Normalisation by Evaluation for Dependent Types*

Thorsten Altenkirch and Ambrus Kaposi

University of Nottingham, United Kingdom
{txa,auk}@cs.nott.ac.uk

Abstract

We prove normalisation for a basic dependent type theory using the technique of normalisation by evaluation (NBE). NBE works by evaluating the syntax into a model and then computing normal forms by quoting semantic objects into normal terms. As model we use the syntax glued together with a proof-relevant logical predicate. The syntax is defined internally in type theory as a quotient-inductive type. We use a typed presentation of the syntax, we don't mention preterms. Parts of the construction are formalised in Agda. A full paper will be presented at FSCD 2016.

Specifying normalisation

We denote the type of well typed terms of type A in context Γ by $\mathsf{Tm}\,\Gamma\,A$. This type is defined as a quotient inductive inductive type (QIIT, see [3]): in addition to normal constructors for terms such as lam and app, it also has equality constructors e.g. expressing the β computation rule for functions. An equality $t \equiv_{\mathsf{Tm}\,\Gamma\,A} t'$ expresses that t and t' are convertible. Typed normal forms are denoted $\mathsf{Nf}\,\Gamma\,A$ and are defined mutually with neutral terms $\mathsf{Ne}\,\Gamma\,A$ and the embedding $\ulcorner - \urcorner : \mathsf{Nf}\,\Gamma\,A \rightarrow \mathsf{Tm}\,\Gamma\,A$. Normalisation is given by a function norm which is an isomorphism[1]:

$$\text{completeness}\circlearrowleft \qquad \mathsf{norm}\downarrow \frac{\mathsf{Tm}\,\Gamma\,A}{\mathsf{Nf}\,\Gamma\,A} \uparrow \ulcorner - \urcorner \qquad \curvearrowright\text{stability}$$

Completeness says that normalising a term produces a term which is convertible to it: $t \equiv \ulcorner \mathsf{norm}\,t \urcorner$. Stability expresses that there is no redundancy in the type of normal forms: $n \equiv \mathsf{norm}\,\ulcorner n \urcorner$. The usual notion of soundness of the semantics, that is, if $t \equiv t'$ then $\mathsf{norm}\,t \equiv \mathsf{norm}\,t'$ is given by congruence of equality. The elimination rule for the QIIT of the syntax ensures that every function defined from the syntax respects the equality constructors.

NBE for simple types

NBE is one way to implement the above specification. It works by evaluating the syntax in a model and defining a quote function which turns semantic objects into normal terms. We follow the categorical approach to NBE as given by [2] for simple types. Here the model is a presheaf model over the category of renamings (objects are contexts, morphisms are lists of variables) and the interpretation of the base type is the set of normal terms at the base type.

The structure of the normalisation proof is given by the following diagram. It summarizes normalisation of a substitution into context Δ, a similar diagram can be given for terms.

$$\begin{tikzcd}
\mathsf{NE}_\Delta \arrow[r, "\mathsf{u}_\Delta"] \arrow[dr, "\ulcorner_\urcorner"'] & \boxed{\Sigma\,(\mathsf{TM}_\Delta \times [\![\Delta]\!])\,\mathsf{R}_\Delta} \arrow[r, "\mathsf{q}_\Delta"] \arrow[d, "\mathsf{proj}"] & \mathsf{NF}_\Delta \arrow[dl, "\ulcorner_\urcorner"] \\
& \mathsf{TM}_\Delta &
\end{tikzcd}$$

*Supported by USAF grant FA9550-16-1-0029.

[1] This might by surprising however it can be explained by the fact that the conversion relation is part of the equality structure on terms.

NE_Δ, TM_Δ and NF_Δ denote the Yoneda embeddings of neutral terms, terms and normal terms, respectively. These are all presheaves over the category of renamings where the action on objects returns lists of neutral terms Nes, substitutions Tms and lists of normal forms Nfs, respectively.

$$\mathsf{NE}_\Delta\,\Gamma := \mathsf{Nes}\,\Gamma\,\Delta \qquad\qquad \mathsf{TM}_\Delta\,\Gamma := \mathsf{Tms}\,\Gamma\,\Delta \qquad\qquad \mathsf{NF}_\Delta\,\Gamma := \mathsf{Nfs}\,\Gamma\,\Delta$$

The presheaf interpretation of the context Δ is denoted $[\![\Delta]\!]$. R_Δ denotes a binary logical relation at context Δ. This is a relation between the syntax TM and the presheaf semantics $[\![-]\!]$ and is equality at the base type. u_Δ denotes the unquote natural transformation and q_Δ is quotation. These are defined mutually by induction on contexts and types.

Normalisation of a substitution σ is given by evaluating it in the presheaf model $[\![\sigma]\!]$ and then using quote. It also needs the semantic counterpart of the identity substitution which is given by unquoting the identity substitution $\mathsf{u}_\Gamma\,\mathsf{id}$ and a witness of the logical relation which is given by the fundamental theorem for the logical relation R_σ.

$$\mathsf{norm}_\Delta\,(\sigma : \mathsf{TM}_\Delta\,\Gamma) : \mathsf{NF}_\Delta\,\Gamma := \mathsf{q}_\Delta\,\big(\sigma, [\![\sigma]\!]\,(\mathsf{u}_\Gamma\,\mathsf{id}), \mathsf{R}_\sigma\,(\mathsf{u}_\Gamma\,\mathsf{id})\big)$$

Completeness is given by commutativity of the right hand triangle. Stability can be proven by mutual induction on terms and normal forms.

NBE for dependent types

NBE has been extended to dependent types using untyped realizers [1] and a typed version has been given by [4] however without a proof of soundness. Our goal was to extend the categorical approach summarized in the previous section to dependent types. The straightforward generalisation does not work because there seems to be no way of defining unquote for Π. Here we need to define a semantic function which works for arbitrary inputs, not only those which are related to a term. It seems that we need to restrict the presheaf model to only contain such functions.

We solve this problem by merging the presheaf model $[\![-]\!]$ and the logical relation R into a proof-relevant logical predicate P. That is, we replace the boxed part of the above diagram by $\boxed{\Sigma\,\mathsf{TM}_\Delta\,\mathsf{P}_\Delta}$. In the presheaf model, the interpretation of the base type were normal forms of the base type, and the logical relation at the base type was equality of the term and the normal form (equality was proof irrelevant, hence the logical relation was proof irrelevant). In our case, the logical predicate at the base type says that there exists a normal form which is equal to the term, hence it needs to be proof relevant. It can be seen as an instance of categorical glueing.

References

[1] Andreas Abel. *Normalization by Evaluation: Dependent Types and Impredicativity.* PhD thesis, Habilitation, Ludwig-Maximilians-Universität München, 2013.

[2] Thorsten Altenkirch, Martin Hofmann, and Thomas Streicher. Categorical reconstruction of a reduction free normalization proof. In David Pitt, David E. Rydeheard, and Peter Johnstone, editors, *Category Theory and Computer Science*, LNCS 953, pages 182–199, 1995.

[3] Thorsten Altenkirch and Ambrus Kaposi. Type theory in type theory using quotient inductive types. In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL 2016, pages 18–29, New York, NY, USA, 2016. ACM.

[4] Nils Anders Danielsson. A formalisation of a dependently typed language as an inductive-recursive family. In *Types for Proofs and Programs*, pages 93–109. Springer, 2006.