A model of PCF in Guarded Type Theory

Marco Paviotti¹ Rasmus Møgelberg¹ Lars Birkedal²

 1 IT University of Copenhagen

²Aarhus University

June 23th, 2015 MFPS 2015 Nijmegen, Netherlands

In Type Theory

unrestricted fix-point fix:
$$(A \rightarrow A) \rightarrow A$$
 is inconsistent e.g. fix(id) : A leads to every type to be inhabited

In Guarded Type Theory

restricted fix-points are allowed by using the ▷ operator

- next : $A \rightarrow \triangleright A$
- \circledast : $\triangleright (A \rightarrow B) \rightarrow \triangleright A \rightarrow \triangleright B$
- fix: $(\triangleright A \to A) \to A$

s.t. f(next(fix(f))) = fix(f)

• $X \cong A \times \triangleright X$

In Type Theory

unrestricted fix-point fix:
$$(A \rightarrow A) \rightarrow A$$
 is inconsistent e.g. fix(id) : A leads to every type to be inhabited

In Guarded Type Theory

restricted fix-points are allowed by using the ▷ operator

- next : $A \rightarrow \triangleright A$
- \circledast : $\triangleright (A \rightarrow B) \rightarrow \triangleright A \rightarrow \triangleright B$
- fix: $(\triangleright A \rightarrow A) \rightarrow A$

s.t.
$$f(\text{next}(\text{fix}(f))) = \text{fix}(f)$$

• $X \cong A \times \triangleright X$

Streams

$$\mathsf{Str}_{A} \cong A \times \mathsf{Str}_{A}$$

Streams in Coq

- ones = 1::ones
- bad = tail bad
- nats = 0::map(1+) nats









Streams

$$\mathsf{Str}_{A} \cong A \times \mathsf{Str}_{A}$$

Streams in Coq

- ones = 1::ones
- bad = tail bad
- nats = 0:: map (1+) nats

$\operatorname{\mathsf{Str}}^g_A \cong A \times \triangleright \operatorname{\mathsf{Str}}^g_A$

Guarded Streams

- $_::_:A \to \triangleright \operatorname{\mathsf{Str}}^{\mathcal{G}}_{\Delta} \to \operatorname{\mathsf{Str}}^{\mathcal{G}}_{\Delta} \quad \mathsf{head}: \operatorname{\mathsf{Str}}^{\mathcal{G}}_{\Delta} \to A \quad \mathsf{tail}: \operatorname{\mathsf{Str}}^{\mathcal{G}}_{\Delta} \to \triangleright \operatorname{\mathsf{Str}}^{\mathcal{G}}_{\Delta}$
 - ones = $1::ones:Str^g_A$
 - bad = tail bad :/ Str^g
 - nats = 0:: next(map (1+)) \circledast nats: Str_A^g









Model of Guarded Type Theory Birkedal and Møgelberg '12

The category of presheaves over ω

$$\operatorname{Str}_{A}^{g} \qquad A \times 1 \stackrel{r_{1}}{\longleftarrow} A \times (A \times 1) \stackrel{r_{2}}{\longleftarrow} A \times (A \times A \times 1)$$

$$\triangleright \operatorname{Str}_{A}^{g} \qquad 1 \stackrel{!}{\longleftarrow} A \times 1 \stackrel{r_{2}}{\longleftarrow} A \times A \times 1$$

$$A \times \triangleright \operatorname{Str}_{A}^{g} \qquad A \times 1 \stackrel{r_{1}}{\longleftarrow} A \times A \times 1 \stackrel{r_{2}}{\longleftarrow} A \times A \times A \times 1$$

Can we do denotational semantics in Guarded Type Theory ?

in particular, is it possible to model recursion with guarded recursion?

Can we do denotational semantics in Guarded Type Theory ?

in particular, is it possible to model recursion with guarded recursion?

- Motivations Mechanising denotational semantics in a proof-assistant
- Contributions
 - + Model of PCF in GTT
 - + Adequacy Theorem proved in GTT

Similar to Escardo's metric model ¹, but here the whole development is entirely carried out within guarded type theory

¹M.H. Escardo, "A metric model of PCF". Presented at the *Workshop on Realizability Semantics and Applications*, 1999

Outline

• Operational Semantics of PCF

Denotational Semantics

• Computational Adequacy

• Discussion

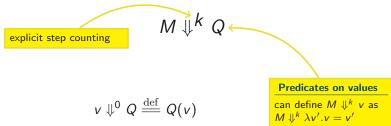
PCF

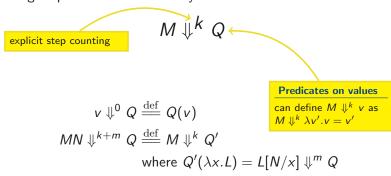
$$\begin{split} \sigma,\tau &:= \mathbf{nat} \mid \sigma \to \tau \\ \mathit{L},\mathit{M},\mathit{N} &:= \underline{n} \mid x \mid \lambda x.\mathit{M} \mid \mathsf{pred} \ \mathit{M} \mid \mathsf{succ} \ \mathit{M} \mid \mathsf{Y} \ \mathit{M} \mid \mathsf{ifz} \ \mathit{L} \ \mathit{M} \ \mathit{N} \end{split}$$

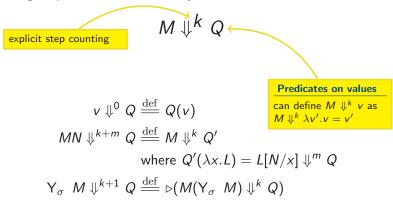
$$M \Downarrow^k Q$$

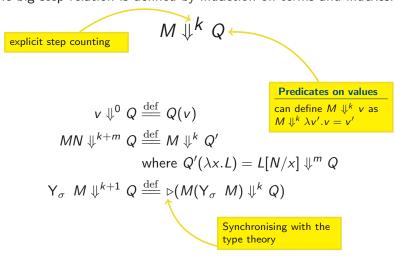












Small-Step Operational Semantics

$$\frac{(\lambda x : \sigma.M)(N) \to^{0} M[N/x]}{\frac{M \to^{k} M'}{M(N) \to^{k} M'(N)}} \overline{Y_{\sigma} M \to^{1} M(Y_{\sigma} M)}$$

Let \rightarrow^0_* be the reflexive, transitive closure of \rightarrow^0 .

$$M \Rightarrow^0 Q \stackrel{\mathrm{def}}{=\!\!\!=\!\!\!=} \Sigma N \colon \mathrm{Term}_{\scriptscriptstyle{ ext{PCF}}}.M \to^0_* N \ \mathrm{and} \ Q(N)$$
 $M \Rightarrow^{k+1} Q \stackrel{\mathrm{def}}{=\!\!\!=\!\!\!=} \Sigma M', M'' \colon \mathrm{Term}_{\scriptscriptstyle{ ext{PCF}}}.M \to^0_* M'$
and $M' \to^1 M'' \ \mathrm{and} \ \triangleright (M'' \Rightarrow^k Q)$

Define $M \Rightarrow^k v$ as $M \Rightarrow^k \lambda v'.v = v'$

Lemma

$$M \Downarrow^k v \Leftrightarrow M \Rightarrow^k v$$

Outline

• Operational Semantics of PCF

• Denotational Semantics

• Computational Adequacy

• Discussion

Lifting Monad

$$LA \cong A + \triangleright LA$$

Lifting monad

- $\eta: A \rightarrow LA$ $\theta: \triangleright LA \rightarrow LA$
- Time step operation : $\delta = \theta \circ \text{next} : LA \to LA$
- Bottom element $\bot = fix(\theta)$
- LA is a free ⊳-algebra on A
- L is the guarded recursive version of Capretta's partiality monad¹

¹Venanzio Capretta, "General Recursion via Co-Inductive Types", In *Logical Methods in Computer Science*, 2005

Lifting monad

$$LA \cong A + \triangleright LA$$

Lifting monad $L\mathbb{N}\cong\mathbb{N}+\triangleright L\mathbb{N}$ $L\mathbb{N} \cong \mathbb{N}+\triangleright L\mathbb{N}$ $\mathbb{N}+1 \stackrel{r_1}{\longleftarrow} \mathbb{N}+\mathbb{N}+1 \stackrel{r_2}{\longleftarrow} \mathbb{N}+\mathbb{N}+\mathbb{N}+1$ $\mathbb{N}+\triangleright L\mathbb{N} = \mathbb{N}+1 \stackrel{r_1}{\longleftarrow} \mathbb{N}+\mathbb{N}+1 \stackrel{r_2}{\longleftarrow} \mathbb{N}+\mathbb{N}+\mathbb{N}+1$

Interpreting PCF

Interpreting Types

$$\llbracket \mathbf{nat} \rrbracket \stackrel{\mathrm{def}}{=\!\!\!=\!\!\!=} L \mathbb{N}$$
$$\llbracket \tau \to \sigma \rrbracket \stackrel{\mathrm{def}}{=\!\!\!=\!\!\!=} \llbracket \tau \rrbracket \to \llbracket \sigma \rrbracket$$

- All types are \triangleright -algebras with $\theta_{\sigma} : \triangleright \llbracket \sigma \rrbracket \to \llbracket \sigma \rrbracket$
- Interpreting terms $\llbracket t \rrbracket : \llbracket \Gamma \rrbracket \to \llbracket \sigma \rrbracket$

$$\llbracket \Gamma \vdash \mathsf{Y}_{\sigma} \ M \rrbracket(\gamma) = (\mathsf{fix}_{\llbracket \sigma \rrbracket})(\lambda x : \triangleright \llbracket \sigma \rrbracket . \theta_{\sigma}(\mathsf{next}(\llbracket M \rrbracket(\gamma))) \circledast x))$$

Interpreting PCF

Interpreting Types

$$[\![\mathbf{nat}]\!] \stackrel{\mathrm{def}}{=\!\!\!=\!\!\!=} L\mathbb{N}$$
$$[\![\tau \to \sigma]\!] \stackrel{\mathrm{def}}{=\!\!\!=\!\!\!=} [\![\tau]\!] \to [\![\sigma]\!]$$

- All types are \triangleright -algebras with $\theta_{\sigma} : \triangleright \llbracket \sigma \rrbracket \to \llbracket \sigma \rrbracket$
- Interpreting terms $\llbracket t \rrbracket : \llbracket \Gamma \rrbracket \to \llbracket \sigma \rrbracket$

$$\llbracket \Gamma \vdash \mathsf{Y}_{\sigma} \ M \rrbracket(\gamma) = (\mathsf{fix}_{\llbracket \sigma \rrbracket})(\lambda x : \rhd \llbracket \sigma \rrbracket.\theta_{\sigma}(\mathsf{next}(\llbracket M \rrbracket(\gamma))) \circledast x))$$

can be thought of $\theta \circ \rhd \llbracket \pmb{M}
rbracket$

Interpreting PCF

Interpreting Types

$$\llbracket \mathbf{nat} \rrbracket \stackrel{\mathrm{def}}{=\!\!\!=\!\!\!=} L \mathbb{N}$$
$$\llbracket \tau \to \sigma \rrbracket \stackrel{\mathrm{def}}{=\!\!\!=\!\!\!=} \llbracket \tau \rrbracket \to \llbracket \sigma \rrbracket$$

- All types are \triangleright -algebras with $\theta_{\sigma} : \triangleright \llbracket \sigma \rrbracket \to \llbracket \sigma \rrbracket$
- Interpreting terms $\llbracket t \rrbracket : \llbracket \Gamma \rrbracket \to \llbracket \sigma \rrbracket$

$$\llbracket \Gamma \vdash \mathsf{Y}_{\sigma} \ M \rrbracket(\gamma) = (\mathsf{fix}_{\llbracket \sigma \rrbracket})(\lambda x : \triangleright \llbracket \sigma \rrbracket . \theta_{\sigma}(\mathsf{next}(\llbracket M \rrbracket(\gamma))) \circledast x))$$

Lemma

Let
$$\Gamma \vdash M : \sigma \to \sigma$$
 then $[\![Y_\sigma \ M]\!] = \delta_\sigma \circ [\![M(Y_\sigma \ M)]\!]$

Soundness

Theorem (Soundness)

Let M be a closed term of type τ , if $M \Downarrow^k v$ then $[\![M]\!](*) = \delta^k [\![v]\!](*)$

Outline

• Operational Semantics of PCF

Denotational Semantics

• Computational Adequacy

if
$$[\![M]\!](*) = \delta^k [\![v]\!](*)$$
 then $M \Downarrow^k v$

Discussion

Adequacy proved by (proof-relevant) logical relation

$$d \mathcal{R}_{\tau} M$$

Define $\mathcal{R}_{ au}$ by induction on au

Adequacy proved by (proof-relevant) logical relation

$$d \mathcal{R}_{\tau} M$$

Define $\mathcal{R}_{ au}$ by induction on au

$$\eta(v) \; \mathcal{R}_{\mathsf{nat}} \; M \stackrel{\mathrm{def}}{=\!\!\!=\!\!\!=} \; M \downarrow^0 v$$

$$\theta_{\mathsf{nat}}(r) \; \mathcal{R}_{\mathsf{nat}} \; M \stackrel{\mathrm{def}}{=\!\!\!=\!\!\!=} \; \Sigma M', M'' \colon \mathsf{Term}_{\mathsf{PCF}}.M \to^0_* M'$$

$$\mathsf{and} \; M' \to^1 M'' \; \mathsf{and} \; r \, \triangleright \mathcal{R}_{\mathsf{nat}} \; \mathsf{next}(M'')$$

$L\mathbb{N} \cong \mathbb{N} + \triangleright L\mathbb{N}$

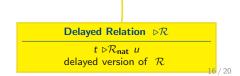
an element in this type is either of the form $\eta(v)$ or $\theta_{nat}(r)$

Adequacy proved by (proof-relevant) logical relation

$$d \mathcal{R}_{\tau} M$$

Define $\mathcal{R}_{ au}$ by induction on au

$$\begin{split} \eta(v) \ \mathcal{R}_{\text{nat}} \ M & \stackrel{\text{def}}{=\!\!\!=} \ M \Downarrow^0 v \\ \theta_{\text{nat}}(r) \ \mathcal{R}_{\text{nat}} \ M & \stackrel{\text{def}}{=\!\!\!=} \ \Sigma M', M'' \colon \texttt{Term}_{\texttt{PCF}}.M \to^0_* M' \\ & \text{and} \ M' \to^1 M'' \ \text{and} \ r \rhd \mathcal{R}_{\text{nat}} \ \text{next}(M'') \end{split}$$



Adequacy proved by (proof-relevant) logical relation

$$d \mathcal{R}_{\tau} M$$

Define $\mathcal{R}_{ au}$ by induction on au

Adequacy

Lemma (Fundamental Lemma)

Let $\Gamma \vdash t : \tau$, suppose $\Gamma \equiv x_1 : \tau_1, \dots, x_n : \tau_n$ and $t_i : \tau_i, \alpha_i : \llbracket \tau_i \rrbracket$ and $\alpha_i \ \mathcal{R}_{\llbracket \tau_i \rrbracket} \ t_i$ for $i \in \{1, \dots, n\}$, then $\llbracket t \rrbracket (\vec{\alpha}) \ \mathcal{R}_{\tau} \ t[\vec{t}/\vec{x}]$

Theorem (Computational Adequacy)

If M is a closed term of type $\operatorname{\mathbf{nat}}$ then $M \downarrow^k v$ iff $[\![M]\!](*) = \delta^k [\![v]\!]$.

Outline

• Operational Semantics of PCF

Denotational Semantics

• Computational Adequacy

Discussion

 $\operatorname{Set}^{\omega^{\operatorname{op}}}$ also models the topos logic.

The following is derivable in the non-proof-relevant topos logic:

$$\exists k. \exists v. Y_{\mathsf{nat}} (\lambda x. x) \Downarrow^k v$$

Proof (Sketch)

- The argument is by Guarded Recursion: assume $\triangleright (\exists k. \exists v. Y_{nat} (\lambda x. x) \Downarrow^k v)$
- by property of $\operatorname{Set}^{\omega^{\operatorname{op}}} \exists k. \exists v. \triangleright (\mathsf{Y}_{\mathsf{nat}} (\lambda x. x) \Downarrow^k v)$
- which implies $\exists k. \exists v. Y_{\mathsf{nat}} \ (\lambda x. x) \Downarrow^{k+1} v$

$$\sum k.\sum v. Y_{\mathbf{nat}} \lambda x. x \downarrow^k v$$
 is not globally inhabited

 $\operatorname{Set}^{\omega^{\operatorname{op}}}$ also models the topos logic.

The following is derivable in the non-proof-relevant topos logic:

$$\exists k. \exists v. Y_{\mathsf{nat}} (\lambda x. x) \Downarrow^k v$$

Proof (Sketch)

- The argument is by Guarded Recursion: assume $\triangleright (\exists k. \exists v. Y_{nat} (\lambda x. x) \Downarrow^k v)$
- by property of $\operatorname{Set}^{\omega^{\operatorname{op}}} \exists k. \exists v. \triangleright (\mathsf{Y}_{\mathsf{nat}} (\lambda x. x) \Downarrow^k v)$
- which implies $\exists k.\exists v. Y_{\mathsf{nat}} (\lambda x. x) \Downarrow^{k+1} v$

$$\sum k.\sum v. Y_{\text{nat}} \lambda x. x \Downarrow^k v$$
 is not globally inhabited

 $\operatorname{Set}^{\omega^{\operatorname{op}}}$ also models the topos logic.

The following is derivable in the non-proof-relevant topos logic:

$$\exists k. \exists v. Y_{\mathsf{nat}} (\lambda x. x) \Downarrow^k v$$

Proof (Sketch)

- The argument is by Guarded Recursion: assume $\triangleright (\exists k. \exists v. Y_{nat} (\lambda x. x) \Downarrow^k v)$
- by property of Set $^{\omega^{op}} \exists k. \exists v. \triangleright (Y_{nat} (\lambda x. x) \downarrow^k v)$
- which implies $\exists k. \exists v. Y_{nat} (\lambda x. x) \downarrow^{k+1} v$

$$\sum k.\sum v. Y_{\text{nat}} \lambda x.x \Downarrow^k v$$
 is not globally inhabited

 $\mathtt{Set}^{\omega^{\mathtt{op}}}$ also models the topos logic.

The following is derivable in the non-proof-relevant topos logic:

$$\exists k. \exists v. Y_{\mathsf{nat}} (\lambda x. x) \Downarrow^k v$$

Proof (Sketch)

- The argument is by Guarded Recursion: assume $\triangleright (\exists k. \exists v. Y_{\text{nat}} (\lambda x. x) \Downarrow^k v)$
- by property of Set $\omega^{op} \exists k. \exists v. \triangleright (\mathsf{Y}_{\mathsf{nat}} (\lambda x. x) \Downarrow^k v)$
- which implies $\exists k. \exists v. Y_{nat} (\lambda x. x) \downarrow^{k+1} v$

$$\sum k.\sum v. Y_{nat} \lambda x.x \downarrow^k v$$
 is not globally inhabited

Conclusions

We presented a model for PCF that is adequate w.r.t. the operational semantics.

The work has been carried out entirely in guarded type theory:

- Operational Semantics with explicit step-indexing is synchronised with the time steps in the type theory
- Denotational semantics with proof of adequacy

Main message

Guarded type theory as a meta theory for denotational semantics of programming languages.

Future work

- Using the model to reason about contextual equivalence
- FPC in guarded type theory

Thanks!

Conclusions

We presented a model for PCF that is adequate w.r.t. the operational semantics.

The work has been carried out entirely in guarded type theory:

- Operational Semantics with explicit step-indexing is synchronised with the time steps in the type theory
- Denotational semantics with proof of adequacy

Main message

Guarded type theory as a meta theory for denotational semantics of programming languages.

Future work

- Using the model to reason about contextual equivalence
- FPC in guarded type theory

Thanks!