Partiality and Non-Termination in Type Theory

Niccolò Veltri

Department of Software Science
Tallinn University of Technology

EWSC'17, 08 March 2017

Introduction

- In type theory, all functions are total.
  $((1 \to 0) +$ "proposition as types" $\rightsquigarrow$ inconsistency)
- In this talk:
  1. A general technique for representing partial functions in type theory.
     A type-theoretic reformulation of ideas by Rosolini, Mulry, Hyland, Taylor, etc.
  2. Some examples, with particular emphasis on the partiality monad, used for implementing possibly non-terminating computations.

Notation and some definitions

- Type theory $\leftrightsquigarrow$ 0-truncated homotopy type theory.
- X is 0-truncated $\leftrightsquigarrow$ X is a set $\leftrightsquigarrow$ $\Pi x, y : X. \Pi p, q : x = y. p = q)$.
- X is –1-truncated $\leftrightsquigarrow$ X is a (mere) proposition $\leftrightsquigarrow$ $\Pi x, y : X. x = y$.

$$\mathsf{isProp}\, X \equiv \Pi x, y : X. x = y$$

- The type of propositions: $\Omega \equiv \Sigma X : \mathcal{U}. \mathsf{isProp}\, X$. ($\mathcal{U} \leftrightsquigarrow$ Type, $\mathcal{U} \leftrightsquigarrow$ Set, $\Omega \leftrightsquigarrow$ Prop)
- For elements $(x, \_) : X : \Omega$ we write $x : X$.

Classical partiality

- Two equivalent techniques:
  1. A partial function $f : X \rightharpoonup Y$ is a function $f : X \to Y + 1$;
  2. A partial function $f : X \rightharpoonup Y$ is a function $f : \Sigma x : X.\, (U\, x = 1) \to Y$ for a predicate $U : X \to 2$.
- Using univalence:

$$(X \to Y + 1)$$
$$=$$
$$\Sigma U : X \to 2.\, (\Sigma x : X.(U\, x = 1) \to Y)$$

- Classically, it is a satisfactory representation of all partial functions.
- Constructively, we cannot represent partial functions with undecidable domain of definedness.

Rosolini's dominances

- The type 2 in the previous slide should be replaced by a less restricting set of truth values.
  $\Rightarrow$ Rosolini's notion of dominance.
- A type D is a dominance if it comes with the following data:
  - an injective map $[\![-]\!] : D \to \Omega$
  - an element $1_D : D$
  - an operation $\Sigma_D : \Pi X : D.\, ([\![X]\!] \to D) \to D$
  - a proof of $[\![1_D]\!] = 1$
  - a proof of $[\![\Sigma_D\, X\, Y]\!] = \Sigma x : [\![X]\!].\, [\![Y\, x]\!]$, for all $X : D$ and $Y : [\![X]\!] \to D$

Rosolini's dominances: examples

- The smallest dominance is the unit type 1
- The biggest dominance is Ω
- 2
- The Sierpinski set S (that we construct explicitly later). It has two elements, $1 : \mathrm{S}$ representing termination and $0 : \mathrm{S}$ representing non-termination. Every $U : X \to \mathrm{S}$ is a semidecidable predicate on X.
- In examples above, the interpretation morphism is defined as $[\![X]\!] \equiv (X = 1_D)$.
- Every monad $j : \Omega \to \Omega$ specifies a dominance $D_j \equiv \Sigma X : \Omega.\, (j\, X = X)$.
  An example: $j\, X \equiv \neg\neg X$.

Back to partial maps

- Fix a dominance D.
- A D-partial function $f : X \rightharpoonup_D Y$ is a function $f : \Sigma x : X.\, [\![U\, x]\!] \to Y$ for a certain predicate $U : X \to D$.
- Composition of partial functions? Total functions? The dominance data takes care of everything.
- Remember
$$(X \to Y + 1) = (X \rightharpoonup_2 Y)$$
- We have a similar equality for a general dominance D
$$(X \to \mathrm{P}_D Y) = (X \rightharpoonup_D Y)$$
where $\mathrm{P}_D Y \equiv \Sigma U : D.\, ([\![U]\!] \to Y)$, the partial map classifier associated to the dominance D. P_D is a monad, since D is a dominance.

Some partial map classifiers

- If $D \equiv 1$, then $\mathsf{P}_1 X = X$, i.e. P_1 is the identity monad.
- If $D \equiv 2$, then $\mathsf{P}_2 X = X + 1$, i.e. P_2 is the maybe monad.
- If D is the Sierpinski set S, then P_S is the monad delivering free ω-complete pointed partial orders, i.e. the so-called partiality monad. It admits alternative concrete implementations:
  - Chapman et al. : quotiented delay monad.
    (Requires semi-classical assumptions, such as countable choice)
  - Altenkirch et al. : partiality monad.
    (Constructed employing higher inductive-inductive types)
- A map $f : X \to \mathsf{P}_\mathsf{S} Y$ is a possibly non-terminating computation with input in X and output in Y.

The Sierpinski set as a HIT

- 0-constructor

$$\frac{}{1_S : S} \qquad \frac{}{0_S : S} \qquad \frac{x : S \quad y : S}{x \wedge y : S} \qquad \frac{s : \mathbb{N} \to S}{\bigvee s : S}$$

- 1-constructor

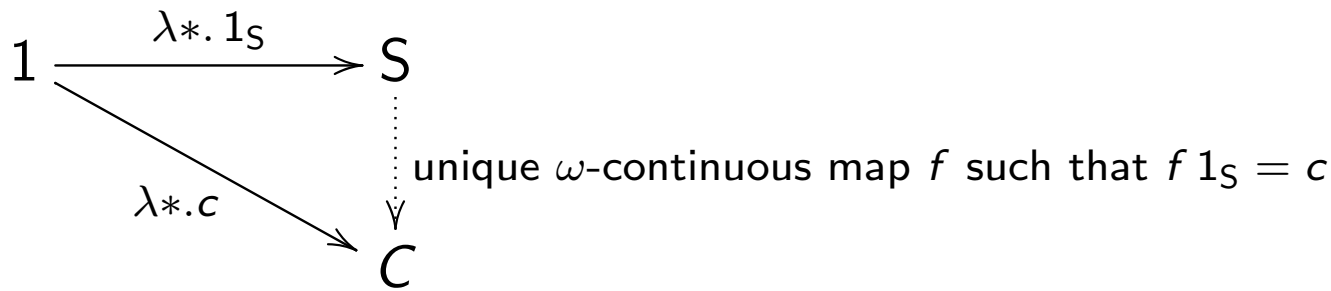
$$\frac{}{x \wedge y = y \wedge x} \qquad \frac{}{x \wedge (y \wedge z) = (x \wedge y) \wedge z} \qquad \frac{}{x \wedge x = x}$$

$$\frac{}{x \wedge 1_S = x} \qquad \frac{}{x \wedge 0_S = 0_S} \qquad \frac{}{\Pi n : \mathbb{N}.\, s\, n \wedge \bigvee s = s\, n}$$

$$\frac{}{\bigvee s \wedge x = \bigvee(\lambda n.\, s\, n \wedge x)} \qquad \text{(the 0-truncation constructor)}$$

The Sierpinski set as a HIT

- Previous slide: algebraic theory of σ-frames.
- A σ-frame is a partial order with finite meets and countable joins, and meets distribute over joins.
- By construction, S is the initial σ-frame, i.e. there exists a unique structure-preserving map from S into any other σ-frame.
- Theorem: initial σ-frame coincides with the free ω-complete pointed partial order on the unit type.



- The partiality monad can be defined as

$$\mathsf{P}_{\mathsf{S}}X \equiv \Sigma v : \mathsf{S}.\,(v = 1_{\mathsf{S}} \to X)$$

Conclusions

- We showed how to represent partial functions in type theory, using topos-theoretical techniques developed by Rosolini, Mulry, Hyland, Taylor, etc.
- Emphasis on the partiality monad, important for the representation of possibly non-terminating computations
- We gave a construction of the Sierpinski set, and consequently of the partiality monad, which employs standard higher inductive types. This could provide a direct implementation of the partiality monad in the Coq's HoTT library.