The Scott Model of PCF in Univalent Type Theory

Tom de Jong

University of Birmingham, United Kingdom

TYPES, June 2019

UNIVERSITY OF BIRMINGHAM

I would like to thank Martín Escardó for suggesting and supervising this project.

I have also benefited from Benedikt Ahrens's support and his help with UniMath.

Background

- PCF is a typed programming language with a fixed point combinator. Its types:
  - ι for natural numbers;
  - $\sigma \Rightarrow \tau$ for functions.

Because of the fixed point combinator, a "standard" set-theoretic interpretation will not work (i.e. one where function types are interpreted as exponentials in Set).

We call a directed complete poset with a least element dcpo with bottom. A map between dcpos (with bottom) is continuous if it preserves directed suprema. The point is that such maps have fixed points.
The continuous maps between two dcpos with bottom form another dcpo with bottom with the pointwise ordering.

Background

- PCF is a typed programming language with a fixed point combinator. Its types:
  - ι for natural numbers;
  - $\sigma \Rightarrow \tau$ for functions.

  Examples of PCF terms:
  - $\underline{0}, \underline{1}, \underline{2}, \ldots : \iota$;
  - $\mathsf{pred} : \iota \Rightarrow \iota$ and $\mathsf{pred}\,\underline{5} : \iota$;
  - $\mathsf{fix}_\sigma : (\sigma \Rightarrow \sigma) \Rightarrow \sigma$.

Because of the fixed point combinator, a "standard" set-theoretic interpretation will not work (i.e. one where function types are interpreted as exponentials in Set).

We call a directed complete poset with a least element dcpo with bottom. A map between dcpos (with bottom) is continuous if it preserves directed suprema. The point is that such maps have fixed points.
The continuous maps between two dcpos with bottom form another dcpo with bottom with the pointwise ordering.

Background

- PCF is a typed programming language with a fixed point combinator. Its types:
  - ι for natural numbers;
  - $\sigma \Rightarrow \tau$ for functions.

  Examples of PCF terms:
  - $\underline{0}, \underline{1}, \underline{2}, \ldots : \iota$;
  - pred : $\iota \Rightarrow \iota$ and pred $\underline{5} : \iota$;
  - $\mathsf{fix}_\sigma : (\sigma \Rightarrow \sigma) \Rightarrow \sigma$.

  Examples of reduction:
  - pred $\underline{n+1} \rhd^* \underline{n}$;
  - fix $f \rhd^* f\,(\mathsf{fix}\, f)$.

Because of the fixed point combinator, a "standard" set-theoretic interpretation will not work (i.e. one where function types are interpreted as exponentials in Set).

We call a directed complete poset with a least element dcpo with bottom. A map between dcpos (with bottom) is continuous if it preserves directed suprema. The point is that such maps have fixed points.
The continuous maps between two dcpos with bottom form another dcpo with bottom with the pointwise ordering.

Background

- PCF is a typed programming language with a fixed point combinator. Its types:
  - ι for natural numbers;
  - $\sigma \Rightarrow \tau$ for functions.

  Examples of PCF terms:
  - $\underline{0}, \underline{1}, \underline{2}, \ldots : \iota$;
  - pred : $\iota \Rightarrow \iota$ and pred $\underline{5} : \iota$;
  - $\mathsf{fix}_\sigma : (\sigma \Rightarrow \sigma) \Rightarrow \sigma$.

  Examples of reduction:
  - pred $\underline{n+1} \rhd^* \underline{n}$;
  - fix $f \rhd^* f$ (fix f).
- Scott model
  - interpret PCF types as directed complete posets with a least element;
  - for function types use continuous maps.

Because of the fixed point combinator, a "standard" set-theoretic interpretation will not work (i.e. one where function types are interpreted as exponentials in Set).

We call a directed complete poset with a least element dcpo with bottom. A map between dcpos (with bottom) is continuous if it preserves directed suprema. The point is that such maps have fixed points.
The continuous maps between two dcpos with bottom form another dcpo with bottom with the pointwise ordering.

Results

- Scott model of PCF in constructive predicative univalent type theory, using the lifting monad [EK17; Kna18].
- Soundness:

$$\text{if } s \rhd^* t, \text{ then } [\![s]\!] = [\![t]\!].$$

- Computational adequacy:

$$\text{if } [\![s]\!] = [\![\underline{n}]\!], \text{then } s \rhd^* \underline{n}.$$

Since PCF has a fixed-point combinator, it has non-termination. This is what makes a constructive type-theoretic semantics challenging.
Our constructive approach uses the lifting monad, building on work by Escardó and Knapp, extending the partial map classifier from topos theory.
Directed completeness is interesting from a predicative viewpoint.

Just as in the classical case, we can prove that the Scott model has desirable properties.
Soundness expresses that the model (denotational semantics) respects the reduction relation (operational semantics). If a term s in PCF reduces ("computes") to a term t, then the interpretations of s and t are equal in the model.
Computational adequacy can be regarded as a partial converse to soundness. (A lack of function extensionality in PCF makes a full converse impossible.) An interesting consequence of computational adequacy is that it allows one to reason semantically about termination (reduction to a numeral) in PCF.

Why construct the Scott model in this setting?

- Validate partiality via lifting in constructive type theory.
  Other approaches need some form of countable choice [CUV17] or higher inductive-inductive types [ADK17].
- Show that we can have Scott model and (a form of) directed completeness in a predicative framework.
- Related work [BKV09]:
  - based on Capretta's delay monad;
  - only ω-complete posets;
  - uses Coq's impredicative `Prop` universe;
  - works with setoids.

Coquand et al. have shown that countable choice is independent over univalent type theory:

- [CMR17];
- [Coq18].

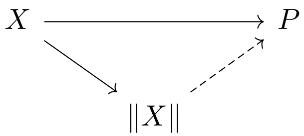
In a predicative framework, we cannot formulate directed completeness using subsets, so instead we work with families.
Directed completeness will then be relative to a universe $\mathcal{U}$: a poset is directed complete w.r.t. $\mathcal{U}$ if it is directed complete with w.r.t. all families indexed by types in $\mathcal{U}$.

Rather than setoids, we work with the identity types, as usual in univalent type theory. Quotienting the setoids is problematic as it needs choice to yield another monad[2].

Type-theoretic framework

We work in intensional Martin-Löf Type Theory with

- $\prod$- and $\sum$-types;
- general inductive types, including: 0, 1 and N;
- function extensionality;
- propositional extensionality;
- propositional truncation
  for every type X, there is a proposition $\|X\|$ and a map $X \to \|X\|$, such that every map from X to a *proposition* P factors through it.



Function extensionality states that pointwise equal functions are equal.

Propositional extensionality states that logically equivalent propositions are equal.

The map $\|X\| \to P$ is unique, because of function extensionality and the fact that P is a proposition.

We work predicatively, so we do not assume propositional resizing.

Our framework and univalence

- Propositional and function extensionality are consequences of univalence.
  We do not need full univalence.
- We emphasise the importance of the idea of h-levels, which is fundamental to univalent type theory.

We do not need full univalence, because the types under consideration are all propositions and sets (dcpos).

Arguably, univalent type theory is much more about the concept of h-levels than about the univalence axiom.

Directed complete families

Since we work predicatively, we consider families, rather than power sets.

Definition

Let $(P, \leq)$ be a poset. A family $\alpha : I \to P$ is directed if it is inhabited (that is, $\|I\|$) and $\prod_{i,j:I} \|\sum_{k:I} \alpha_i \leq \alpha_k \times \alpha_j \leq \alpha_k\|$.

Observe that this is property, rather than structure.

Definition

A poset $(P, \leq)$ is $\mathcal{U}$-directed-complete if it has suprema for every directed family indexed by a type in $\mathcal{U}$.

Here $\mathcal{U}$ is a (type) universe.

The lifting of a type

Fix a type universe $\mathcal{T}$ and write $\Omega \overset{\text{def}}{\equiv} \sum_{P:\mathcal{T}} \text{isaprop}(P)$.

Definition

The lifting $\mathcal{L}(X)$ of a type X is defined as

$$\mathcal{L}(X) \overset{\text{def}}{\equiv} \sum_{P:\Omega} P \to X.$$

Definition

We can embed a type into its lifting: $\eta_X : X \to \mathcal{L}(X)$, $x \mapsto (1, \lambda t.x)$.

Definition

The element $\bot \overset{\text{def}}{\equiv} (0, \text{fromempty}_X) : \mathcal{L}(X)$ represents "undefined".

With the law of excluded middle, this is all, i.e. $\mathcal{L}(X) \simeq X + 1$.

The lifting of a set is a dcpo with bottom

Lemma

If X is a set, then so is $\mathcal{L}(X)$.

Definition

Let isdefined : $\mathcal{L}(X) \to \Omega$ be $(P, \varphi) \mapsto P$.

Definition

Let X be a set. Define a partial order $\sqsubseteq$ on $\mathcal{L}(X)$ by:

$$l \sqsubseteq m \overset{\text{def}}{\equiv} \text{isdefined}\, l \to l = m.$$

Theorem (Knapp, Escardó)

For X a set, $(\mathcal{L}(X), \sqsubseteq)$ is $\mathcal{U}_0$-directed complete and $\bot$ is its least element.

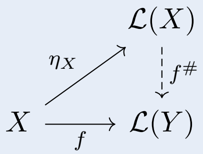
Here $\mathcal{U}_0$ denotes the least universe.

The lifting as a monad

The "monad" bumps universe levels. This is not a problem, however, as the Kleisli triple equations can be type checked and proved nonetheless.

Theorem (Knapp, Escardó, dJ)

Any $f : X \to \mathcal{L}(Y)$ *can be extended to* $f^{\#} : \mathcal{L}(X) \to \mathcal{L}(Y)$.



This extension is continuous and $\left(\mathcal{L}, \eta, (-)^{\#}\right)$ *satisfies the Kleisli triple equations.*

The Scott model using the lifting monad (PCF types)

Definition (Interpreting PCF types)

The base type (for natural numbers) is interpreted as

$$[\![\iota]\!] \stackrel{\text{def}}{\equiv} \mathcal{L}(\mathsf{N}).$$

The function types are interpreted using the dcpo of continuous maps:

$$[\![\sigma \Rightarrow \tau]\!] \stackrel{\text{def}}{\equiv} [\![\tau]\!]^{[\![\sigma]\!]}$$

The continuous maps between two dcpos with bottom form another dcpo with bottom with the pointwise ordering.

The Scott model using the lifting monad (PCF terms)

Definition (Interpreting PCF terms)

The PCF terms are interpreted using the lifting monad, e.g.

- $[\![\underline{n}]\!] \overset{\text{def}}{\equiv} \eta(n)$ for every $n : \mathsf{N}$;
- $[\![\mathsf{pred}]\!] \overset{\text{def}}{\equiv} \mathcal{L}(P)$, where P is the predecessor map on N.

As usual, $[\![\mathsf{fix}]\!] : [\![(\sigma \Rightarrow \sigma) \Rightarrow \sigma]\!]$ is defined using the directed supremum of the N-indexed family $\bot \leq f(\bot) \leq f^2(\bot) \leq \dots$

$$[\![\mathsf{fix}]\!](f) \overset{\text{def}}{\equiv} \bigsqcup_{n:\mathsf{N}} f^n(\bot).$$

Theorem

The model is sound and computationally adequate.

Proving soundness and computational adequacy

- Soundness is proved using a standard induction on the rules of the reduction relation $\triangleright^*$.
  Also use the Kleisli triple equations, e.g. to reduce $f^{\#}(\eta(n))$ to $f(n)$.
- Computational adequacy is proved using a standard logical relation.

To avoid dealing with variables, we work with a combinatory version of PCF and therefore a combinatory version of the operational semantics.

The logical relation is necessary, because the statement of computational adequacy does not allow for a direct proof by induction.

Use of propositional truncation

- To define directed families:

$$\|I\| \times \prod_{i,j:I} \left\| \sum_{k:I} \alpha_i \leq \alpha_k \times \alpha_j \leq \alpha_k \right\|.$$

- To define a propositional valued reduction relation $\triangleright^*$.
- To define the supremum $\bigsqcup \alpha$ of a directed family $\alpha : I \to \mathcal{L}(X)$:

$$\text{isdefined}\left(\bigsqcup \alpha\right) \stackrel{\text{def}}{\equiv} \left\| \sum_{i:I} \text{isdefined}(\alpha_i) \right\|.$$

A significant role in our development is played by the propositional truncation. The map to X that is the second component of $\bigsqcup \alpha$ is constructed using [Kra+17, Theorem 5.4], which says that a constant map $f : X \to Y$ to a set factors through $\|X\|$.

Universe levels of the lifting

- Recall $\mathcal{L}(X) \overset{\text{def}}{\equiv} \sum_{P:\Omega}(P \to X)$ with $\Omega \overset{\text{def}}{\equiv} \sum_{P:\mathcal{T}} \mathsf{isaprop}(P)$. Hence,

$$\mathcal{L}(X : \mathcal{U}) : \mathcal{T}^{+} \sqcup \mathcal{U}.$$

- From now on, take $\mathcal{T} \overset{\text{def}}{\equiv} \mathcal{U}_0$.
  Note:

$$\mathcal{L}(\mathsf{N} : \mathcal{U}_0) : \mathcal{U}_1$$

  and

$$\sqsubseteq : \mathcal{L}(\mathsf{N}) \to \mathcal{L}(\mathsf{N}) \to \mathcal{U}_1.$$

Here $\mathcal{U}^{+}$ denotes the successor universe of $\mathcal{U}$ and $\mathcal{U} \sqcup \mathcal{V}$ denotes the least universe bigger than $\mathcal{U}$ and $\mathcal{V}$.
Furthermore, $\mathcal{U}_0$ is the least universe and $\mathcal{U}_1$ is a short-hand for $(\mathcal{U}_0)^{+}$.

Universe levels and dcpos

- Write $\mathcal{W}$-DCPO$_{\mathcal{U},\mathcal{V}}$ for the type of dcpos with bottom with
  - underlying type in $\mathcal{U}$;
  - partial order taking values in $\mathcal{V}$;
  - suprema for directed $\mathcal{W}$-families.

  So,

$$\mathcal{L}(\mathsf{N}) : \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}.$$

We are interested in these universe levels, because to interpret PCF types like $\iota \Rightarrow \iota \Rightarrow \iota \Rightarrow \iota$, we need to be able to iterate the exponential.

For the Scott model, we need $\mathcal{U}_0$-directed completeness, because we (only) need N-indexed families.

In general,

$$(-)^{(-)} : \mathcal{W}\text{-DCPO}_{\mathcal{U},\mathcal{V}} \to \mathcal{W}\text{-DCPO}_{\mathcal{U}',\mathcal{V}'} \to \mathcal{W}\text{-DCPO}_{\mathcal{W}^+ \sqcup \mathcal{U} \sqcup \mathcal{V} \sqcup \mathcal{U}' \sqcup \mathcal{V}', \mathcal{U} \sqcup \mathcal{V}'}.$$

In particular,

$$(-)^{(-)} : \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1} \to \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1} \to \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}.$$

These universe levels have been checked using Agda (see slide after Conclusion).

Universe levels and dcpos

- Write $\mathcal{W}$-DCPO$_{\mathcal{U},\mathcal{V}}$ for the type of dcpos with bottom with
  - underlying type in $\mathcal{U}$;
  - partial order taking values in $\mathcal{V}$;
  - suprema for directed $\mathcal{W}$-families.

  So,
  $$\mathcal{L}(\mathsf{N}) : \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}.$$
- The type of continuous functions mentions all directed families, so taking exponentials potentially increases universe levels.

We are interested in these universe levels, because to interpret PCF types like $\iota \Rightarrow \iota \Rightarrow \iota \Rightarrow \iota$, we need to be able to iterate the exponential.

For the Scott model, we need $\mathcal{U}_0$-directed completeness, because we (only) need N-indexed families.

In general,

$$(-)^{(-)} : \mathcal{W}\text{-DCPO}_{\mathcal{U},\mathcal{V}} \to \mathcal{W}\text{-DCPO}_{\mathcal{U}',\mathcal{V}'} \to \mathcal{W}\text{-DCPO}_{\mathcal{W}^+ \sqcup \mathcal{U} \sqcup \mathcal{V} \sqcup \mathcal{U}' \sqcup \mathcal{V}', \mathcal{U} \sqcup \mathcal{V}'}.$$

In particular,

$$(-)^{(-)} : \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1} \to \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1} \to \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}.$$

These universe levels have been checked using Agda (see slide after Conclusion).

Universe levels and dcpos

- Write $\mathcal{W}$-DCPO$_{\mathcal{U},\mathcal{V}}$ for the type of dcpos with bottom with
  - underlying type in $\mathcal{U}$;
  - partial order taking values in $\mathcal{V}$;
  - suprema for directed $\mathcal{W}$-families.

  So,
  $$\mathcal{L}(\mathsf{N}) : \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}.$$
- The type of continuous functions mentions all directed families, so taking exponentials potentially increases universe levels.
- Nonetheless,
  $$[\![-]\!] : \text{PCF-types} \to \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}.$$

We are interested in these universe levels, because to interpret PCF types like $\iota \Rightarrow \iota \Rightarrow \iota \Rightarrow \iota$, we need to be able to iterate the exponential.

For the Scott model, we need $\mathcal{U}_0$-directed completeness, because we (only) need N-indexed families.

In general,

$$(-)^{(-)} : \mathcal{W}\text{-DCPO}_{\mathcal{U},\mathcal{V}} \to \mathcal{W}\text{-DCPO}_{\mathcal{U}',\mathcal{V}'} \to \mathcal{W}\text{-DCPO}_{\mathcal{W}^+ \sqcup \mathcal{U} \sqcup \mathcal{V} \sqcup \mathcal{U}' \sqcup \mathcal{V}', \mathcal{U} \sqcup \mathcal{V}'}.$$

In particular,

$$(-)^{(-)} : \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1} \to \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1} \to \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}.$$

These universe levels have been checked using Agda (see slide after Conclusion).

Conclusion

Scott model of PCF in constructive predicative univalent type theory:

- uses the lifting monad
- soundness and computational adequacy
- important role for propositional truncation
- careful handling of universes

The point is that lifting works: we can define the Scott model of PCF in constructive predicative univalent type theory and prove fundamental properties like soundness and computational adequacy.
We deviate from the classical treatment in

- using the lifting monad from Knapp and Escardó;
- our use of the propositional truncation;
- a careful treatment of directed completeness and universe levels (because of the predicative setting).

Conclusion

Scott model of PCF in constructive predicative univalent type theory:

- uses the lifting monad
- soundness and computational adequacy
- important role for propositional truncation
- careful handling of universes

Thank you for your attention!

The point is that lifting works: we can define the Scott model of PCF in constructive predicative univalent type theory and prove fundamental properties like soundness and computational adequacy.
We deviate from the classical treatment in

- using the lifting monad from Knapp and Escardó;
- our use of the propositional truncation;
- a careful treatment of directed completeness and universe levels (because of the predicative setting).

Full paper & formalisation

Tom de Jong. *The Scott model of PCF in univalent type theory*. June 2019. arXiv: 1904.09810 [math.LO].

Any comments, questions, feedback are most welcome!

Coq formalisation using UniMath:
`https://github.com/tomdjong/UniMath/tree/paper/UniMath`

Agda formalisation (to check the universe levels) using Martín Escardó's Agda development:
`https://github.com/martinescardo/TypeTopology`

At present, it is impossible to check universe levels in UniMath. Therefore, we have redone part of our development in Agda.

References I

[ADK17] Thorsten Altenkirch, Nils Anders Danielsson and Nicolai Kraus. 'Partiality, Revisited: The Partiality Monad as a Quotient Inductive-Inductive Type'. In: *Foundations of Software Science and Computation Structures*. Ed. by Javier Esparza and Andrzej S. Murawski. Springer Berlin Heidelberg, 2017, pp. 534–549. DOI: 10.1007/978-3-662-54458-7_31.

[BKV09] Nick Benton, Andrew Kennedy and Carsten Varming. 'Some Domain Theory and Denotational Semantics in Coq'. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2009, pp. 115–130. DOI: 10.1007/978-3-642-03359-9_10.

[CUV17] James Chapman, Tarmo Uustalu and Niccolò Veltri. 'Quotienting the delay monad by weak bisimilarity'. In: *Mathematical Structures in Computer Science* 29.1 (2017), pp. 67–92. DOI: 10.1017/s0960129517000184.

References II

[Coq18] Thierry Coquand. 'A survey of constructive presheaf models of univalence'. In: *SIGLOG News* 5.3 (2018), pp. 54–65. DOI: 10.1145/3242953.3242962.

[CMR17] Thierry Coquand, Bassel Mannaa and Fabian Ruch. 'Stack semantics of type theory'. In: *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. June 2017. DOI: 10.1109/LICS.2017.8005130.

[EK17] Martín H. Escardó and Cory M. Knapp. 'Partial Elements and Recursion via Dominances in Univalent Type Theory'. In: *26th EACSL Annual Conference on Computer Science Logic (CSL 2017)*. Ed. by Valentin Goranko and Mads Dam. Vol. 82. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017, 21:1–21:16. DOI: 10.4230/LIPIcs.CSL.2017.21.

References III

[Kna18] Cory Knapp. 'Partial Functions and Recursion in Univalent Type Theory'. PhD thesis. School of Computer Science, University of Birmingham, June 2018.

[Kra+17] Nicolai Kraus et al. 'Notions of Anonymous Existence in Martin-Löf Type Theory'. In: *Logical Methods in Computer Science* 13 (1 2017). DOI: 10.23638/LMCS-13(1:15)2017.