\sigma]\!]$ are the same.

Proof sketch.

Prove $[\![\mathsf{M}]\!] = [\![\mathsf{V}]\!] \in [\![\tau]\!]$ by structural induction on the derivation of $\mathsf{M} \Downarrow \mathsf{V}$. $\square$

Proof of correctness

We show the case ($\Downarrow$-suc) first and the cases ($\Downarrow$-zero) and ($\Downarrow$-lam) are similar and easy.

- For ($\Downarrow$-suc), we show that $[\![\texttt{suc}\ \mathsf{M}]\!] = [\![\texttt{suc}\ \mathsf{V}]\!]$ if $[\![\mathsf{M}]\!] = [\![\mathsf{V}]\!]$. By definition, we simply calculate its denotation directly:

$$[\![\texttt{suc}\ \mathsf{M}]\!] = S([\![\mathsf{M}]\!]) = S([\![\mathsf{V}]\!]) = [\![\texttt{suc}\ \mathsf{V}]\!]$$

  where the middle equality follows from the induction hypothesis.

Try to do the cases ($\Downarrow$-zero), ($\Downarrow$-lam), and ($\Downarrow$-$\texttt{ifz}_0$).

The case ($\Downarrow$-app) is slightly complicated, as we have to address the binding structure using Substitution Lemma.

- For ($\Downarrow$-app), we show that $[\![\mathsf{M}\ \mathsf{N}]\!] = [\![V]\!]$ if $[\![\mathsf{M}]\!] = [\![\lambda x.\,\mathsf{E}]\!]$ and $[\![E[\mathsf{N}/x]]\!] = [\![V]\!]$. We calculate the denotation as follows

$$\begin{aligned}
[\![\mathsf{M}\ \mathsf{N}]\!] &= ev([\![\mathsf{M}]\!], [\![\mathsf{N}]\!]) \\
&= ev([\![\lambda x.\,\mathsf{E}]\!], [\![\mathsf{N}]\!]) \\
&= ev([\![x : \sigma \vdash \mathsf{E} : \tau]\!], [\![\mathsf{N}]\!]) \\
&= [\![x : \sigma \vdash \mathsf{E} : \tau]\!]([\![\mathsf{N}]\!]) = [\![\mathsf{E}[\mathsf{N}/x]]\!] = [\![\mathsf{V}]\!]
\end{aligned}$$

  where the last but one equation follows from Substitution Lemma.

- Complete the remaining two (interesting) cases ($\Downarrow$-$\mathtt{ifz}_1$) and ($\Downarrow$-fix). *Hint.* Consider Substitution Lemma and the properties of the fixpoint operator μ.

Logical equivalence

It is natural to define an "equality" between programs according to its computation outcome:

Definition 3 (Applicative approximation)

For each type σ, define a relation $\precsim_\sigma$ between programs of σ:

1. For the type `nat`, define

$$\mathsf{M} \precsim_{\mathtt{nat}} \mathsf{N}$$

   if for all $n \in \mathbb{N}$, $\mathsf{M} \Downarrow \underline{n}$ implies $\mathsf{N} \Downarrow \underline{n}$

2. For every type $\sigma \to \tau$, define

$$\mathsf{M} \precsim_{\sigma \to \tau} \mathsf{N}$$

   if for all program P, $\mathsf{M}\ P \precsim_\tau \mathsf{N}\ P$.

Two programs M and N of the same type σ are **logically equivalent** denoted $\mathsf{M} \simeq_\sigma \mathsf{N}$ if $\mathsf{M} \precsim_\sigma \mathsf{N}$ and $\mathsf{M} \succsim_\sigma \mathsf{N}$.

The relation $\precsim_\sigma$ is a preorder, so $\simeq_\sigma$ is indeed an equivalence.

Proposition 4

The logical equivalence $\simeq_\sigma$ is a reflexive, symmetry, and transitive relation, i.e. an equivalence relation.

A program M can be replaced by another program N without changing results if M $\simeq_\sigma$ N and it is desirable for efficiency.

Example 5

The following two programs are logically equivalent:

$$\lambda x.\, x \quad \text{and} \quad \lambda x.\, \mathtt{pred}\,(\mathtt{suc}\, x)$$

Reduction respects logical equivalence

Recall that from $\mathsf{M} \rightsquigarrow^* \mathsf{M}'$ and $\mathsf{M}' \Downarrow \mathsf{V}$ it follows that $\mathsf{M} \Downarrow \mathsf{V}$ in the agreement between $\rightsquigarrow$ and $\Downarrow$.

Proposition 6

Let M *and* M' *be programs of type* σ. *If* $\mathsf{M} \rightsquigarrow^* \mathsf{M}'$, *then* $\mathsf{M} \gtrsim_\sigma \mathsf{M}'$.

The other direction follows from the determinacy and closed values cannot be reduced further:

Proposition 7

Let V *be a closed value, and* M *and* N *terms. Suppose that* $\mathsf{M} \rightsquigarrow^* \mathsf{V}$ *and* $\mathsf{M} \rightsquigarrow^* \mathsf{M}'$. *Then it follows that* $\mathsf{M}' \rightsquigarrow^* \mathsf{V}$.

Corollary 8 (Reduction entails logical equivalence)

If $\mathsf{M} \rightsquigarrow^* \mathsf{M}'$, *then* $\mathsf{M} \simeq_\sigma \mathsf{M}'$.

However, logical equivalence goes beyond reduction. Consider the following two programs of type $\mathtt{nat} \to \mathtt{nat} \to \mathtt{nat}$:

$$\lambda x.\, \lambda y.\, x + y$$

and

$$\lambda x.\, \lambda y.\, y + x$$

Surely the addition of natural numbers are commutative, but *why?* By definition they are already closed values, so they cannot be reduced to each other.

Remark 2.1

We can show directly that these two programs are logically equivalent in dependent type theory. Yet, we will present an external approach using denotational semantics in the absense of the identity type.

In the following, we will show that for every program M of type `nat` if $[\![\mathrm{M}]\!] = n$ then M reduces to the numeral $\underline{n}$.

- Define a relation R_σ for each type σ between the domain $[\![\sigma]\!] = D_\sigma$ and the collection of programs of type σ:

  $$R_\sigma \subseteq D_\sigma \times \mathrm{Prg}_\sigma$$

  for every type σ where $\mathrm{Prg}_\sigma = \{\, \mathrm{M} \mid\ \vdash \mathrm{M} : \sigma \,\}$.
- Then prove that $[\![\mathrm{M}]\!]\ R_\sigma\ \mathrm{M}$ for every program M of type σ, and, by construction $[\![\mathrm{M}]\!]\ R_{\mathtt{nat}}\ \mathrm{M}$ is equivalent to that $[\![M]\!] = n$ implies $\mathrm{M} \Downarrow \underline{n}$.

With this property, we can conclude that denotational equivalence entails logical equivalence.[1]

[1] But, the converse may fail.

Logical relation between semantics and syntax

Definition 9 (Logical relation)

For every type σ, define a relation $R_\sigma \subseteq D_\sigma \times \mathrm{Prg}_\sigma$ inductively as follows:

- $d \; R_{\mathtt{nat}} \; \mathrm{M}$ if M reduces to $\underline{n}$ whenever d is a natural number:

$$d \; R_{\mathtt{nat}} \; \mathrm{M} \quad \text{if} \quad \forall n \in \mathbb{N}.\, d = n \implies \mathrm{M} \Downarrow \underline{n}$$

- for every function type, $f \; R_{\sigma \to \tau} \; \mathrm{M}$ if the outcome is always related whenever the input is related:

$$f \; R_{\sigma \to \tau} \; \mathrm{M} \quad \text{if}$$

$$\forall d, \mathrm{N}.\, d \; R_\sigma \; \mathrm{N} \implies f(d) \; R_\tau \; \mathrm{M \; N}$$

For example, $0 \; R_{\mathtt{nat}} \; \mathtt{zero}$, and $n + 1 \; R_{\mathtt{nat}} \; \mathtt{suc} \; \mathrm{M}$ wherever $n \; R_{\mathtt{nat}} \; \mathrm{M}$ for $n \in \mathbb{N}$.

Properties of R_σ

Lemma 10

For every type σ, the following statements are true:

1. *If $d' \sqsubseteq d$ and $d\ R_\sigma\ \mathrm{M}$, then $d'\ R_\sigma\ \mathrm{M}$;*
2. *For every $\mathrm{M} \in \mathrm{Prg}_\sigma$, the set*

$$R_\sigma \mathrm{M} := \{\, d \in D_\sigma \mid d\ R_\sigma\ \mathrm{M} \,\}$$

   contains $\perp$ and is closed under directed sups;[2]

3. *If $d\ R_\sigma\ \mathrm{M}$ and $M \precsim_\sigma \mathrm{M}'$, then $d\ R_\sigma\ \mathrm{M}'$.*

Proof.

By induction on σ. □

[2] Let S be an arbitrary directed subset of D_σ, if $d\ R_\sigma\ \mathrm{M}$ for every $d \in S$, then $\bigsqcup S\ R_\sigma\ \mathrm{M}$.

Lemma 11 (General recursion)

If we have $f \; R_{\sigma\to\sigma} \; (\lambda x.\, \mathrm{M})$*, then* $\mu(f) \; R_{\sigma} \; (\Upsilon x.\, \mathrm{M})$*.*

Proof sketch.

By definition $\mu(f)$ is the directed supremum of the following directed sequence

$$\bot \sqsubseteq f(\bot) \sqsubseteq f^2(\bot) \sqsubseteq \cdots \sqsubseteq f^i(\bot) \sqsubseteq \cdots ,$$

so it suffices to show that

$$f^i(\bot) \; R_\sigma \; (\Upsilon x.\, \mathrm{M})$$

for every $i \in \mathbb{N}$, because $R_\sigma(\Upsilon x.\mathrm{M})$ is closed under directed sups. We prove it by induction on i and properties of R_σ. $\square$

The complete proof is listed below.

For $i = 0$: By definition $f^0(\bot) = \bot$, so $\bot \; R_\sigma \; (\mathtt{Y}x.\, \mathrm{M})$ follows.

For $i = n + 1$: By the assumption $f \; R_{\sigma\to\sigma} \; (\lambda x.\mathrm{M})$, it follows that

$$f^{n+1}(\bot) \; R_\sigma \; (\lambda x.\, \mathrm{M}) \, (\mathtt{Y}x.\, \mathrm{M})$$

by the induction hypothesis $f^n(\bot) \; R_\sigma \; (\mathtt{Y}x.\, \mathrm{M})$. The RHS reduces to $\mathrm{M}[\mathtt{Y}x.\mathrm{M}/x]$ and $\mathtt{Y}x.\mathrm{M} \rightsquigarrow \mathrm{M}[\mathtt{Y}x.\mathrm{M}/x]$, so the RHS is logically equivalent to $\mathtt{Y}x.\mathrm{M}$. Hence, it follows that

$$f^{n+1}(\bot) \; R_\sigma \; (\mathtt{Y}x.\, \mathrm{M}).$$

Therefore, it follows that $\bigsqcup_{i\in\mathbb{N}} f^i(\bot) \; R_\sigma \; (\mathtt{Y}x.\, \mathrm{M})$.

The Main Lemma – Substitution

Lemma 12 (Substitution)

Let $\Gamma = x_1 : \sigma_1, \ldots, x_k : \sigma_k$ *be a context and* $d_i \; R_{\sigma_i} \; \mathsf{N}_i$ *for* $i = 1, \ldots, n$. *For every well-typed term* M *we have*

$$[\![\, \Gamma \vdash \mathsf{M} : \tau \,]\!](\vec{d}) \; R_\tau \; \mathsf{M}[\vec{N}/\vec{x}]$$

Theorem 13 (Completeness)

For every program M *of type* `nat`*, we have* $\mathsf{M} \Downarrow \underline{n}$ *if* $[\![\mathsf{M}]\!] = n$.

Proof.

A special case of the previous lemma:

$$[\![\, \vdash \mathsf{M} : \tau]\!](*) \; R_\sigma \; \mathsf{M}$$

where the LHS is $[\![\mathsf{M}]\!]$. □

Proof of the Main Lemma

To prove the lemma, do induction on the typing rules for **PCF**. For convenience, we write

$$\vec{d}\ R\ \vec{\mathsf{N}} \quad \text{for} \quad d_i\ R_{\sigma_i}\ \mathsf{N}_i \quad \text{indexed by } i = 1, \ldots, n$$

where $\vec{d}$ stands for $(d_1, \ldots, d_n)$ and $\vec{\mathsf{N}}$ stands for $(\mathsf{N}_1, \ldots, \mathsf{N}_n)$.

(z), (s) These two cases follow from $0\ R_{\mathtt{nat}}\ \mathtt{zero}$ and $n+1\ R_{\mathtt{nat}}\ \mathtt{suc}\ \mathsf{M}$ whenever $n\ R_{\mathtt{nat}}\ \mathsf{M}$.

(var) To show that

$$[\![\ldots, x_i : \sigma_i, \cdots \vdash x_i : \sigma_i]\!] \; R_{\sigma_i} \; x_i[\vec{\mathsf{N}}/\vec{x}]$$

we check both sides separately. By definition, we have

$$[\![\ldots, x_i : \sigma_i, \cdots \vdash x_i : \sigma_i]\!](\vec{d}) = d_i \quad \text{and} \quad [\vec{\mathsf{N}}/\vec{x}] = \mathsf{N}_i.$$

Therefore, from the assumption it follows that $d_i \; R_\sigma \; \mathsf{N}_i$ for every i.

(abs) We need to show that

$$[\![\Gamma \vdash \lambda x.\, \mathsf{M} : \tau]\!](\vec{d}) \; R_{\sigma \to \tau} \; (\lambda x.\, \mathsf{M})[\vec{\mathsf{N}}/\vec{x}] \tag{1}$$

under the induction hypothesis

$$[\![\Gamma, x : \sigma \vdash \mathsf{M} : \tau]\!](\vec{d}, d) \; R_{\tau} \; \mathsf{M}[\, \vec{\mathsf{N}}, \mathsf{N} \,/\, \vec{x}, x \,].$$

- For the LHS, we have by definition

$$\begin{aligned} & [\![\Gamma \vdash \lambda x.\mathsf{M} : \tau]\!](\vec{d})(d) \\ = \; & [\![\Gamma, x : \sigma \vdash \mathsf{M} : \tau]\!](\vec{d}, d). \end{aligned}$$

- For the RHS, we have

$$\begin{aligned} & (\lambda x.\, \mathsf{M})[\vec{\mathsf{N}}/\vec{x}] \; \mathsf{N} \\ \rightsquigarrow \; & (\lambda x.\, \mathsf{M})[\vec{\mathsf{N}}/\vec{x}][\mathsf{N}/x] \\ = \; & (\lambda x.\, \mathsf{M})[\, \vec{\mathsf{N}}, \mathsf{N} \,/\, \vec{x}, x \,] \end{aligned}$$

  and it follows that these two terms are logically equivalent. Thus, (1) follows by the definition of $R_{\sigma \to \tau}$.

(Y) We show that $[\![\Gamma \vdash \mathtt{Y}x.\,\mathsf{M} : \sigma]\!](\vec{d})\ R_\sigma\ (\mathtt{Y}x.\,\mathsf{M})[\vec{\mathsf{N}}/\vec{x}]$ under the assumption that

$$[\![\Gamma, x : \sigma \vdash \mathsf{M} : \sigma]\!](\vec{d}, d)\ R_\sigma\ \mathsf{M}[\,\vec{\mathsf{N}}, \mathsf{N}/\vec{x}, x\,] \qquad (2)$$

Recall the lemma for general recursion. It suffices to show $\Lambda[\![\Gamma, x : \sigma \vdash \mathsf{M} : \sigma]\!](\vec{d})\ R_{\sigma\to\sigma}\ \lambda x.\,\mathsf{M}[\vec{\mathsf{N}}/\vec{x}]$ or, equivalently

$$[\![\Gamma, x : \sigma \vdash \mathsf{M} : \sigma]\!](\vec{d}, d)\ R_\sigma\ (\lambda x.\,\mathsf{M}[\vec{\mathsf{N}}/\vec{x}])\ \mathsf{N} \qquad (3)$$

for every $d\ R_\sigma\ \mathsf{N}$. The RHS can be reduced to

$$\mathsf{M}[\vec{\mathsf{N}}/\vec{x}][\mathsf{N}/x] = \mathsf{M}[\,\vec{\mathsf{N}}, \mathsf{N}\ /\ \vec{x}, x\,],$$

so (2) implies (3) by logical equivalence.

(app), (ifz) In-class Exercises.

Applicative approximation coincides with logical relation

Lemma 14

For every program M *and* N *of type* σ*,*

$$\mathsf{M} \precsim_\sigma \mathsf{N} \quad \textit{if and only if} \quad [\![\mathsf{M}]\!] \; R_\sigma \; \mathsf{N}.$$

Proof.

$\mathsf{M} \precsim_\sigma \mathsf{N}$. By adequacy, we have $[\![\mathsf{M}]\!] \; R_\sigma \; \mathsf{M}$, so $[\![\mathsf{M}]\!] \; R_\sigma \; \mathsf{N}$.

$[\![M]\!] \; R_\sigma \; \mathsf{N}$. Prove it by induction on σ.

- $\mathtt{nat}$: If $[\![\mathsf{M}]\!] \; R_{\mathtt{nat}} \; \mathsf{N}$, then $\mathsf{N} \Downarrow \underline{n}$ whenever $[\![\mathsf{M}]\!] = n$.
- $\sigma \to \tau$: For $\sigma \to \tau$, by adequacy, we have $[\![P]\!] \; R_\sigma \; P$ for every P, so by assumption and $[\![\mathsf{M} \; P]\!] = [\![\mathsf{M}]\!]([\![P]\!]) \; R_\tau \; \mathsf{N} \; P$. By induction hypothesis, $\mathsf{M} \; P \precsim_\tau \mathsf{N} \; P$ for every P, so $\mathsf{M} \precsim_{\sigma \to \tau} \mathsf{N}$ by definition.

$\square$

Corollary 15

Given two programs M *and* N *of type* σ*, if* $[\![\mathsf{M}]\!] = [\![\mathsf{N}]\!]$*, then* M *and* N *are logically equivalent.*

Proof.

1. By adequacy $[\![\mathsf{M}]\!]\ R\ \mathsf{M}$ and by assumption $[\![\mathsf{N}]\!] = [\![\mathsf{M}]\!]\ R\ \mathsf{M}$, it follows that $\mathsf{N} \precsim \mathsf{M}$.
2. Similarly, $[\![\mathsf{M}]\!]\ R\ \mathsf{N}$, so $\mathsf{M} \precsim \mathsf{N}$.

Hence, M and N are logically equivalent. □

From this property, techniques and results in denotational semantics can be used to argue logical equivalence and reductions.

Compactness

Recall that the semantics of general recursion is the least upper bound of its finite unfoldings

$$[\![\mathtt{Y}x.\,\mathsf{M}]\!] = \bigsqcup_{i\in\mathbb{N}} [\![\mathtt{Y}^i x.\,\mathsf{M}]\!]$$

where $\mathtt{Y}^i x.\,\mathsf{M}$ is defined inductively by

1. $\mathtt{Y}^0 x.\,\mathsf{M} := \mathtt{Y}x.\,x$ and
2. $\mathtt{Y}^{n+1} x.\,\mathsf{M} := \mathsf{M}[\mathtt{Y}^n x.\,\mathsf{M}/x]$

and $[\![\mathtt{Y}^i x.\,\mathsf{M}]\!] = [\![\lambda x.\,\mathsf{M}]\!]^i(\bot)$.

Theorem 16

Suppose that $x \neq y$,

$$y : \sigma \vdash E : \mathtt{nat} \quad \textit{and} \quad \vdash \mathtt{Y}x.\,\mathsf{M} : \sigma.$$

If $E[\mathtt{Y}x.\,\mathsf{M}/y] \Downarrow \underline{n}$ then $E[\mathtt{Y}^m x.\,\mathsf{M}/y] \Downarrow \underline{n}$ for some m.

Proof.

By the Substitution Lemma, we have

$$[\![E[\mathtt{Y}x.\,\mathrm{M}/y]]\!] = [\![y : \sigma \vdash E : \mathtt{nat}]\!]([\![\mathtt{Y}x.\,\mathrm{M}]\!]).$$

Let $g := [\![y : \sigma \vdash E : \mathtt{nat}]\!]$ and $f := [\![x : \sigma \vdash \mathrm{M} : \sigma]\!]$.

$$\begin{aligned}
[\![y : \sigma \vdash E : \mathtt{nat}]\!]([\![\mathtt{Y}x.\,\mathrm{M}]\!]) &= g(\mu f) \\
&= g\Big(\bigsqcup_{i \in \mathbb{N}} f^i(\bot)\Big) \\
&= \bigsqcup_{i \in \mathbb{N}} (g \circ f^i)(\bot) = n
\end{aligned}$$

Therefore there exists some $m \in \mathbb{N}$ such that $(g \circ f^m)(\bot) = n$. By adequacy, it follows that $E[\mathtt{Y}^m x.\,\mathrm{M}/y] \Downarrow \underline{n}$. $\square$

Finite unfoldings approximate general recursion

Lemma 17

Suppose that $x : \sigma \vdash \mathsf{M} : \sigma$*. Then for every* $i \in \mathbb{N}$*, we have*

$$\mathtt{Y}^i x.\,\mathsf{M} \precsim_\sigma \mathtt{Y}x.\,\mathsf{M}.$$

The proof is left as an exercise.

Theorem 18 (Fixed Point Induction)

Suppose that $x : \sigma \vdash \mathsf{M} : \sigma$, $x : \sigma \vdash \mathsf{N} : \sigma$ *and*

$$\mathtt{Y}^i x.\,\mathsf{M} \simeq_\sigma \mathtt{Y}^i x.\,\mathsf{N}$$

for every $i \in \mathbb{N}$*. Then, we also have*

$$\mathtt{Y}x.\,\mathsf{M} \simeq_\sigma \mathtt{Y}x.\,\mathsf{N}$$

Proof.

We show that $\mathtt{Y}x.\,\mathsf{M} \precsim_\sigma \mathtt{Y}x.\,\mathsf{N}$, or equivalently $[\![\mathtt{Y}x.\,\mathsf{M}]\!]\ R_\sigma\ \mathtt{Y}x.\,\mathsf{N}$, and the other direction follows similarly.
Let $f := [\![x : \sigma \vdash \mathsf{M} : \sigma]\!]$ and $g := [\![x : \sigma \vdash \mathsf{N} : \sigma]\!]$. Since the set

$$R_\sigma(\mathtt{Y}x.\,\mathsf{N}) = \{\, d \in D_\sigma \mid d\ R_\sigma\ \mathtt{Y}x.\,\mathsf{N} \,\}$$

is closed under directed supremum, it suffices to show that

$$[\![\mathtt{Y}^i x.\,\mathsf{M}]\!]\ R_\sigma\ \mathtt{Y}x.\,\mathsf{N}$$

for every i. By assumption, we have $[\![\mathtt{Y}^i x.\,\mathsf{M}]\!]\ R_\sigma\ \mathtt{Y}^i x.\,\mathsf{N}$, so it suffices to show that $\mathtt{Y}^i x.\,\mathsf{N} \precsim_\sigma \mathtt{Y}x.\,\mathsf{N}$. By the previous lemma the statement follows. $\square$

Exercises

Show that the following pairs of programs are logically equivalent.

1