Proving Strong Normalization of CC by Modifying Realizability Semantics

Thorsten Altenkirch

Department of Computer Science, Chalmers University of Technology
412 64 Gothenburg, Sweden

1 Introduction

We will outline a strong normalization argument for the Calculus of Constructions (CC) which is obtained by modifying a realizability interpretation (the D-set or ω-set model [1]). By doing so we pursue two goals:

- We want to illustrate how *semantics can be used to prove properties of syntax.*
- We present a simple and extensible SN proof for CC. An example of such an extension is a system with inductive types and *large eliminations.*

This presentation corresponds to a part of the author's PhD thesis [Alt93a], a preliminary version has been presented in [Alt93b]. In my thesis I present a more general soundness result for a class of models for CC — CC-structures — from which the strong normalization argument can be derived as an instance. Here we shall restrict ourselves to the reasoning needed for the strong normalization proof.

The proof that every term typable in the calculus of constructions is strongly normalizing is known to be notoriously difficult. The original proof in Coquand's PhD thesis [Coq85] contained a bug which was fixed in [CG90] by using a Kripke-style interpretation of contexts. Although this solves the original problem the proof remains quite intricate due to the use of typed terms and contexts. Another construction is due to Geuvers and Nederhof (see [Geu93], p. 168), who define a forgetful, reduction-preserving map from CC to F^ω. Thereby, they reduce the problem to strong normalization for F^ω, which can be shown using the usual Girard-Tait method. The main problem with this construction is that it is not all clear, how this argument can be extended to a system with large eliminations (e.g. see [Wer92]), this is a system which allows the definition of a dependent type by primitive recursion. As an example consider the recursive definition of

[1] See [Ehr89, Str89, Str91].

a type $T : \mathrm{Nat} \to \mathrm{Set}$:

$$\begin{aligned} T(0) &= A \\ T(n+1) &= Tn \to Tn \end{aligned}$$

where $A : \mathrm{Set}$ is arbitrary. The problem is to find a non-dependent type which approximates T. The obvious choice seems to be a recursive type which solves the equation $A = A \to A$ but such a calculus would not be strongly normalizing.

Our construction avoids the use of Kripke-structures and can be understood as a generalization of the concept of saturated sets to dependent types. Moreover it is straightforward to extend it to inductive types with large eliminations and allows to interpret types like T. We shall not treat this here but refer to [Alt93a], pp. 76.

The paper is organized as follows: We start by introducing a judgement presentation of CC and define some basic notations. The presentation of the model construction is divided in two parts: First we present Λ-sets and note that these do not give rise to a sound interpretation. Then we solve this problem by introducing saturated Λ-sets and show soundness. As a corollary we obtain strong normalization for the stripped terms. We then show how strong normalization for typed terms and decidability of equality can be derived by simple syntactic reasoning.

2 The judgement presentation of CC

CC is often presented in the equality-as-conversion style [CH88, Bar92], i.e. the equality is just the untyped β-conversion between preterms. When we are interested in a semantical analysis of the system it seems easier to use the equality-as-judgement presentation, as it is usual for Martin-Löf's Type Theory. The reason is that it is not clear how untyped conversion can be interpreted semantically. Not surprisingly this presentation is used in [Str91] who studies the categorical semantics of CC.

We will also follow [Str91] in that we use a very explicit notation: we differentiate between operations on Set (often called Prop) and types; we annotate applications and λ-abstractions with types and in one place we go even further and also annotate the codomain of a λ-abstraction. Essentially our terms are a linear notation for derivations where the applications of the conversion rule are omitted. The more implicit notation can be justified (e.g. see [Str91, Alt93a]), but semantically it seems to be more appropriate to consider the explicit presentation as the fundamental one.

We introduce precontexts Cn, pretypes Ty, preterms Tm and constructions Co [2] by the following grammar - the set of natural numbers ($i, j, k \in \omega$) is used for variables, since we use de-Bruijn-indices.

$$\mathrm{Cn}(\Gamma) \quad ::= \quad \bullet \mid \Gamma.\sigma$$

[2] In the following definition we introduce the sets together with a naming convention.

$$\begin{array}{rcl} \mathrm{Ty}(\sigma,\tau) & ::= & \Pi\sigma.\tau \mid \mathrm{Set} \mid \mathrm{El}(M) \\ \mathrm{Tm}(M,N) & ::= & i \mid \lambda\sigma(M)^{\tau} \mid \mathrm{app}^{\sigma.\tau}(M,N) \mid \forall\sigma.M \\ \mathrm{Co}(C,D) & ::= & \mathrm{Ty} \mid \mathrm{Tm} \end{array}$$

In our use of de-Bruijn-indices [3] we follow [Bar84], pp.577 with the minor difference that we start counting with 0. We denote substitution for the free variable with index i by $M[N]^i, \sigma[N]^i$ and all the variables with a greater index are decreased by one. We also require the operation of weakening M^{+i}, σ^{+i} which increases the indices of all free variables greater or equal i by one. If $i = 0$ we omit it. The precise definition of these operations can be found in [Alt93a],p. 24.

Given a sequence of terms $\vec{N} = N_{n-1}, N_{n-2} \ldots, N_0$ [4] we can define a notion of parallel substitution as a derived notion:

$$M[\vec{N}] = M[N_0\overbrace{+\cdots+}^{n-1 \text{ times}}][N_1\overbrace{+\cdots+}^{n-2 \text{ times}}]\ldots[N_{n-1}]$$

and analogously for $\sigma[\vec{N}]$. If the indices of all free variables in M are less than n then

$$M[n-1, n-2, \ldots, 0] = M.$$

We define the following judgements: $\vdash \Gamma$ (context validity), $\Gamma \vdash \sigma$ (type validity), $\Gamma \vdash M : \sigma$ (typing), $\Gamma \vdash \sigma \simeq \tau$ (type equality) and $\Gamma \vdash M \simeq N : \sigma$ (equality). The derivable judgements are given as the least relations closed under the following rules — we have omitted the obvious congruence rules to save space.

$$\frac{}{\vdash \bullet} \quad (\textsc{Empty})$$

$$\frac{\vdash \Gamma \qquad \Gamma \vdash \sigma}{\vdash \Gamma.\sigma} \quad (\textsc{Compr})$$

$$\frac{\Gamma.\sigma \vdash \tau}{\Gamma \vdash \Pi\sigma.\tau} \quad (\textsc{Pi})$$

$$\frac{\vdash \Gamma}{\Gamma \vdash \mathrm{Set}} \quad (\textsc{Set})$$

$$\frac{\Gamma \vdash A : \mathrm{Set}}{\Gamma \vdash \mathrm{El}(A)} \quad (\textsc{El})$$

$$\frac{\Gamma.\sigma \vdash A : \mathrm{Set}}{\Gamma \vdash \mathrm{El}(\forall\sigma.A) \simeq \Pi\sigma.\mathrm{El}(A)} \quad (\textsc{All-Elim})$$

[3] We believe that de-Bruijn-indices are the best way to make the notion of bound variables precise. We can often omit side conditions and reason about λ-terms in a purely algebraic fashion. Moreover, this notation reflects our semantic intuition that variables denote projections out of a context. However, when presenting syntax we may used named variables, meaning the obvious translation into a de-Bruijn-term.

[4] We write these sequences backwards since contexts are also written backwards.

$$\frac{\Gamma \vdash M : \sigma \qquad \Gamma \vdash \sigma \simeq \tau}{\Gamma \vdash M : \tau} \quad (\text{CONV})$$

$$\frac{\Gamma \vdash \sigma}{\Gamma.\sigma \vdash 0 : \sigma^+} \quad (\text{VAR-0})$$

$$\frac{\Gamma \vdash i : \sigma \qquad \Gamma \vdash \tau}{\Gamma.\tau \vdash i+1 : \sigma^+} \quad (\text{VAR-S})$$

$$\frac{\Gamma.\sigma \vdash M : \tau}{\Gamma \vdash \lambda\sigma(M)^\tau : \Pi\sigma.\tau} \quad (\text{LAM})$$

$$\frac{\Gamma \vdash M : \Pi\sigma.\tau \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash \mathrm{app}^{\sigma,\tau}(M,N) : \tau[N]} \quad (\text{APP})$$

$$\frac{\Gamma.\sigma \vdash A : \mathrm{Set}}{\Gamma \vdash \forall\sigma.A : \mathrm{Set}} \quad (\text{ALL})$$

$$\frac{\Gamma.\sigma \vdash M : \tau \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash \mathrm{app}^{\sigma,\tau}(\lambda\sigma(M)^\tau, N) \simeq M[N] : \tau[N]} \quad (\text{BETA-EQ})$$

We can easily establish a number of rather trivial properties of this presentation such that all judgements are consistent with weakening and substitution - see [Alt93a] for details.

3 Saturated Λ-sets and strong normalization

3.1 Λ-sets

In the following section we define an interpretation of CC which resembles the ω-set semantics. The main difference is that we use λ-terms instead of ω (i.e. indices of recursive functions). Another novelty is that we present this interpretation in elementary terms avoiding the use of categories - although the construction is clearly motivated by the categorical semantics of CC.

Definition 1 *Assuming some encoding of pairing (x,y) and projections π_1, π_2 we have the usual set-theoretic counterparts of the basic type-theoretic operations (assume A is a set and $\{B_a\}_{a\in A}$ a family of sets indexed by A):*

$$\begin{aligned} \Sigma a \in A.B_a &= \{(a,b) \mid a \in A, b \in B_a\} \\ \Pi a \in A.B_a &= \{f \subseteq \Sigma a \in A.B_a \mid \forall_{a\in A}\exists!_{b\in B_a}(a,b) \in f\} \end{aligned}$$

We consider application $f(x)$ as a partial operation which is defined if there is an $(x,y) \in f$ and then $f(x) = y$. We denote set-theoretic λ-abstraction by $\mapsto$, i.e.

$$x \in A \mapsto E[x] \equiv \{(x, E[x]) \mid x \in A\}.$$

Given a set X we denote the set of finite sequences over X by X^*. The empty sequence is denoted by ϵ and given a sequence $\vec{x} \in X^*$ and $y \in X$ we denote the extended sequence by $\vec{x}y \in X^*$.

Definition 2 *We use Λ to denote the set of untyped λ terms enriched by a special binder $\forall M$. To every preterm M we assign a stripping $|M| \in \Lambda$ by deleting all types.*

$\triangleright \subseteq \Lambda \times \Lambda$ is the usual one-step β-reduction extended by a ξ-rule for $\forall$. $\mathrm{SN} \subseteq \Lambda$ is the set of strongly normalizing (w.r.t. $\triangleright$) λ-terms.

We are ready to define Λ-sets which are used to interpret types and Λ^*-sets for interpreting contexts.

Definition 3 (Λ-sets)
A Λ-set X is a pair $(\overline{X}, \Vdash_X)$ with X is a set and $\Vdash_X \subseteq \Lambda \times \overline{X}$ s.t.

$$\forall_{x \in \overline{X}} \exists_{i \in \Lambda} i \Vdash_X x.$$

We denote the class of Λ-sets by $\mathfrak{L}$ and for any Λ-set $X \in \mathfrak{L}$ we use $\overline{X}$ and $\Vdash_X$ to denote its components.

$\mathfrak{L}^*$ is defined analogously by replacing Λ by Λ^*, i.e. sequences of λ-terms.

We introduce operations on Λ- and Λ^*-sets corresponding to the context and type forming operations. Additionally we define sections which given $\Gamma \vdash \sigma$ correspond to $\{M \mid \Gamma \vdash M : \sigma\}$ in the syntax.

Definition 4 *Assume $G \in \mathfrak{L}^*$, $\{Y_\gamma \in \mathfrak{L}\}_{\gamma \in \overline{G}}$, $X \in \mathfrak{L}$, $\{Z_x \in \mathfrak{L}\}_{x \in \overline{X}}$ and let:*

$$\begin{array}{rcl}
\mathbf{1}_\Lambda & = & (\{\epsilon\}, \Lambda \times \{\epsilon\}) \\
& \in & \mathfrak{L}^* \\
\Sigma_\Lambda(G, \{Y_\gamma\}_{\gamma \in \overline{G}}) & = & (\Sigma_{\gamma \in \overline{G}} \overline{Y_\gamma}, \{(\vec{M}N, (\gamma, y)) \mid \vec{M} \Vdash_G \gamma \wedge N \Vdash_{Y_\gamma} y\}) \\
& \in & \mathfrak{L}^* \\
\mathrm{Sect}(G, \{Y_\gamma\}_{\gamma \in \overline{G}}) & = & \{f \in \Pi_{\gamma \in \overline{G}} \overline{Y_\gamma} \mid \exists_{M \in \Lambda} M \Vdash_{\mathrm{Sect}(G, \{Y_\gamma\}_\gamma)} f\} \\
& \in & \mathfrak{L} \\
\multicolumn{3}{c}{\textit{where } \Vdash_{\mathrm{Sect}(G, \{Y_\gamma\}_\gamma)} = \{(M, f) \mid \forall_{\gamma \in \overline{G}} \forall_{\vec{N} \in \Lambda^*} \vec{N} \Vdash_G \gamma \to M[\vec{N}] \Vdash_{Y_\gamma} f(\gamma)\}} \\
\Pi_\Lambda(X, \{Z_x\}_{x \in \overline{X}}) & = & (\{f \in \Pi_{x \in \overline{X}} \overline{Z_x} \mid \exists_{M \in \Lambda} M \Vdash_{\Pi_\Lambda(X, \{Z_x\}_x)} f\}, \Vdash_{\Pi_\Lambda(X, \{Z_x\}_x)}) \\
& \in & \mathfrak{L} \\
\multicolumn{3}{c}{\textit{where } \Vdash_{\Pi_\Lambda(X, \{Z_x\}_x)} = \{(M, f) \mid \forall_{x \in \overline{X}} \forall_{N \in \Lambda} M \Vdash_X x \to MN \Vdash_{Z_x} f(x)\}}
\end{array}$$

Note that the only difference between Sect and Π_Λ is that the first one uses substitution and the second application. Indeed they are identified in the ω-set semantics.

We have not yet given an interpretation for Set and El, which is the main problem in finding an interpretation for CC. As in the usual ω-set semantics we will use the set of partial equivalence relations which is *equivalent* [5] to the subclass of modest Λ sets.

[5] The properties we show can be used to establish an equivalence of categories. We do not make this precise because we do not introduce PERs and Λ-sets as categories.

Definition 5 *We call $X \in \mathfrak{L}$ modest, iff*

$$\forall_{x,y \in \overline{X}} \forall_{M \in \Lambda} M \Vdash_X x \wedge M \Vdash_X y \to x = y,$$

We write $\mathfrak{M}$ for the subclass of modest Λ-sets.

A straightforward but important property of modest Λ-sets is that they are closed under Π_Λ:

Lemma 1 *Assume $X \in \mathfrak{L}$ and $\{Y_x \in \mathfrak{M}\}_x \in \overline{X}$ then*

$$\Pi_\Lambda(X, \{Y_x\}_x) \in \mathfrak{M}$$

Proof: Simple. ∎

We define the set of PERs together with translation operators to and from modest Λ-sets:

Definition 6

$$\mathrm{PER}(\Lambda) = \{R \subset \Lambda \times \Lambda \mid R \text{ is symmetric and transitive}\}$$

For any $R \in \mathrm{PER}(\Lambda)$ we define the set of equivalence classes $\Lambda/R \in \mathcal{P}(\Lambda)$ in the usual way.

Assume $R \in \mathrm{PER}(\Lambda)$ and $X \in \mathfrak{M}$:

$$\begin{array}{rcl} \mathrm{EL}(R) & = & (\Lambda/R, \in) \\ & \in & \mathfrak{M} \\ \mathrm{EL}^{-1}(X) & = & \{(M,N) \mid \exists_{x \in \overline{X}} M \Vdash_X x \ \wedge N \Vdash_X x\} \\ & \in & \mathrm{PER}(\Lambda) \end{array}$$

It is easy to see that we have $\mathrm{EL}^{-1}(\mathrm{EL}(R)) = R$ but the converse fails. Indeed the operation

$$\Theta(X) = \mathrm{EL}(\mathrm{EL}^{-1}(X))$$

assigns to any modest Λ-set X a canonical representation where $x \in \overline{X}$ is replaced by the set of its realizers. This is reflected by the fact that we have:

$$\begin{array}{rcl} \vartheta_X(x \in \overline{X}) & = & \{M \mid M \Vdash_X x\} \\ & \in & \Theta(X) \end{array}$$

with the following properties:

Lemma 2 *Let X be a modest Λ-set*

1. *ϑ_X is a bijection.*
2. *$M \Vdash_X x$ iff $M \Vdash_{\Theta(X)} \vartheta_X(x)$*

Proof: The preservation of realizers is quite easy to check and implies the first property since X is modest. ∎

We will use Θ to *normalize* modest sets and hence reflect type equality by equality of sets. To simplify notation we introduce $\tilde{\Theta}$ and $\tilde{\vartheta}$ as an extension of Θ and ϑ which are just identities on non-modest sets.

The following defines a partial interpretation of the syntax in terms of Λ-sets. We use $\cong$ for Kleene-equality and $\underset{\sim}{\in}$ to denote a partial version of $\in$: if both sides are defined then the relation $\in$ holds.

Definition 7
We define partial interpretation functions $[\![\vdash \Gamma]\!] \underset{\sim}{\in} \mathfrak{L}^*$, $\{[\![\Gamma \vdash \sigma]\!]\gamma \underset{\sim}{\in} \mathfrak{L}\}_{\gamma \underset{\sim}{\in} \overline{[\![\vdash \Gamma]\!]}}$ *and* $\{[\![\Gamma \vdash M]\!]\gamma\}_{\gamma \underset{\sim}{\in} \overline{[\![\vdash \Gamma]\!]}}$ *by induction over the structure of the syntax:*

$$
\begin{array}{rcl}
[\![\vdash \bullet]\!] & \cong & \mathbf{1}^{\Lambda} \\
[\![\vdash \Gamma.\sigma]\!] & \cong & \Sigma^{\Lambda}([\![\vdash \Gamma]\!], [\![\Gamma \vdash \sigma]\!]) \\
[\![\Gamma \vdash \Pi\sigma.\tau]\!]\gamma & \cong & \tilde{\Theta}(\Pi^{\Lambda}([\![\Gamma \vdash \sigma]\!]\gamma, \{[\![\Gamma.\sigma \vdash \tau]\!](\gamma, x)\}_x)) \\
[\![\Gamma \vdash \mathrm{Set}]\!]\gamma & \cong & (\mathrm{PER}(\Lambda), \Lambda \times \mathrm{PER}(\Lambda)) \\
[\![\Gamma \vdash \mathrm{El}(A)]\!]\gamma & \cong & \mathrm{EL}([\![\Gamma \vdash A]\!]\gamma) \\
[\![\Gamma \vdash i]\!]\gamma & \cong & \pi_2(\pi_1^i(\gamma)) \\
[\![\Gamma \vdash \lambda\sigma(M)^{\tau}]\!]\gamma & \cong & \tilde{\vartheta}_{[\![\Gamma \vdash \Pi\sigma.\tau]\!]\gamma}(x \in \overline{[\![\Gamma \vdash \sigma]\!]} \mapsto [\![\Gamma.\sigma \vdash M]\!](\gamma, x)) \\
[\![\Gamma \vdash \mathrm{app}^{\sigma.\tau}(M, N)]\!]\gamma & \cong & \tilde{\vartheta}^{-1}_{[\![\Gamma \vdash \Pi\sigma.\tau]\!]\gamma}([\![\Gamma \vdash M]\!]_{\gamma})([\![\Gamma \vdash N]\!]\gamma) \\
[\![\Gamma \vdash \forall^{\sigma}.A]\!]\gamma & \cong & \mathrm{EL}^{-1}(\Pi^{\Lambda}([\![\Gamma \vdash \sigma]\!]\gamma, \{\mathrm{EL}([\![\Gamma.\sigma \vdash A]\!](\gamma, x))\}_x))
\end{array}
$$

This interpretation is *not sound*, where by soundness we mean the following properties:

1. $$\frac{\vdash \Gamma}{[\![\vdash \Gamma]\!] \text{ is defined.}}$$
2. $$\frac{\Gamma \vdash \sigma \qquad \gamma \in \overline{[\![\vdash \Gamma]\!]}}{[\![\Gamma \vdash \sigma]\!]\gamma \text{ is defined.}}$$
3. $$\frac{\Gamma \vdash M : \sigma}{[\![\Gamma \vdash M]\!] \in \mathrm{Sect}([\![\vdash \Gamma]\!], [\![\Gamma \vdash \sigma]\!])}$$
4. $$\frac{\Gamma \vdash \sigma \simeq \tau \qquad \gamma \in \overline{[\![\vdash \Gamma]\!]}}{[\![\Gamma \vdash \sigma]\!]\gamma = [\![\Gamma \vdash \tau]\!]\gamma}$$
5. $$\frac{\Gamma \vdash M \simeq N : \sigma \qquad \gamma \in \overline{[\![\vdash \Gamma]\!]}}{[\![\Gamma \vdash M]\!]\gamma = [\![\Gamma \vdash N]\!]\gamma}$$

We will see in the next section how we can obtain soundness by a small modification. To motivate this it is instructive to see where soundness for the

interpretation above fails. Indeed, the above interpretation is not closed under (LAM).

For simplicity assume we have $\sigma \vdash M : \tau$ from which we can derive $\bullet \vdash \lambda\sigma(M)^\tau : \Pi\sigma.\tau$. Now as a hypothesis we assume

$$[\![\sigma \vdash M]\!] \in \mathrm{Sect}([\![\vdash \sigma]\!], [\![\sigma \vdash \tau]\!]).$$

From the definition of Sect it follows that there is an $M' \in \Lambda$ s.t. for all $N \Vdash_{[\![\vdash\sigma]\!]} x$ we have that $M'[N] \Vdash_{[\![\sigma\vdash\tau]\!]x} [\![\sigma \vdash M]\!]x$.

Can we conclude that

$$[\![\bullet \vdash \lambda\sigma(M)^\tau : \Pi\sigma.\tau]\!] \in \mathrm{Sect}([\![\vdash \bullet]\!], [\![\Pi\sigma.\tau]\!])?$$

By expanding the definition of the interpretation this goal can be reduced to showing:

$$[\![\sigma \vdash M]\!] \in \Pi^\Lambda([\![\vdash \sigma]\!], [\![\sigma \vdash \tau]\!])$$

I.e. we have to find a realizer M'' s.t. for any $N \Vdash_{[\![\vdash\sigma]\!]} x$ we have that

$$M''N \Vdash_{[\![\sigma\vdash\tau]\!]} [\![\sigma \vdash M]\!](x).$$

An obvious guess would be $M'' = \lambda M'$. However, since we have not identified β-equal terms we cannot reason that $M''N = M'[N]$ and indeed there is no reason to assume that an appropriate realizer exists at all.

This failure also suggests an obvious way to repair the problem: identify β-equal terms, i.e. use $\Lambda/=_\beta$ instead of Λ. Actually, it is not even necessary to identify all β-equal terms, it is sufficient to use weak β-equality, the equality generated by combinatory logic. This construction brings us very close to ω-sets or its generalization to arbitrary *Partially Combinatory Algebras* D-sets. [6]

However, we would hope to obtain a system which only contains strongly normalizing realizers and even weak β-equality is not closed under strong normalization. Hyland and Ong [HO93] propose to overcome this problem by using a generalization of PCAs (conditional PCAs) which can be used to define a partial congruence which identifies only strongly normalizing terms. Here we will go another way and generalize the notion of *saturated sets*, which are used in the strong normalization arguments of simply typed λ-calculus or System F.

3.2 Saturated Λ-sets

In this section we identify the subclass of saturated Λ-sets which has the following properties:

- All realizers are strongly normalizing.
- Π-types are closed under saturated Λ-sets.

[6] The D-set semantics differs only in two ways from the one proposed above: one uses a partial combinatory algebra which is a slight generalization of a combinatory algebra and the substitution machinery which we just imported from the untyped λ calculus is encoded by combinators.

- The set of realizers for a certain element are closed under certain β-expansions, s.t. (LAM) is sound.

By modifying the interpretation of Set we can obtain an interpretation which interprets every type by a saturated Λ-set. By establishing also that every interpretation of a term is realized by its stripping we obtain strong normalization as a simple corollary.

We introduce the notion of weak head-reduction, which means that only a head-redex not inside a λ-abstraction is reduced. This can be defined inductively by the following rules:

$$(\lambda M)N \rhd_{\mathrm{whd}} M[N] \qquad \frac{M \rhd_{\mathrm{whd}} M'}{MN \rhd_{\mathrm{whd}} M'N}$$

Certainly we have that $\rhd_{\mathrm{whd}} \subseteq \rhd$.

Void $\subseteq$ SN is the set of strongly normalizing weak-head normal forms which are not λ-abstractions. This set can be inductively defined as: [7]

1. $i \in \mathrm{Void}$.
2. $$\frac{M \in \mathrm{Void} \qquad N \in \mathrm{SN}}{MN \in \mathrm{Void}}$$
3. $$\frac{M \in \mathrm{SN}}{\forall M \in \mathrm{Void}}$$

We need the following properties of SN:

Lemma 3

1. $$\frac{M, N, M[N] \in \mathrm{SN}}{(\lambda M)N \in \mathrm{SN}}$$
2. $$\frac{M' \rhd_{\mathrm{whd}} M \qquad MN \in \mathrm{SN}}{M'N \in \mathrm{SN}}$$

Proof: See [Alt93a], pp.69. ∎

These properties can be shown by noetherian induction, i.e. induction over the longest reduction of a strongly normalizing term. For the second proposition it is useful to establish as a lemma that weak-head reductions can be always postponed.

It is interesting to note that these are precisely the same properties which are needed to show strong normalization in the simply typed case.

[7] Yet another alternative is to say that void terms have the form $iM_1 \ldots M_n$ with $M_i \in \mathrm{SN}$. However, our presentation has the advantage that it is easier to generalize to inductive types (see [Alt93a], p. 87).

Definition 8 *We call a Λ-set X saturated — $X \in \mathfrak{S}$ — iff the following conditions hold:*

SAT1 *Every realizer is strongly normalizing.*

$$\forall_{M \Vdash_X x} M \in \mathrm{SN}$$

SAT2 *There is a $\bot_X \in \overline{X}$ which is realized by every void term.*

SAT3 *The set of realizers for a certain element x is closed under weak head expansion inside* SN*:*

$$\forall_{M \Vdash_X x} \forall_{M' \in \mathrm{SN}} (M' \rhd_{\mathrm{whd}} M) \to (M' \Vdash_X x)$$

This can be extended to $\mathfrak{L}^*$-sets by the following inductive definition:

1. $\mathbf{1}_\Lambda \in \mathfrak{S}^*$.
2. $$\frac{G \in \mathfrak{S}^* \qquad \{X_\gamma \in \mathfrak{S}\}_{\gamma \in \overline{G}}}{\Sigma_\Lambda(G, \{X_\gamma\}_{\gamma \in \overline{G}}) \in \mathfrak{S}^*}$$

Note that for any saturated Λ-set $(\overline{X}, \Vdash_X)$ the set of realizers $\{M \mid \exists_{x \in \overline{X} M \Vdash_X x}\}$ is saturated in the conventional sense [8]

$\mathbf{1}_\Lambda$ and Σ_Λ restrict to operations on saturated Λ-sets by definition but it remains to show that this is also true for Π_Λ:

Lemma 4 *Assume $X \in \mathfrak{S}$, $\{Y_x \in \mathfrak{S}\}_{x \in \overline{X}}$ then $\Pi_\Lambda(X, \{Y_x\}_x) \in \mathfrak{S}$.*

Proof:

SAT1 Assume $M \Vdash_{\Pi_\Lambda(X, \{Y_x\}_x)} f$, certainly $0 \Vdash_X \bot_X$ (**SAT2** for X). Now we know that $M0 \Vdash_{Y_{\bot_X}} f(\bot_X)$, therefore $M0 \in \mathrm{SN}$ (**SAT1** for Y_x), which implies $M \in \mathrm{SN}$.

SAT2 Assume $M \in \mathrm{Void}$, now for every $N \Vdash_X x$ we have that $MN \in \mathrm{Void}$ (**SAT1** for X and definition of Void) and therefore $MN \Vdash_{Y_x} \bot_{Y_x}$. This implies $M \Vdash_{\Pi_\Lambda(X, \{Y_x\}_x)} x \mapsto \bot_{Y_x}$, so we just set $\bot_{\Pi_\Lambda(X, \{Y_x\}_x)} = x \mapsto \bot_{Y_x}$.

SAT3 Assume $M \Vdash_{\Pi_\Lambda(X, \{Y_x\}_x)} f$, $M' \in \mathrm{SN}$ and $M' \rhd_{\mathrm{whd}} M$. For any $N \Vdash_X x$ we have that $MN \Vdash_{Y_x} f(x)$. By (App-L) $M'N \rhd_{\mathrm{whd}} MN$ and by lemma 3 (2.) $M'N \in \mathrm{SN}$. Using **SAT3** for Y_x we have that $M'N \Vdash_{Y_x} f(x)$. Therefore we have established that $M' \Vdash_{\Pi_\Lambda(X, \{Y_x\}_x)} f$.

∎

The essential idea of saturated Λ-sets is that we can prove closure under the λ-introduction rule.

[8] E.g. see [Bar92].

Lemma 5 *Let* $G \in \mathfrak{S}^*, \{X_\gamma \in \mathfrak{S}\}_{\gamma\in\overline{G}}, \{Z_\delta \in \mathfrak{S}\}_{\delta\in\overline{\Sigma_\Lambda(G,\{X_\gamma\}_\gamma)}}$ *then*

$$\frac{M \Vdash_{\mathrm{Sect}(\Sigma_\Lambda(G,\{X_\gamma\}_\gamma),\{Z_\delta\}_\delta)} f}{\lambda M \Vdash_{\mathrm{Sect}(G,\{\Pi_\Lambda(X_\gamma,\{Z_{(\gamma,x)}\}_x)\}_\gamma)} \gamma \in G \mapsto (x \in \overline{X_\gamma} \mapsto f(\gamma,x))}$$

Proof: Assume any $\gamma \in \overline{G}, \vec{N} \Vdash_G \gamma, x \in X_\gamma, N \Vdash_{X_\gamma} x$. We would like to show that

$$(\lambda M)[\vec{N}]N \Vdash_{Z_{(\gamma,x)}} f(\gamma,x).$$

Now $(\lambda M)[\vec{N}]N = (\lambda M[\vec{N}0])N \rhd_{\mathrm{whd}} M[\vec{N}N]$ and

$$M[\vec{N}N] \Vdash_{Z_{(\gamma,x)}} f(\gamma,x)$$

follows from the premise.

To apply (SAT3) we have to verify that $N, M[\vec{N}N], M[\vec{N}0] \in \mathrm{SN}$. The first two are immediate by (SAT1) and for the last one we need that $0 \Vdash_{X_\gamma} \bot$ (SAT2) and by premise

$$M[\vec{N}0] \Vdash_{Z_{\gamma,\bot}} f(\gamma,\bot)$$

and therefore $M[\vec{N}0] \in \mathrm{SN}$ (SAT1). ∎

We will now modify the interpretation simply by changing the interpretation of Set.

Definition 9 *We define a new interpretation* $[\![\vdash \Gamma]\!]', \{[\![\Gamma \vdash \sigma]\!]'\gamma\}_{\gamma \lesssim \overline{[\![\vdash\Gamma]\!]'}}$, $\{[\![\Gamma \vdash M]\!]'\gamma\}_{\gamma \lesssim \overline{[\![\vdash\Gamma]\!]'}}$ *by the same rules as before but modifying* $[\![\Gamma \vdash \mathrm{Set}]\!]$*:*

$$[\![\Gamma \vdash \mathrm{Set}]\!]\gamma \quad \cong \quad (\mathrm{PER}'(\Lambda), \mathrm{SN} \times \mathrm{PER}'(\Lambda))$$

where

$$\mathrm{PER}'(\Lambda) = \{R \in \mathrm{PER}(\Lambda) \mid \mathrm{EL}(R) \in \mathfrak{S}\}.$$

Before we can prove the general soundness theorem, we need a technical result, i.e. that weakening and substitution are interpreted properly.

Lemma 6 (Soundness of weakening and substitution) *For any* $\gamma \in \overline{[\![\vdash \Gamma]\!]'}$ *and* $x \in \overline{[\![\Gamma \vdash \tau]\!]'\gamma}$ *we have*

$$\begin{aligned}
[\![\Gamma \vdash \sigma]\!]'\gamma &\cong [\![\Gamma.\tau \vdash \sigma^+]\!]'\gamma x \\
[\![\Gamma \vdash M]\!]'\gamma &\cong [\![\Gamma.\tau \vdash M^+]\!]'\gamma x \\
[\![\Gamma.\tau \vdash \sigma]\!]'\gamma([\![\Gamma \vdash N]\!]'\gamma) &\cong [\![\Gamma \vdash \sigma[N]]\!]'\gamma \\
[\![\Gamma.\tau \vdash M]\!]'\gamma([\![\Gamma \vdash N]\!]'\gamma) &\cong [\![\Gamma \vdash M[N]]\!]'\gamma
\end{aligned}$$

Proof: See [Alt93a], section 3.2. ∎

It should be noted that only a generalization of the proposition to arbitrary weakenings and substitutions can be shown by induction over the syntax.

Theorem 1 (Soundness)

1. $$\frac{\vdash \Gamma}{[\![\vdash \Gamma]\!]' \in \mathfrak{S}^*}$$

2. $$\frac{\Gamma \vdash \sigma \qquad \gamma \in \overline{[\![\vdash \Gamma]\!]'}}{[\![\Gamma \vdash \sigma]\!]'\gamma \in \mathfrak{S}}$$

3. *(a)* $$\frac{\Gamma \vdash M : \sigma}{[\![\Gamma \vdash M]\!]' \in \mathrm{Sect}([\![\vdash \Gamma]\!]', [\![\Gamma \vdash \sigma]\!]')}$$

   (b) $$\frac{\Gamma \vdash M : \sigma}{|M| \Vdash_{\mathrm{Sect}([\![\Gamma]\!]', [\![\Gamma \vdash \sigma]\!]')} [\![\Gamma \vdash M]\!]'}$$

4. $$\frac{\Gamma \vdash \sigma \simeq \tau \qquad \gamma \in \overline{[\![\vdash \Gamma]\!]'}}{[\![\Gamma \vdash \sigma]\!]'\gamma = [\![\Gamma \vdash \tau]\!]'\gamma}$$

5. $$\frac{\Gamma \vdash M \simeq N : \sigma \qquad \gamma \in \overline{[\![\vdash \Gamma]\!]'}}{[\![\Gamma \vdash M]\!]'\gamma = [\![\Gamma \vdash N]\!]'\gamma}$$

Proof: (Sketch) The result can be obtained by a straightforward induction over the structure of derivations. All the congruence rules and (CONV) follow directly from the fact that we interpret syntactic equality by semantic (i.e. set-theoretic) equality.

1. Immediate from the definition of $\mathfrak{S}^*$.

2. For (PI) we need Lemma 4 and observe that Θ preserves saturatedness. (EL) follows from the definition of $\mathrm{PER}'(\Lambda)$ and (SET) is straightforward as well.

3. (VAR-0),(VAR-S) require soundness of weakening, (APP) is straightforward but uses soundness of substitution. (LAM) follows directly from Lemma 5.

4. The only interesting case is (ALL-ELIM):

$$\begin{aligned}
&[\![\Gamma \vdash \mathrm{El}(\forall\sigma.A)]\!]'\gamma \\
&\quad = \quad \mathrm{EL}(\mathrm{EL}^{-1}(\Pi_\Lambda([\![\Gamma \vdash \sigma]\!]'\gamma, \{\mathrm{EL}([\![\Gamma.\sigma \vdash A]\!]'(\gamma, x))\}_x))) \\
&\quad = \quad \Theta(\Pi_\Lambda([\![\Gamma \vdash \sigma]\!]'\gamma, \{\mathrm{EL}([\![\Gamma.\sigma \vdash A]\!]'(\gamma, x))\}_x)) \\
&\quad = \quad [\![\Pi_\Lambda\sigma.\mathrm{El}(A)]\!]'\gamma
\end{aligned}$$

   Note that we implicitly use Lemma 1

5. (BETA-EQ) requires soundness of substitution.

∎

The theorem has strong normalization as a corollary:

Corollary 1 (Strong normalization) *If* $\Gamma \vdash M : \sigma$ *then* $|M| \in \mathrm{SN}$.

Proof: Let n by the length of Γ.Using (SAT2) we know that

$$n-1, n-2, \ldots 0 \Vdash_{[\![\Gamma]\!]'} \bot, \bot \ldots \bot = \vec{\bot}$$

by Theorem 1, (3b) we know

$$|M| = |M|[n-1, n-2, \ldots 0] \Vdash_{[\![\Gamma \vdash \sigma]\!]' \vec{\bot}} [\![\Gamma \vdash M]\!]' \vec{\bot}$$

and therefore $M \in \mathrm{SN}$ by SAT1. ∎

4 Decidability

We have only established strong normalization for the stripped terms. It is not immediate that this implies strong normalization for typed terms and decidability of equality. The main problem with typed terms is that we have to allow reductions inside the type annotations to reflect the congruence rules.

It would be possible to redo the model construction using typed terms instead. However, it seems that the presentation of the interpretation would get quite overloaded with a lot of trivial syntactic reasoning. Here we go another way and show how this result can be derived from strong normalization for the stripped terms by a simple syntactic argument.

In the following text we assume a notion of reduction on types and terms $\rhd_l \subseteq \mathrm{Cn} \times \mathrm{Cn}$ which is just the natural extension of untyped β reduction to constructions. We also use SN_l to denote the set of strongly normalizing constructions wrt. $\rhd_l$. The l stands for *loose* in contrast it to tight reduction $\rhd_t$ where only redexes with agreeing types can be reduced (see below).

We define a type-preserving map blow which blows up terms such that every reduction in a typed term can be mirrored by a reduction in a stripped term:

Definition 10 *Let*

$$\begin{aligned} \bot &= \forall x : \mathrm{Set}.x \\ M(\sigma, N) &= \mathrm{app}^{x:\mathrm{Set}.\sigma^+}(\lambda x : \mathrm{Set}(M^{+x})^{\sigma^{+x}}, N) \end{aligned}$$

We now define $\mathrm{blow} \in \mathrm{Cn} \to \mathrm{Cn}$*:*

$$\begin{aligned} \mathrm{blow}(\Pi\sigma.\tau) &= \mathrm{blow}(\sigma)(\mathrm{Set}, \mathrm{blow}(\tau)) \\ \mathrm{blow}(\mathrm{Set}) &= \bot \\ \mathrm{blow}(\mathrm{El}(A)) &= \mathrm{blow}(A) \\ \mathrm{blow}(i) &= i \\ \mathrm{blow}(\mathrm{app}^{\sigma.\tau}(M, N)) &= \mathrm{app}^{\sigma.\tau}(\mathrm{blow}(M), \mathrm{blow}(N))(\tau[N], \mathrm{blow}(\sigma))(\tau[N], \mathrm{blow}(\tau)) \\ \mathrm{blow}(\lambda\sigma(M)^\tau) &= \lambda\sigma(\mathrm{blow}(M))^\tau(\Pi\sigma.\tau, \mathrm{blow}(\sigma))(\Pi\sigma.\tau, \mathrm{blow}(\tau)) \\ \mathrm{blow}(\forall\sigma.A) &= \forall\sigma.\mathrm{blow}(A)(\mathrm{Set}, \mathrm{blow}(\sigma)) \end{aligned}$$

We have the following properties:

Lemma 7

1. $\dfrac{\Gamma \vdash \sigma}{\Gamma \vdash \mathrm{blow}(\sigma) : \mathrm{Set}}$
2. $\dfrac{\Gamma \vdash M : \sigma}{\Gamma \vdash \mathrm{blow}(M) : \sigma}$
3. *If* $C \rhd_{\mathrm{l}} D$ *then* $|\mathrm{blow}(C)| \rhd_{\mathrm{l}}^{+} |\mathrm{blow}(D)|$.

From this it should be obvious how to derive the following (using Corollary 1):

Lemma 8

1. $\dfrac{\Gamma \vdash \sigma}{\sigma \in \mathrm{SN}_l}$
2. $\dfrac{\Gamma \vdash M : \sigma}{M \in \mathrm{SN}_l}$

In the conversion presentation the previous result would suffice to establish decidability because conversion is just defined as the transitive symmetric closure of $\rhd_{\mathrm{l}}$. In our presentation the reasoning is a bit more intricate, because we would have to establish a *subject reduction property*, which is a non-trivial property of the system.

To avoid this we define another notion of reduction — tight reduction:

$$\mathrm{app}^{\sigma.\tau}(\lambda\sigma(M)^{\tau}, N) \rhd_{\mathrm{t}} M[N] \qquad \text{(BETA-RED)}$$

For $\rhd_{\mathrm{t}}$ the subject reduction property can be easily established. We can also show the weak Church Rosser property and it is easy to see that $\rhd_{\mathrm{t}}$ is strongly normalizing for derivable terms and types because $\rhd_{\mathrm{t}} \subseteq \rhd_{\mathrm{l}}$.

5 Discussion

It should be noted that our strong normalization argument (i.e. Corollary 1) can be extended to η-reduction without any problems — this relies on the fact that Lemma 3 also holds for η-reduction. Alas, this does not entail decidability for CC $\beta\eta$-equality — this is the CC extended by the rule:

$$\frac{\Gamma \vdash M : \Pi\sigma.\tau}{\Gamma \vdash \lambda\sigma(\mathrm{app}^{\sigma^{+}.\tau^{+1}}(M^{+}, 0))^{\tau} \simeq M : \Pi\sigma.\tau} \qquad \text{(ETA-EQ)}$$

The problem is that we need *strengthening*:

$$\frac{\Gamma.\sigma \vdash M^{+} : \tau^{+}}{\Gamma \vdash M : \tau}$$

to derive subject reduction for tight reduction. However, it is not clear to me how to prove strengthening (I conjecture that this is not derivable by simple syntactic reasoning). [9]

The essential problem in extending our strong normalization argument to a system with inductive types which allows the definition of Sets by recursion is to extend the usual realizability interpretation since the extension to saturated Λ-sets follows the same lines. This corresponds to showing that initial T-algebras exist in D-set for a general class of functors on modest sets. Although this proposition seems to be folklore we could not find a satisfying presentation. In [Alt93a] we show how the D-set and the saturated Λ-set semantics can be extended to a non-algebraic inductive type with large eliminations. We claim that the same argument works for a general class of inductive definitions.

Acknowledgements

I would like to thank Stefano Berardi, Rod Burstall, Thierry Coquand, Peter Dybjer, Herman Geuvers, Healfdene Goguen, Martin Hofmann, Zhaohui Luo, Eike Ritter, Thomas Streicher and Benjamin Werner for interesting discussions related to the subject. I learnt a lot about ω-sets from Wesley Phoa's lectures [Pho92] and about the D-set semantics of CC from Thomas Streicher's book [Str91]. I would also like to thank the referees for their helpful and detailed comments on the preliminary version of the paper.

References

[Alt93a] Thorsten Altenkirch. *Constructions, Inductive Types and Strong Normalization.* PhD thesis, University of Edinburgh, November 1993.

[Alt93b] Thorsten Altenkirch. Yet another strong normalization proof for the Calculus of Constructions. In *Proceedings of El Vintermöte*, number 73 in Programming Methodology Group Reports. Chalmers University, Göteborg, 1993.

[Bar84] H.P. Barendregt. *The Lambda Calculus - Its Syntax and Semantics (Revised Edition).* Studies in Logic and the Foundations of Mathematics. North Holland, 1984.

[Bar92] H.P. Barendregt. Lambda calculi with types. In *Handbook of Logic in Computer Science, Vol. 2*, pages 118 – 310. Oxford University Press, 1992.

[9] In Nijmegen I proposed to use a modified η-rule instead:

$$\frac{\Gamma.\sigma \vdash M : \Pi\sigma^+.\tau^{+1}}{\Gamma \vdash \lambda\sigma(\mathrm{app}^{\sigma^+.\tau^{+1}}(M^+,0))^\tau \simeq M : \Pi\sigma.\tau} \quad (\text{ETA-EQ'})$$

For this rule subject reduction is derivable. However, as Thomas Streicher showed me, this rule is highly problematic, since it forbids models with empty types.

[CG90] Thierry Coquand and Jean Gallier. A proof of strong normalization for the theory of constructions using a Kripke-like interpretation. Informal Proceedings of the First Annual Workshop on Logical Frameworks, Antibes, 1990.

[CH88] Thierry Coquand and Gerard Huet. The calculus of constructions. *Information and Computation*, 76:95 – 120, 1988.

[Coq85] Thierry Coquand. *Une théorie des constructions.* PhD thesis, Université Paris VII, 1985.

[Ehr89] Thomas Ehrhard. Dictoses. In D.H. Pitt et al., editors, *Category Theory and Computer Science*, pages 213–223. Springer, 1989. LNCS 389.

[Geu93] Herman Geuvers. *Logics and Type Systems.* PhD thesis, Katholieke Universiteit Nijmegen, 1993.

[HO93] J.M.E. Hyland and C.-H. L. Ong. Modified realizability toposes and strong normalization proofs. In J.F. Groote M. Bezem, editor, *Typed Lambda Calculi and Applications*, LNCS 664, 1993.

[Pho92] Wesley Phoa. An introduction to fibrations, topos theory, the effective topos and modest sets. LFCS report ECS-LFCS-92-208, University of Edinburgh, 1992.

[Str89] Thomas Streicher. *Correctness and Completeness of a Categorical Semantics of the Calculus of Constructions.* PhD thesis, Universität Passau, Passau, West Germany, June 1989.

[Str91] Thomas Streicher. *Semantics of Type Theory.* Birkhäuser, 1991.

[Wer92] Benjamin Werner. A normalization proof for an impredicative type system with large eliminations over integers. In *Workshop on Logical Frameworks.* BRA Types, 1992. Preliminary Proceedings.