A short and flexible proof of Strong Normalization for the Calculus of Constructions

Herman Geuvers[*]

Faculty of Mathematics and Computer Science,
Eindhoven University of Technology
The Netherlands

1. Introduction

In the literature there are several different proofs of Strong Normalization (SN) for the Calculus of Constructions (CC). Some of them are of purely syntactical nature (like the ones in [Coquand 1985], [Geuvers and Nederhof 1991] and in [Coquand and Gallier 1990]), while others give a proof of normalization by describing an appropriate semantics (like [Ong and Ritter 1994] and [Altenkirch 1993], who describe an denotational semantics, but also [Goguen 1994], who describes a typed operational semantics). Apart from these, proofs of SN for CC can be found in [Berardi 1988], [Luo 1990] (containing a proof of SN for the 'Extended' Calculus of Constructions), [Terlouw 1993] and [Geuvers 1993] (containing a proof of SN for CC with β and η reduction). Each of these proofs exploits the idea of interpreting types as specific sets of strongly normalizing λ-terms. Then the terms are interpreted in such a way that, (1) if t is of type σ, then the interpretation of t is in the set associated with σ, and (2) for any term t, if its interpretation is SN, then t itself is SN.

For systems without type dependency (like the polymorphic λ calculus), it is rather well-known by now how to give a proof of SN using so called 'saurated sets' as interpretations for the types. These saturated sets are sets of untyped λ terms that satisfy some specific closure conditions and that are rather easy to work with. A possible drawback of this approach is that the interpretation of the typed term t should be an untyped term, and hence the interpretation will remove all type information from the term t (and hence it may remove some redexes). For the polymorphic λ calculus, this is not a real problem, because the reduction that comes from type-abstractions and type-applications can not be the source of an infinite reduction. In a system with type dependency, the situation is rather more complicated, because types can contain terms as subexpressions. (So, if one removes all types, then one also removes some terms.) In the Calculus of Constructions the situation is furthermore complicated by the fact that the system is higher order, which means that there are reductions in type-constructors.

One possible approach to coping with type dependency is to look at sets of typed terms instead of untyped terms. This is done, for example, in [Berardi 1988] and [Coquand and Gallier 1990]. Another possibility is to reduce the question

[*] e-mail: herman@win.tue.nl

of SN for a system with type dependency to SN for a system without type dependency. This is done in [Geuvers and Nederhof 1991]. Both approaches lead to rather involved proofs that consist of putting several steps together. Furthermore, these proofs do not easily scale up to extensions of CC with other type constructors.

The approach that we use here is based on saturated sets. It yields a (relatively short) direct proof of SN for CC using two different interpretations, $[\![-]\!]_\xi$ and $(\![-]\!)_\rho$. The first gives a set or a set-theoretic function for every type, constructor, kind or universe of CC. This is done modulo a valuation function ξ, which assigns a set or set-theoretic function to the constructor variables. (For those not familiar with CC, this terminology is explained below.) The second gives an untyped term for every object, type, constructor or kind of CC. This is done modulo a valuation function ρ, which assigns an untyped term to the constructor variables and the object variables. SN for CC then follows from the fact that

(1) if ξ and ρ are valuations that 'agree with' the context Γ and $\Gamma \vdash M : T$, then $(\![M]\!)_\rho \in [\![T]\!]_\xi$

(2) one can choose these valuations ξ and ρ in such a way that $(\![M]\!)_\rho$ is SN if and only if M is SN.

In 3.1 we give some more technical intuition for the proof.

One nice aspect of this approach is that the proof of SN for CC is carried out in exactly the same structure as where the proof of SN for Fω is usually done. This again emphasises that the proof of SN for CC is of the same proof-theoretic complexity as the proof of SN for Fω. (This has already been shown in [Berardi 1988] and [Geuvers and Nederhof 1991].) Furthermore, the proof uses only a minimal part of the meta-theory of CC. This makes it possible to extend the proof of SN for CC to larger systems (with more type constructors). In Section 4 we show this by proving SN for CC with W-types. In Section 5 we treat the extension with Σ-types and inductive kinds (where the inductive 'types' are of type $\Box$; these are also called *large* inductive types). For each of these extensions, the proof of SN is a natural generalization of the proof of SN for CC.

Of course there is a limitation to this: some meta-theory is still required and the approach we have chosen here requires that we can always define a kind of 'proof-irrelevant' interpretation (which interprets the types Pt and Pq as the same saturated set, independent of the objects t and q). This implies that the proof does not scale up to the extension with *small* inductive types (where the inductive type is of type $\star$), because there one can form a type constructor P such that $P0$ is convertible with $\Pi\alpha{:}\star.\alpha$ and $P1$ is convertible with $\Pi\alpha{:}\star.\alpha\to\alpha$. We discuss the restrictions of the method in more detail in the conclusions.

2. The Calculus of Constructions

We now give a precise definition of the Calculus of Constructions and at the same time we fix some terminology. In CC there are two specific constants, $\star$

and $\Box$. The first represents the universe of *types* (so we shall say that σ *is a type* if $\sigma : \star$) and the second represents the universe of *kinds* (so we shall say that A *is a kind* if $A : \Box$). The universe $\star$ is a specific example of a kind, so it will be the case that $\star : \Box$. To present the derivation rules for CC we first fix the set of *pseudoterms* from which the derivation rules select the (typable) terms.

2.1. DEFINITION. The set of pseudoterms, T, is defined by

$$\mathsf{T} ::= \star \mid \Box \mid \mathsf{Var} \mid (\Pi\mathsf{Var}{:}\mathsf{T}.\mathsf{T}) \mid (\lambda\mathsf{Var}{:}\mathsf{T}.\mathsf{T}) \mid \mathsf{TT},$$

where Var is a countable set of expressions, called variables. Both Π and λ bind variables and we have the usual notions of *free variable* and *bound variable.* The substitution of N for v in M is denoted by $M[N/v]$. On T we have the usual notion of β-reduction, denoted by $\longrightarrow_\beta$. We also adopt from the untyped λ calculus the conventions of denoting the transitive reflexive closure of $\longrightarrow_\beta$ by $\twoheadrightarrow_\beta$ and the transitive symmetric closure of $\twoheadrightarrow_\beta$ by $=_\beta$.

The typing of terms is done under the assumption of specific types for the free variables that occur in the term. These are listed in a *context*, which is a sequence of declarations $v_1{:}T_1, \ldots, v_n{:}T_n$, where the v_i are distinct variables and the T_i are pseudoterms. Contexts are denoted by the symbol Γ. For Γ a context and v a variable, v is said to be *Γ-fresh* if it is not among the variables that are declared in Γ.

2.2. DEFINITION. The Calculus of Constructions (CC) is the typed λ calculus with the following deduction rules.

$$(\text{ax}) \qquad \vdash \star : \Box$$

$$(\text{var}) \qquad \frac{\Gamma \vdash T : \star/\Box}{\Gamma, v{:}T \vdash v : T} \qquad \text{if } v \text{ is } \Gamma\text{-fresh}$$

$$(\text{weak}) \qquad \frac{\Gamma \vdash T : \star/\Box \quad \Gamma \vdash M : U}{\Gamma, v{:}T \vdash M : U} \qquad \text{if } v \text{ is } \Gamma\text{-fresh}$$

$$(\Pi) \qquad \frac{\Gamma \vdash T : \star/\Box \quad \Gamma, v{:}T \vdash U : s}{\Gamma \vdash \Pi v{:}T.U : s} \qquad \text{if } s \in \{\star, \Box\}$$

$$(\lambda) \; \frac{\Gamma, v{:}T \vdash M : U \quad \Gamma \vdash \Pi x{:}T.U : \star/\Box}{\Gamma \vdash \lambda v{:}T.M : \Pi v{:}T.U}$$

$$(\text{app}) \qquad \frac{\Gamma \vdash M : \Pi v{:}T.U \quad \Gamma \vdash N : T}{\Gamma \vdash MN : U[N/x]}$$

$$(\text{conv}) \qquad \frac{\Gamma \vdash M : T \quad \Gamma \vdash U : \star/\Box}{\Gamma \vdash M : U} \qquad T = U$$

The equality in the side condition to the conversion rule (conv) is the β-equality on the set of pseudoterms T.
The set of terms of CC is defined by $\mathsf{Term} = \{A \mid \exists\Gamma, B[\Gamma \vdash A : B \vee \Gamma \vdash B : A]\}$.

2.1. Required meta-theory

The set of terms of CC is devided into layers, because, if $M \in \mathsf{Term}$, then one of the following four situations occurs:

(1) $M \equiv \Box$
(2) $\Gamma \vdash M : \Box$
(3) $\Gamma \vdash M : T$ with $\Gamma \vdash T : \Box$
(4) $\Gamma \vdash M : T$ with $\Gamma \vdash T : \star$

Note that $M \equiv \star$ is a special case of (2) and $\Gamma \vdash M : \star$ is a special case of (3). It is well-known that these cases are disjoint if we are slightly more careful with the presentation of the syntax. Hence the following definition is useful.

2.3. DEFINITION. 1. The set of *kinds* is defined by $\mathsf{Kind} := \{A \mid \exists\Gamma[\Gamma \vdash A : \Box]\}$.
2. The set of *types* is defined by $\mathsf{Type} := \{A \mid \exists\Gamma[\Gamma \vdash A : \star]\}$.
3. The set of *constructors* is defined by $\mathsf{Constr} := \{P \mid \exists A, \Gamma[\Gamma \vdash P : A : \Box]\}$.
4. The set of *objects* is defined by $\mathsf{Obj} := \{P \mid \exists A, \Gamma[\Gamma \vdash P : A : \star]\}$.
Here $\Gamma \vdash P : A : \star$ denotes the fact that $\Gamma \vdash P : A$ and $\Gamma \vdash A : \star$.

2.4. CONVENTION. We devide the set of variables Var in two disjoint sets $\mathsf{Var}^{\star}$ and $\mathsf{Var}^{\Box}$. Elements from $\mathsf{Var}^{\star}$ are called *object variables*; we use x, y and z to denote object variables. Elements from $\mathsf{Var}^{\Box}$ are called *constructor variables*; we use α, β and γ to denote constructor variables.
In the (var) and (weak) rules we now make the restriction that, if $\Gamma \vdash T : \star$, then the new variable has to be taken from the set $\mathsf{Var}^{\star}$ and if $\Gamma \vdash T : \Box$, then the new variable has to be taken from the set $\mathsf{Var}^{\Box}$.

The usefulness of this definition is due to the following lemma. (For a detailed proof see [Geuvers 1993].)

2.5. LEMMA (Classification). *In CC,* $\mathsf{Kind} \cap \mathsf{Type} = \emptyset$ *and* $\mathsf{Constr} \cap \mathsf{Obj} = \emptyset$.

The Lemma implies that, when we define a mapping on terms of CC by induction on the structure, we can always distinguish cases according to whether a specific subterm is a kind or type, respectively a constructor or object, without making reference to a specific context.

For the extensions of CC that are considered in later sections, this property also holds. The usual proof of the Classification Lemma uses the Church-Rosser property, Subject Reduction and Uniqueness of Types. However, for CC and the extensions of CC considered here, a direct proof can be given. (This can be done along the lines of [Barbanera et al. 1995], where a proof of the Classification Lemma is given for the extension of CC with higher order algebtraic rewriting.) Note however that, even if there is no Classification Lemma, the definitions in this paper can still go through (with slightly more technical effort) in case one can distinguish cases according to whether a specific subterm is a type or kind *in a fixed context.* The other property of type systems that is really actually required for the constructions in this paper to go through is a slight strengthening of the

Stripping property (also called *Generation*). This property says, for example, that if $\Gamma \vdash \lambda v{:}T.M : U$ has a derivation D, then one can find a subderivation of D with conclusion $\Gamma', v{:}T \vdash M : T'$, where $\Pi v{:}T.T'$ is convertible with U and Γ' is a begin-part of Γ. (There are similar cases for terms of the form MN and $\Pi v{:}U.T$.) What we need in this paper is that the T' is not just such that $\Pi v{:}T.T'$ is convertible with U, but also that there is a path of reductions and expansions from $\Pi v{:}T.T'$ to U that *remains inside the set of well-typed terms.* (Remember that the side condition in the conversion rule says that U and T should be equal *as pseudoterms.*) This strengthening of Stripping holds straightforwardly for CC, because there we only consider β-conversion, which happens to be Church-Rosser on the pseudoterms and one has the Subject Reduction property.

3. Strong Normalization for the Calculus of Constructions

3.1. Intuition for the proof

Before giving the technical details we want to give some (technical) intuition for the proof. In order to do that we first look at the situation for Fω. In that case one defines mappings $\mathcal{V}$: $\mathsf{Kind}{\to}$Set, $[\![-]\!]_\xi$: $\mathsf{Constr}{\to}$Set, and $(\![-]\!)_\rho$: $\mathsf{Obj}{\to}\Lambda$. Here, ξ is a valuation of constructor-variables and ρ is a valuation of object-variables. These mappings are such that, if ξ, ρ form a *valuation of* Γ (this notion will be defined in detail later), then

$$\Gamma \vdash P : A(:\Box) \Longrightarrow [\![P]\!]_\xi \in \mathcal{V}(A),$$
$$\Gamma \vdash t : \sigma(:\star) \Longrightarrow (\![t]\!)_\rho \in [\![\sigma]\!]_\xi.$$

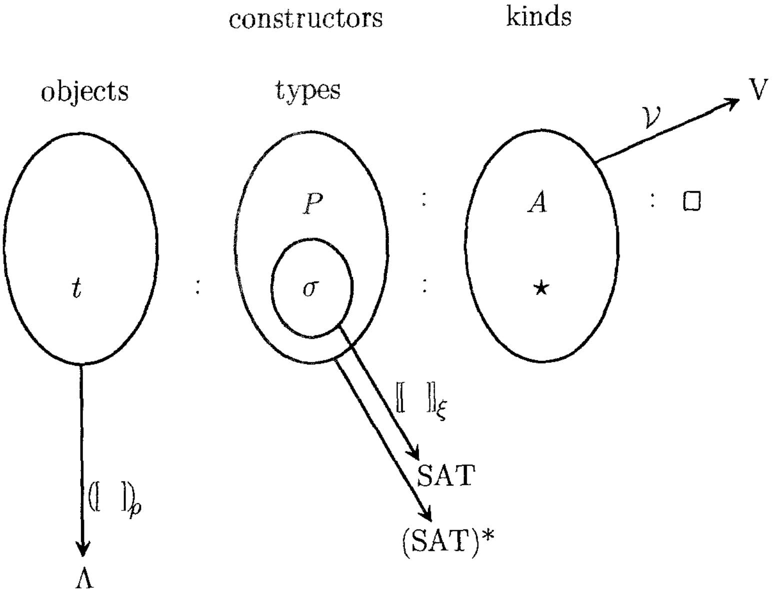
Furthermore, ρ can always be chosen in such a way that

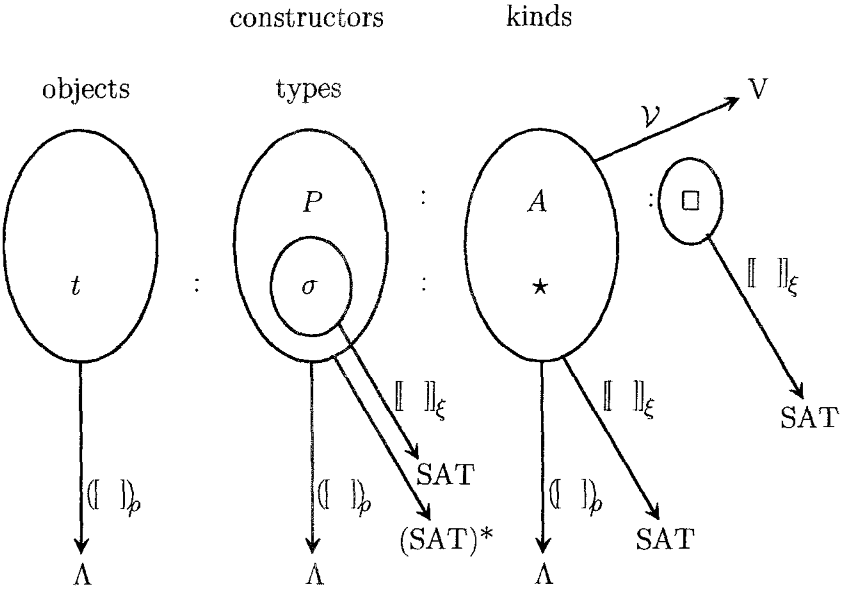
$$(\![M]\!)_\rho \text{ is SN iff } M \text{ is SN.}$$

In fact, $(\![-]\!)_\rho$ will in almost all cases be the extension of the valuation ρ to a substitution. (So, $(\![M]\!)_\rho$ is the term obtained by substituting $\rho(v)$ for v in M for all variables v.)

The situation is represented in the first picture on the next page. Here SAT denotes the set of *saturated sets* and (SAT)* denotes the union of the function spaces built from SAT, so (SAT)* $:= \bigcup\{$SAT, SAT→SAT, (SAT→SAT)→SAT, SAT→SAT→SAT, ...$\}$, where the arrow denotes set-theoretic function space.

This construction will only prove SN for the objects of Fω, and it requires some further tricks to show that this implies SN for all terms of Fω. For CC the situation is more complicated, because constructors and kinds can also contain objects as subterms. So, even if one would have constructed mappings $\mathcal{V}$, $[\![-]\!]_\xi$ and $(\![-]\!)_\rho$ as above, it is not so easy to see how SN for the objects of CC implies SN for the full CC.





The solution that we propose here is to define the mapping $(\![-]\!)_\rho$ for all terms of CC. To show that the image of $(\![-]\!)_\rho$ is a strongly normalizing term, we also have to extend the mapping $[\![-]\!]_\xi$ to kinds. So, the kinds of CC will have two interpretations: first as sets under $\mathcal{V}$ ($\star$ is interpreted as SAT and the other kinds are interpreted by appropriate elements of $\{\mathrm{SAT}, \mathrm{SAT}{\rightarrow}\mathrm{SAT}, (\mathrm{SAT}{\rightarrow}\mathrm{SAT}){\rightarrow}\mathrm{SAT}, \mathrm{SAT}{\rightarrow}\mathrm{SAT}{\rightarrow}\mathrm{SAT}, \ldots\}$, where the arrow denotes set-theoretic function space, second as saturated sets (elements of SAT). This is done to allow an interpretation of constructors as pseudoterms under $(\![-]\!)_\rho$, making sure that the constructors are strongly normalizing as well. The new situation is visualized in the second picture.

3.2. The proof

Different from what is usually done, we don't define the saturated sets as sets of untyped λ terms, but as sets of pseudoterms. (So, SAT $\subset \wp(\mathrm{T})$ instead of SAT $\subset \wp(\Lambda)$.) This slight modification is not really important, but makes the technical presentation a bit shorter. Let in the following $\mathsf{SN} \subset \mathrm{T}$ be the set of pseudoterms that are Strongly Normalizing under β-reduction. The well-known notion of 'saturated set of terms' is defined in a slightly more general way than is necessary. This is done to make it easier to extend the proof of SN later.

3.1. Definition. The set of *base terms* $\mathcal{B}$ is defined by

1. $\mathsf{Var} \subset \mathcal{B}$ and $d \in \mathcal{B}$,
2. $\star, \Box \in \mathcal{B}$,
3. If $M \in \mathcal{B}$ and $N \in \mathsf{SN}$, then $MN \in \mathcal{B}$,
4. If $M, N \in \mathsf{SN}$, then $\Pi v{:}M.N \in \mathcal{B}$.

3.2. Definition. The *key redex* of an untyped lambda term is defined by

1. If M is a redex, then M is its own key redex,
2. If M has key redex N, then MP has key redex N.

The term that is obtained from M by contracting its key redex is denoted by $\mathsf{red}_k(M)$.

All base terms are SN. Note that the key redex of M is unique, if it exists. Furthermore, every key redex is a head redex (but not the other way around).

3.3. Definition. A set of untyped lambda terms X is *saturated* if

1. $X \subset \mathsf{SN}$,
2. $\mathcal{B} \subset X$,
3. If $\mathsf{red}_k(M) \in X$ and $M \in \mathsf{SN}$, then $M \in X$.

The collection of saturated sets is denoted by SAT.

This definition of saturated set is equivalent to saying that X is saturated if

1. $X \subset \mathsf{SN}$,
2. $\forall \mathbf{Q} \in \mathsf{SN} \forall v \in \mathsf{Var}[v\mathbf{Q} \in X]$,
3. $\forall \mathbf{Q}, M, N \in \mathsf{SN}[(\Pi v{:}M.N)\mathbf{Q} \in X]$,
4. $\forall \mathbf{Q}, M, P, N \in \mathsf{SN}[M[P/v]\mathbf{Q} \in X \Longrightarrow (\lambda v{:}N.M)P\mathbf{Q} \in X]$.

By definition, SN is itself saturated and all saturated sets are nonempty.

As we already pointed out, the types of CC will be interpreted as saturated sets. This requires some closure properties for the set of saturated sets which will be proved in Lemma 3.5. The set-interpretation of the kinds of CC (by the map $\mathcal{V}$) can be seen as first taking the underlying Fω-kind (which is a kind that consists of just the symbols $\rightarrow$ and $\star$), and then taking the set-interpretation of kinds of Fω. Here we define the set-interpretation of CC-kinds immediately.

3.4. Definition. For $A \in \mathsf{Kind}(\mathrm{CC})$, *the set-interpretation of* A, $\mathcal{V}(A)$, is defined inductively as follows.

$$\begin{aligned}\mathcal{V}(\star) &= \mathrm{SAT}\ (= \{X \mid X \subset \Lambda \text{ is saturated}\}),\\ \mathcal{V}(\Pi\alpha{:}B.C) &= \{f \mid f : \mathcal{V}(B)\rightarrow\mathcal{V}(C)\},\ \text{if } B \text{ is a kind},\\ \mathcal{V}(\Pi x{:}\sigma.C) &= \mathcal{V}(C) \text{ if } \sigma \text{ is a type}.\end{aligned}$$

The collection of all set-interpretations is denoted by $(\mathrm{SAT})^*$, so $(\mathrm{SAT})^* := \bigcup\{\mathcal{V}(A) \mid A \in \mathsf{Kind}(\mathrm{CC})\}$.

See the remark after Lemma 2.5, that justifies the case distinction in this definition.

The types are interpreted as saturated sets and the kinds also have a second interpretation as saturated sets. We need the following (well-known) closure properties on SAT.

3.5. Lemma. *The set of saturated sets (*SAT*) is closed under arbitrary intersections and function spaces. That is,*

1. *for I a set and X_i saturated for all $i \in I$, $\bigcap_{i\in I}X_i$ is saturated*
2. *for X and Y saturated, $X\rightarrow Y := \{M \in \Lambda \mid \forall N \in X[MN \in Y]\}$ is saturated.*

3.6. Definition. For Γ a context of CC, a *constructor valuation of Γ* is a map $\xi : \mathsf{Var}^{\square} \rightarrow (\mathrm{SAT})^*$ (notation $\xi \models^{\square} \Gamma$) such that

$$\alpha{:}A \in \Gamma \Longrightarrow \xi(\alpha) \in \mathcal{V}(A).$$

3.7. Definition. For Γ a context of CC and ξ a constructor valuation of Γ, the interpretation function

$$\llbracket - \rrbracket_\xi : \Gamma\text{-}\mathsf{Term}(\mathrm{CC}) \setminus \Gamma\text{-}\mathsf{Obj}(\mathrm{CC}) \rightarrow (\mathrm{SAT})^*$$

is defined inductively as follows.

$$\begin{aligned}\llbracket\star\rrbracket_\xi &= \llbracket\square\rrbracket_\xi = \mathsf{SN},\\ \llbracket\alpha\rrbracket_\xi &= \xi(\alpha),\\ \llbracket PQ\rrbracket_\xi &= \llbracket P\rrbracket_\xi(\llbracket Q\rrbracket_\xi),\ \text{if } Q \text{ is a constructor},\\ \llbracket Pt\rrbracket_\xi &= \llbracket P\rrbracket_\xi,\ \text{if } t \text{ is an object},\\ \llbracket\lambda\alpha{:}A.Q\rrbracket_\xi &= \boldsymbol{\lambda} a \in \mathcal{V}(A).\llbracket Q\rrbracket_{\xi(\alpha:=a)},\ \text{if } A \text{ is a kind},\\ \llbracket\lambda x{:}\sigma.Q\rrbracket_\xi &= \llbracket Q\rrbracket_\xi,\ \text{if } \sigma \text{ is a type},\\ \llbracket\Pi x{:}\sigma.T\rrbracket_\xi &= \llbracket\sigma\rrbracket_\xi\rightarrow\llbracket T\rrbracket_\xi,\ \text{if } \sigma \text{ is a type},\\ \llbracket\Pi\alpha{:}A.T\rrbracket_\xi &= \llbracket A\rrbracket_\xi\rightarrow \textstyle\bigcap_{a\in\mathcal{V}(A)} \llbracket T\rrbracket_{\xi(\alpha:=a)},\ \text{if } A \text{ is a kind}.\end{aligned}$$

The following Lemma states that the interpretations of the constructors under $[\![-]\!]_\xi$ are elements of the right set. As a matter of fact, it also states that $[\![-]\!]_\xi$ of definition 3.7, is well-defined (e.g. in the case for $[\![PQ]\!]_\xi$). The proof is by simultaneous induction on the structure of Q, respectively A.

3.8. Lemma (Soundness for $[\![-]\!]_\xi$). *For Γ a context of* CC, $Q, A \in \mathsf{Term}(\mathrm{CC})$ *and* $\xi \models^\square \Gamma$,

$$\Gamma \vdash Q : A(:\square) \Longrightarrow [\![Q]\!]_\xi \in \mathcal{V}(A),$$
$$\Gamma \vdash A : \square \Longrightarrow [\![A]\!]_\xi \in SAT.$$

It is easy to verify the substitution property for $[\![-]\!]_\xi$. From it one concludes that $[\![-]\!]_\xi$ preserves equality:

3.9. Fact. If $\xi \models^\square \Gamma$ and P is a constructor, t an object and Q a constructor or a kind in Γ, then $[\![Q[P/\alpha]]\!]_\xi = [\![Q]\!]_{\xi(\alpha:=[\![P]\!]_\xi)}$ and $[\![Q[t/x]]\!]_\xi = [\![Q]\!]_\xi$. Hence we have $Q =_\beta P \Longrightarrow [\![Q]\!]_\xi = [\![P]\!]_\xi$.

3.10. Definition. For Γ a context of CC and $\xi \models^\square \Gamma$, an *object valuation of Γ with respect to ξ* is a map $\rho : \mathsf{Var} \to \mathsf{T}$ (notation $\rho, \xi \models \Gamma$) such that

$$v : T \in \Gamma \Longrightarrow \rho(v) \in [\![T]\!]_\xi.$$

3.11. Definition. For Γ a context of CC with $\rho, \xi \models \Gamma$, the interpretation function

$$(\![-]\!)_\rho : \mathsf{T} \to \mathsf{T}$$

is defined as the extension of ρ to a substitution (for the free variables), so

$$(\![M]\!)_\rho := M[\rho(\mathbf{v})/\mathbf{v}].$$

Note that the interpretation of terms (by $(\![-]\!)_\rho$) does not depend on the interpretation of the constructors and kinds (by $[\![-]\!]_\xi$).

3.12. Definition. For Γ a context and M and T terms of CC, we say that Γ *satisfies that M is of type T*, notation $\Gamma \models M : T$ if

$$\forall \rho, \xi[\rho, \xi \models \Gamma \Longrightarrow (\![M]\!)_\rho \in [\![T]\!]_\xi].$$

3.13. Theorem (Soundness Theorem). *For Γ a context and M and T terms of* CC,

$$\Gamma \vdash M : T \Longrightarrow \Gamma \models M : T.$$

Proof. By induction on the structure of M we prove that if $\rho, \xi \models \Gamma$, then $(\![M]\!)_\rho \in [\![T]\!]_\xi$. So let ρ and ξ be valuations such that $\rho, \xi \models \Gamma$. We treat five cases.

- $M \equiv \lambda x{:}\tau.Q$ with τ a type and Q a constructor. Then $\Gamma, x{:}\tau \vdash Q : B$ for some B with $T =_\beta \Pi x{:}\tau.B$. By IH $(\!|\tau|\!)_\rho \in [\![\star]\!]_\xi$ (and hence $(\!|\tau|\!)_\rho \in \mathsf{SN}$) and also $(\!|Q|\!)_{\rho(x:=p)} \in [\![B]\!]_\xi$ for all $p \in [\![\tau]\!]_\xi$. So, $\lambda x{:}(\!|\tau|\!)_\rho.(\!|Q|\!)_{\rho(x:=x)} \in [\![\tau]\!]_\xi \to [\![B]\!]_\xi$. Hence we are done, because $(\!|\lambda x{:}\tau.Q|\!)_\rho = \lambda x{:}(\!|\tau|\!)_\rho.(\!|Q|\!)_{\rho(x:=x)} \in [\![\tau]\!]_\xi \to [\![B]\!]_\xi = [\![T]\!]_\xi$.
- $M \equiv \lambda\alpha{:}B.t$, with B a kind and t an object. Then $\Gamma, \alpha{:}B \vdash t : \tau$ for some τ with $T =_\beta \Pi\alpha{:}B.\tau$. By IH we find that $(\!|B|\!)_\rho \in [\![\square]\!]_\xi$ (and hence $(\!|B|\!)_\rho \in \mathsf{SN}$) and $(\!|t|\!)_{\rho(\alpha:=p)} \in [\![\tau]\!]_{\xi(\alpha:=f)}$ for all $f \in \mathcal{V}(B)$ and all $p \in [\![B]\!]_\xi$. Hence, $(\!|t|\!)_{\rho(\alpha:=p)} \in \bigcap_{f\in\mathcal{V}(B)} [\![\tau]\!]_{\xi(\alpha:=f)}$ for all $p \in [\![B]\!]_\xi$. But then $(\!|\lambda\alpha{:}B.t|\!)_\rho = \lambda\alpha{:}(\!|B|\!)_\rho.(\!|t|\!)_{\rho(\alpha:=\alpha)} \in [\![B]\!]_\xi \to \bigcap_{f\in\mathcal{V}(B)} [\![\tau]\!]_{\xi(\alpha:=f)} = [\![T]\!]_\xi$.
- $M \equiv tq$, with t and q objects. Then $\Gamma \vdash t : \Pi x{:}\tau.\sigma$ and $\Gamma \vdash q : \tau$ for some τ and σ with $\sigma[q/x] =_\beta T$. By IH $(\!|t|\!)_\rho \in [\![\tau]\!]_\xi \to [\![\sigma]\!]_\xi$ and $(\!|q|\!)_\rho \in [\![\tau]\!]_\xi$, so $(\!|tq|\!)_\rho = (\!|t|\!)_\rho (\!|q|\!)_\rho \in [\![\sigma]\!]_\xi = [\![T]\!]_\xi$. (Note that $[\![\sigma]\!]_\xi = [\![\sigma[q/x]]\!]_\xi$, due to Fact 3.9.)
- $M \equiv PQ$, with P and Q constructors. Then $\Gamma \vdash P : \Pi\alpha{:}A.B$ and $\Gamma \vdash Q : A$ for some B with $B[Q/\alpha] =_\beta T$. By IH $(\!|P|\!)_\rho \in [\![A]\!]_\xi \to \bigcap_{f\in\mathcal{V}(A)} [\![B]\!]_{\xi(\alpha:=f)}$ and $(\!|Q|\!)_\rho \in [\![A]\!]_\xi$, so $(\!|PQ|\!)_\rho = (\!|P|\!)_\rho (\!|Q|\!)_\rho \in \bigcap_{f\in\mathcal{V}(A)} [\![B]\!]_{\xi(\alpha:=f)}$. Furthermore, $[\![Q]\!]_\xi \in \mathcal{V}(A)$, so $(\!|PQ|\!)_\rho \in [\![B]\!]_{\xi(\alpha:=[\![Q]\!]_\xi)} = [\![T]\!]$.
- $M \equiv \Pi x{:}\sigma.B$, with σ a type and B a kind. Then $\Gamma \vdash \sigma : \star$, $\Gamma, x{:}\sigma \vdash B : \square$ and $T \equiv \square$. By IH $(\!|\sigma|\!)_\rho \in [\![\star]\!]_\xi$ and $(\!|B|\!)_{\rho(x:=p)} \in [\![\square]\!]_\xi$ for all $p \in [\![\sigma]\!]_\xi$. Hence $(\!|\sigma|\!)_\rho \in \mathsf{SN}$ and $(\!|B|\!)_{\rho(x:=x)} \in \mathsf{SN}$, so $(\!|\Pi x{:}\sigma.B|\!)_\rho \equiv \Pi x{:}(\!|\sigma|\!)_\rho.(\!|B|\!)_{\rho(x:=x)} \in \mathsf{SN} = [\![\square]\!]_\xi$. □

3.14. THEOREM.

$$\forall M \in \mathsf{Term}(\mathrm{CC})[\mathit{SN}(M)].$$

PROOF. Let M be a term of CC. Then either $M \equiv \square$ or $\Gamma \vdash M : T$ for some Γ and T. In the first case, M is of course SN. In the second case, $\Gamma \models M : T$ by the previous theorem. We define canonical elements c^A in the sets $\mathcal{V}(A)$ (for $A \in \mathsf{Kind}(\mathrm{CC})$) as follows.

$$\begin{aligned} c^\star &:= \mathsf{SN}, \\ c^{\Pi\alpha:A.B} &:= \boldsymbol{\lambda} f \in \mathcal{V}(A).c^B, \text{ if } A{:}\square \\ c^{\Pi x:\sigma.B} &:= c^B, \text{ if } \sigma{:}\,\star\,. \end{aligned}$$

For the constructor valuation for Γ we take ξ with $\xi(\alpha) = c^A$ if $\alpha{:}A \in \Gamma$ (and $\xi(\alpha)$ arbitrary otherwise), and for the object valuation for Γ with respect to this ξ we take ρ with $\rho(v) = v$. Now, $\rho, \xi \models \Gamma$ and so $(\!|M|\!)_\rho \in [\![T]\!]_\xi$, where $(\!|M|\!)_\rho$ is just M. Hence $M \in [\![T]\!]_\xi \subset \mathsf{SN}$, so M is SN. □

4. Beyond CC

The above proof of SN for CC is very flexible and can be extended to many other cases. The main cause for this flexibility is that the proof does not rely on too much (difficult) meta theory of CC. For one thing, we don't require the set of typable terms to be closed under reduction (the so called Subject Reduction property). The only two properties that are seriously used are the ones mentioned in Section 2.1, Classification (in a context a term can not be a type and a kind at the same time) and a strengthened version of Stripping (if $\Gamma \vdash \lambda v{:}T.M : U$, then $\Gamma', v{:}T \vdash M : T'$ with a smaller derivation, where Γ' is a begin-part of Γ and $\Pi v{:}T.T'$ is convertible with U via a path through the set of well-typed terms). For the Calculus of Constructions itself, these properties follow rather easily, but in general this is not the case. Therefore, in [Geuvers and Werner 1994], the notion of *soundness of a type system* is introduced, stating that if two terms M and N (of the same type in the same context) are convertible, then they are convertible via a path through the well-typed terms. It is also shown there that the extension of an arbitrary Pure Type System with η-conversion may not be sound. The reason for calling this property 'soundness' is that it implies the equivalence of the presentation of CC with a *typed conversion rule* with the presentation in Definition 2.2, in which the conversion is untyped.

If the soundness property is not satisfied, then the type system does not conform with our intuition that, if two types are convertible (and hence have the same inhabitants), then they are convertible as *well-typed* terms. So, as a matter of fact, the syntax with untyped conversion rule can only be accepted *after* one has shown that the soundness property holds for it.

Now, if we want to look at an extension of CC, we should not take the system with an untyped conversion rule as basic, because it may be the case that two types are equal as pseudoterms, while they are not convertible via a path through the well-typed terms. (And if that happens, the conversion rule can be applied in a situation where it shouldn't be applied.) Instead, we look at the system where the conversion rule has been replaced by a 'one-step reduction-expansion rule', as follows.

4.1. Definition. In the following, the conversion rule (conv) will not be the one in Definition 2.2, but the following.

$$(\text{conv})\,\frac{\Gamma \vdash M : T \quad \Gamma \vdash U : \star/\Box}{\Gamma \vdash M : U} \text{ if } U \longrightarrow T \text{ or } T \longrightarrow U$$

Here $\longrightarrow$ is a one-step-reduction. (In Section 3 this would be $\longrightarrow_\beta$.)

With this (conv) rule, we obtain the strengthening of Stripping that we are interested in: e.g. if $\Gamma \vdash \lambda v{:}T.M : U$, then $\Gamma', v{:}T \vdash M : T'$ with a smaller derivation, where Γ' is a begin-part of Γ and $\Pi v{:}T.T'$ is convertible with U via a path through the set of well-typed terms.

Another advantage of this slightly different conversion rule is that, in order to show the soundness of the (conv) rule in the proof of Theorem 3.13, one only has to show that if $Q \longrightarrow P$, then $[\![Q]\!]_\xi = [\![P]\!]_\xi$, for Q and P typable.

We treat some examples of extensions of CC and show that they are SN by adapting the proof of Section 3. The extensions that we treat are the ones with W-types (for representing types of well-founded trees), Σ-types and inductive kinds. Before studying these examples we list some general properties about saturated sets that will be used. These properties are proved for the saturated set notion as it has been given in the previous paragraph. For each extension of CC that is treated herefater, the notion of saturated set is slightly adapted, but the proofs of these properties will still go through.

4.1. Saturated sets

Saturated sets are sets of pseudoterms that contain all so-called 'base terms' and are closed under expanding a key redex. We define the notion of key reduction separately.

4.2. Definition. For M and N λ terms, we say that M key-reduces to N, notation $M \xrightarrow{k} N$ if N is obtained from M by contracting the key redex in M. The transitive reflexive closure of $\xrightarrow{k}$ is denoted by $\overset{k}{\twoheadrightarrow}$.

An easy fact about key reduction is that if X is a saturated set and $N \in X$ with $M \overset{k}{\twoheadrightarrow} N$ and $M \in \mathsf{SN}$, then $M \in X$.

We have already seen two constructions that can be performed on saturated sets, namely the function space construction and the intersection. There are many more of those, some of which will be defined and used later. An important trivial fact about SAT is the following.

4.3. Fact. SAT is a complete lattice. The ordering is the inclusion and suprema and infima are given by union and intersection, respectively.

4.4. Definition. A *morphism* from SAT to SAT is an expression $\Phi(X)$ built up from variables ranging over SAT (among which X is one), arrows and intersections. A morphism $\Phi(X)$ is *positive* if X occurs only to the left of an even number of arrows. It is *negative* if X occurs only to the left of an odd number of arrows.

In Definition 4.4 we allow arbitrary intersections, so if $\Phi_i(X)$ is a morphism for every $i \in I$, then $\Phi(X) = \bigcap_{i \in I} \Phi_i(X)$ is also a morphism. This morphism is positive (resp. negative) if $\Phi_i(X)$ is positive (resp. negative) for every $i \in I$.

A positive morphism is indeed monotone, as one would expect. This is stated in the following Lemma, which is proved by induction on the structure of $\Phi(X)$.

4.5. Lemma. *If $\Phi(X)$ is a positive morphism, then $\boldsymbol{\lambda} X.\Phi(X)$ is monotone increasing ($Y \subset Z \Longrightarrow \Phi(Y) \subset \Phi(Z)$) and if $\Phi(X)$ is a negative morphism, then $\boldsymbol{\lambda} X.\Phi(X)$ is monotone decreasing ($Y \subset Z \Longrightarrow \Phi(Z) \subset \Phi(Y)$).*

The following is an immediate consequence of the fact that a positive morphism is a monotone increasing function on the complete lattice of saturated sets.

4.6. Corollary. *If $\Phi(X)$ is a positive morphism on* SAT, *then there is a smallest saturated set* $\mathrm{lfp}(\Phi)$ *for which* $\Phi(\mathrm{lfp}(\Phi)) = \mathrm{lfp}(\Phi)$.

4.2. CC with W-types

We now look at the extension of CC with Martin-Löf's W-types, a type constructor for representing types of well-founded trees. (See [Martin-Löf 1984] or [Nordström et al. 1990] for an extensive treatment of W-types and examples.) We just give the rules for W-types and the proof that the addition of these rules to CC preserves the SN property.

4.7. Definition. The *Calculus of Constructions with W-types*, CC^W, has the following additional rules.

$$(W)\qquad \frac{\Gamma \vdash \sigma : \star \quad \Gamma, x{:}\sigma \vdash \tau : \star}{\Gamma \vdash Wx{:}\sigma.\tau : \star}$$

$$(\mathrm{sup})\qquad \frac{\Gamma \vdash p : \sigma \quad \Gamma \vdash q : \tau[p/x]{\rightarrow}Wx{:}\sigma.\tau}{\Gamma \vdash \mathsf{sup}(p,q) : Wx{:}\sigma.\tau}$$

$$(\mathrm{wrec})\ \frac{\Gamma \vdash Q : (Wx{:}\sigma.\tau){\rightarrow}\star \quad \Gamma \vdash t : \Pi x{:}\sigma.\Pi z{:}\tau{\rightarrow}Wx{:}\sigma.\tau.(\Pi y{:}\tau.Q(zy)){\rightarrow}Q(\mathsf{sup}(x,z))}{\Gamma \vdash \mathsf{wrec}\, t : \Pi w{:}(Wx{:}\sigma.\tau).Qw}$$

The reduction rule associated with wrec and $\mathsf{sup}(-,-)$ is

$$\mathsf{wrec}\, t(\mathsf{sup}(p,q)) \longrightarrow_w tpq(\lambda y{:}\tau[p/x].\mathsf{wrec}\, t(qy)).$$

The conversion rule is adapted to this new reduction.

Now, we extend the untyped λ calculus with wrec and $\mathsf{sup}(-,-)$ operators that have the reduction behaviour

$$\mathsf{wrec}\, P(\mathsf{sup}(N,Q)) \longrightarrow_w PNQ(\lambda y.\mathsf{wrec}\, P(Qy)).$$

The definition of the set of base terms $\mathcal{B}$ is adapted by adding to Definition 3.1 the clauses
5. If $M \in \mathcal{B}$ and $P \in \mathsf{SN}$, then $\mathsf{wrec}\, PM \in \mathcal{B}$,
6. If $M, N \in \mathsf{SN}$, then $Wx{:}M.N \in \mathcal{B}$.
The notion of key redex is extended by adding to Definition 3.2 the clause
3. If M has key redex N, then $\mathsf{wrec}\, PM$ has key redex N (for any P).

The definition of saturated set is the same as in Definition 3.3, with the notions of 'base term' and 'key redex' replaced by the above ones. This new collection of saturated sets is ambiguously denoted by SAT (but there will be no confusion).

4.8. Definition. For $X, Y \in \mathrm{SAT}$, the saturated set $W(X,Y)$ is defined by

$$W(X,Y) := \mathrm{lfp}(\boldsymbol{\lambda} W.\ \{M \mid \forall Z \in \mathrm{SAT} \forall P \in X{\rightarrow}(Y{\rightarrow}W){\rightarrow}(Y{\rightarrow}Z){\rightarrow}Z[\mathsf{wrec}\, PM \in Z]\}).$$

That this least fixed point exists is due to the fact that

$$\lambda W.\{M \mid \forall Z \in \mathrm{SAT}\forall P \in X{\to}(Y{\to}W){\to}(Y{\to}Z){\to}Z[\mathsf{wrec}\, PM \in Z]\}$$

is a monotone function on SAT. This can be seen as follows.

Write $\Phi(W)$ for $\{M \mid \forall Z\forall P \in X{\to}(Y{\to}W){\to}(Y{\to}Z){\to}Z[\mathsf{wrec}\, PM \in Z]\}$ and let W, $W' \in \mathrm{SAT}$, with $W \subset W'$. Let $M \in \Phi(W)$ Then, for all Z and for all $P \in X{\to}(Y{\to}W){\to}(Y{\to}Z){\to}Z$, we have $\mathsf{wrec}\, PM \in Z$. Now, W is negative in $X{\to}(Y{\to}W){\to}(Y{\to}Z){\to}Z$, so
$\forall Z\forall P \in X{\to}(Y{\to}W'){\to}(Y{\to}Z){\to}Z[\mathsf{wrec}\, PM \in Z]$ and so $M \in \Phi(W')$.

The set $W(X,Y)$ can equivalently be defined as
$\cap\{W \mid \mathsf{wrec} \in \cap_{Z\in\mathrm{SAT}}\ (X{\to}(Y{\to}W){\to}(Y{\to}Z){\to}Z){\to}W{\to}Z\}$. The essential closure properties for the W-constructor on SAT are the following.

4.9. Lemma. *Let X and Y be saturated sets and write W for $W(X,Y)$.*
1. If $M \in X$ and $N \in Y{\to}W$, then $\mathsf{sup}(M,N) \in W$.
2. If $P \in X{\to}(Y{\to}W){\to}(Y{\to}Z){\to}Z$, then $\mathsf{wrec}\, P \in W{\to}Z$.

Proof. We use the fact that

$$W = \{M \mid \forall Z\forall P \in X{\to}(Y{\to}W){\to}(Y{\to}Z){\to}Z[\mathsf{wrec}\, PM \in Z]\}).$$

For the first, let $Z \in \mathrm{SAT}$ and $P \in X{\to}(Y{\to}W){\to}(Y{\to}Z){\to}Z$. Then $\mathsf{wrec}\, P(\mathsf{sup}(M,N)) \xrightarrow{k} PMN(\lambda y.\mathsf{wrec}\, P(Ny) \in Z$ and $\mathsf{wrec}\, P(\mathsf{sup}(M,N))$ is SN, so $\mathsf{wrec}\, P(\mathsf{sup}(M,N)) \in Z$ and hence $\mathsf{sup}(M,N) \in W$. For the second, let $M \in W$. Then $\mathsf{wrec}\, PM \in Z$ by definition, so $\mathsf{wrec}\, P \in W{\to}Z$. □

The definition of set-interpretation of 3.4 does not have to be extended, because there are no kinds of the form $Wx{:}\sigma.\tau$. The notion of '$\xi \models^{\Box} \Gamma$' is defined analoguously to Definition 3.6.

4.10. Definition. The function $[\![-]\!]_\xi$ is defined by extending Definition 3.7 with the following clause.

$$[\![Wx{:}\sigma.\tau]\!]_\xi = W([\![\sigma]\!]_\xi, [\![\tau]\!]_\xi).$$

We have the following property for the extended $[\![-]\!]_\xi$.

4.11. Fact. Let Q and P be constructors or kinds with $\Gamma \vdash Q, P : T$ and ξ a valuation with $\xi \models \Gamma$, then

$$Q \longrightarrow_{\beta w} P \Longrightarrow [\![Q]\!]_\xi = [\![P]\!]_\xi.$$

The Soundness Lemma 3.8 is also easily verified:

4.12. Lemma (Soundness for $[\![-]\!]_\xi$). *For Γ a context of CC^W, $Q, A \in \mathsf{Term}(\mathrm{CC}^W)$ and $\xi \models^{\Box} \Gamma$,*

$$\Gamma \vdash Q : A(:\Box) \Longrightarrow [\![Q]\!]_\xi \in \mathcal{V}(A),$$
$$\Gamma \vdash Q : \Box \Longrightarrow [\![Q]\!]_\xi \in \mathrm{SAT}.$$

Let ρ be a valuation that assigns terms to the free variables, as in Definition 3.10. Let also $\llparenthesis - \rrparenthesis_\rho$ be the extension of the valuation ρ to a substitution ($\llparenthesis - \rrparenthesis_\rho : \mathrm{T}{\rightarrow}\mathrm{T}$) as defined in Definition 3.11.

The Strong Normalization follows immediately from the Soundness Theorem for $\llparenthesis - \rrparenthesis_\rho$. To prove the soundness we only have to verify the extra cases that arise from the additional derivation rules.

4.13. Theorem (Soundness Theorem). *For Γ a context and M and T terms of* CC^W,

$$\Gamma \vdash M : T \Longrightarrow \Gamma \models M : T.$$

Proof. By induction on the derivation; we verify the two relevant cases, using Lemma 4.9. Let ρ and ξ be valuations such that $\rho, \xi \models \Gamma$.

- $M \equiv \mathsf{wrec}\, t$ with $\Gamma \vdash t : \Pi x{:}\sigma.\Pi z{:}\tau{\rightarrow}Wx{:}\sigma.\tau.(\Pi y{:}\tau.Q(zy)){\rightarrow}Q(\mathsf{sup}(x,z))$, $\Gamma \vdash Q : (Wx{:}\sigma.\tau){\rightarrow}\star$ and $T \equiv \Pi w{:}Wx{:}\sigma.\tau.Qw$. By IH $\llparenthesis t \rrparenthesis_\rho \in \llbracket\sigma\rrbracket_\xi{\rightarrow}(\llbracket\tau\rrbracket_\xi{\rightarrow}W(\llbracket\sigma\rrbracket_\xi, \llbracket\tau\rrbracket_\xi)){\rightarrow}(\llbracket\tau\rrbracket_\xi{\rightarrow}\llbracket Q\rrbracket_\xi){\rightarrow}\llbracket Q\rrbracket_\xi$, so $\llparenthesis \mathsf{wrec}\, t \rrparenthesis_\rho = \mathsf{wrec}\,\llparenthesis t \rrparenthesis_\rho \in W(\llbracket\sigma\rrbracket_\xi, \llbracket\tau\rrbracket_\xi){\rightarrow}\llbracket Q\rrbracket_\xi (= \llbracket T\rrbracket_\xi)$.
- $M \equiv \mathsf{sup}(p,q)$ with $\Gamma \vdash p{:}\sigma$, $\Gamma \vdash q{:}\tau[p/x]{\rightarrow}Wx{:}\sigma.\tau$ and $T \equiv Wx{:}\sigma.\tau$. By IH $\llparenthesis p \rrparenthesis_\rho \in \llbracket\sigma\rrbracket_\xi$ and $\llparenthesis q \rrparenthesis_\rho \in \llbracket\tau\rrbracket_\xi{\rightarrow}W(\llbracket\sigma\rrbracket_\xi, \llbracket\tau\rrbracket_\xi)$. Hence, $\mathsf{sup}(\llparenthesis p \rrparenthesis_\rho, \llparenthesis q \rrparenthesis_\rho) \in W(\llbracket\sigma\rrbracket_\xi, \llbracket\tau\rrbracket_\xi)(= \llbracket T\rrbracket_\xi)$. □

The proof of the following corollary is now totally similar to the proof of Theorem 3.14.

4.14. Corollary.

$$\forall M \in \mathsf{Term}(\mathrm{CC}^W)[SN(M)].$$

5. CC with Σ-types, extending the method to inductive kinds

It is well-known that one can not extend CC with arbitrary Σ-types: $\Sigma\alpha{:}A.\sigma : \star$ is not allowed if $A : \square$. (If one allows this, it is possible to type non-normalizing terms.) In the proof of SN for CC with 'safe' Σ-types that we give here, it can be seen why the proof-construction does not extend to the 'unsafe' Σ-types.

In order to treat Σ-types, we have to modify the proof of Section 3. This modification turns out to be of more general importance, since it also allows the interpretation of inductive kinds (like a kind of natural numbers that allows the same flexibility as the inductive type of natural numbers in Coq). This modification will be discussed later.

We now first give the rules for Σ-types.

5.1. Definition. The *Calculus of Constructions with Σ-types*, CC^Σ, has the following additional rules. (In these rules s, s_1 and s_2 stand for $\star$ or $\square$.)

$$(\Sigma^\star)\,\frac{\Gamma \vdash \sigma : \star \quad \Gamma, x{:}\sigma \vdash \tau : \star}{\Gamma \vdash \Sigma x{:}\sigma.\tau : \star} \qquad (\Sigma^\square)\,\frac{\Gamma \vdash T : s_1 \quad \Gamma, v{:}T \vdash U : s_2}{\Gamma \vdash \Sigma v{:}T.U : \square}$$

if $s_1 \equiv \square$ or $s_2 \equiv \square$,

$$(\mathrm{proj}_1)\,\frac{\Gamma \vdash M : \Sigma v{:}T.U}{\Gamma \vdash \pi_1 M : T} \qquad (\mathrm{proj}_2)\,\frac{\Gamma \vdash M : \Sigma v{:}T.U}{\Gamma \vdash \pi_2 M : U[\pi_1(M)/v]}$$

$$\text{(pair)}\ \frac{\Gamma \vdash M : T \quad \Gamma \vdash N : U[M/v] \quad \Gamma, v{:}T \vdash U : s}{\Gamma \vdash \langle M, N\rangle : \Sigma v{:}T.U}$$

The reduction rules associated with pairing and projection are

$$\pi_1\langle M, N\rangle \longrightarrow_\pi M, \quad \pi_2\langle M, N\rangle \longrightarrow_\pi N.$$

The conversion rule is adapted to this new reduction, that is, the side condition $T \longrightarrow U$ now stands for $\longrightarrow_{\beta\pi}$, the equivalence relation generated from β- and π- reduction. For convenience we shall speak of CC^{Σ^*} in case we want to restrict to Σ-types of the first sort, so $\Sigma v{:}T.U$, where T and U are types. ($T : *$ and $U : *$)

5.1. Small Σ-types

The proof of SN for CC^{Σ^*} is a direct extension of the proof of SN for CC. We first extend the untyped λ calculus with pairing and projection operators $\langle -, -\rangle$, π_1 and π_2 that have the required reduction behaviour

$$\pi_i(\langle M_1, M_2\rangle) \longrightarrow_\pi M_i \ \ (i \in \{1, 2\}).$$

5.2. Definition. For CC^{Σ^*}, the set of base terms $\mathcal{B}$ is defined by adding to Definition 3.1 the clauses
5. If $M \in \mathcal{B}$, then $\pi_1 M \in \mathcal{B}$ and $\pi_2 M \in \mathcal{B}$,
6. If $M, N \in \mathsf{SN}$, the $\Sigma v{:}M.N \in \mathcal{B}$.
The notion of key redex is extended by adding to Definition 3.2 the clause
3. If M has key redex N, then $\pi_i M$ has key redex N (for $i \in \{1, 2\}$).

The definition of saturated set is the same as in Definition 3.3, with the notions of 'base term' and 'key redex' replaced by the above ones. We ambiguously denote this new collection of saturated sets again by SAT (but there will be no confusion).

5.3. Definition. For $X, Y \in \mathrm{SAT}$, the *product of* X *and* Y, $X \times Y$ is defined by

$$X \times Y := \{M \mid \pi_1 M \in X \ \&\ \pi_2 M \in Y\}.$$

That SAT is closed under products and that elements of product sets behave correctly is stated in the following two lemmas. (The first is immediate.)

5.4. Lemma. *If $X, Y \in \mathrm{SAT}$ then $X \times Y \in \mathrm{SAT}$.*

5.5. Lemma. *Let X, Y and X_i ($\forall i \in I$) be saturated sets.*
1. If $M \in X$ and $N \in Y$, then $\langle M, N\rangle \in X \times Y$.
2. If $M \in X \times Y$, then $\pi_1 M \in X$ and $\pi_2 M \in Y$.

Proof. The second follows immediately from the definition of product. For the first, note that $\pi_1(\langle M, N\rangle) \xrightarrow{k} M \in X$ and $\pi_1(\langle M, N\rangle)$ is SN, hence $\pi_1(\langle M, N\rangle) \in X$. Similarly, $\pi_2(\langle M, N\rangle) \xrightarrow{k} N \in Y$, so $\pi_2(\langle M, N\rangle) \in Y$. □

The notion of '$\xi \models^{\Box} \Gamma$' is defined analoguously to Definition 3.6.

5.6. DEFINITION. The function $[\![-]\!]_\xi$ is defined for CC^{Σ^*} by extending Definition 3.7 with the clause

$$[\![\Sigma x{:}\sigma.\tau]\!]_\xi = [\![\sigma]\!]_\xi \times [\![\tau]\!]_\xi.$$

We have the following property. (Compare with Fact 3.9.)

5.7. FACT. Let Q and P be constructors or kinds with $\Gamma \vdash Q, P : T$ and ξ a valuation with $\xi \models \Gamma$, then

$$Q \longrightarrow_{\beta\pi c} P \Longrightarrow [\![Q]\!]_\xi = [\![P]\!]_\xi.$$

The Soundness Lemma 3.8 is also easily verified:

5.8. LEMMA (Soundness for $[\![-]\!]_\xi$). *For Γ a context of* CC^{Σ^*}, $Q, A \in \mathsf{Term}(\mathrm{CC}^{\Sigma^*})$ *and $\xi \models^{\Box} \Gamma$,*

$$\Gamma \vdash Q : A({:}\Box) \Longrightarrow [\![Q]\!]_\xi \in \mathcal{V}(A),$$
$$\Gamma \vdash Q : \Box \Longrightarrow [\![Q]\!]_\xi \in SAT.$$

The interpretation of typable terms as (strongly normalizing) pseudoterms is again done modulo a valuation ρ that assigns terms to the free variables. So, let ρ be as in Definition 3.10. The the interpretation $(\![-]\!)_\rho : \mathsf{T} \to \mathsf{T}$ is (as in Definition 3.11) defined as the extension of ρ to a substitution.

The Strong Normalization follows immediately from the Soundness Theorem for $(\![-]\!)_\rho$. To prove the soundness we only have to verify the extra cases that arise from the additional derivation rules. This is straightforward.

5.9. THEOREM (Soundness Theorem). *For Γ a context and M and T terms of* CC^{Σ^*},

$$\Gamma \vdash M : T \Longrightarrow \Gamma \models M : T.$$

The following is now immediate by taking the right valuations ξ and ρ.

5.10. COROLLARY.

$$\forall M \in \mathsf{Term}(\mathrm{CC}^{\Sigma^*})[SN(M)].$$

5.2. Large Σ-types

We now come to the interpretation of so called 'large' Σ-types (i.e. where the Σ-type is actually a kind) as saturated sets. It turns out that if σ is a type, then $[\![\Sigma x{:}\sigma.B]\!]_\xi$ can be defined as $[\![\sigma]\!]_\xi \times [\![B]\!]_\xi$. ($\xi$ does not give a value to object variables, so the interpretation of B under $[\![-]\!]_\xi$ does not depend on elements from $[\![\sigma]\!]_\xi$.) If A is a kind, then one can *not* define $[\![\Sigma\alpha{:}A.T]\!]_\xi := [\![A]\!]_\xi \times [\![T]\!]_\xi$, because now $[\![T]\!]_\xi$ depends on the value that ξ takes for α. One would like to define a 'dependent product of saturated sets' and interpret $\Sigma\alpha{:}A.T$ as such a dependent product. This turns out to be very complicated and we therefore take a different approach.

Instead of interpreting kinds as saturated sets under $\llbracket - \rrbracket_\xi$, we interpret kinds as saturated sets parametrized over their set-interpretation. So, if A is a kind, we define $\llbracket A \rrbracket_\xi$ as a function from $\mathcal{V}(A)$ to SAT. For the interpretation of types we take (as before) saturated sets. Then the statement of Soundness of the interpretation will have the following form.

$$\begin{aligned} \Gamma \vdash t{:}\sigma &\Longrightarrow \forall \rho, \xi \models \Gamma[\llparenthesis t \rrparenthesis_\rho \in \llbracket \sigma \rrbracket_\xi], \\ \Gamma \vdash P{:}A &\Longrightarrow \forall \rho, \xi \models \Gamma[\llparenthesis P \rrparenthesis_\rho \in \llbracket A \rrbracket_\xi(\llbracket P \rrbracket_\xi)], \end{aligned}$$

where σ stands for a type and A for a kind.

We now make precise how the definitions of $\mathcal{V}$, $\llbracket - \rrbracket_\xi$ and $\llparenthesis - \rrparenthesis_\rho$ have to be adapted to achieve the above.

5.11. Definition. The extension of the set-interpretation $\mathcal{V}$ to the kinds of CC^Σ is done by adding the following clauses to Definition 3.4.

$$\begin{aligned} \mathcal{V}(\Sigma\alpha{:}A.B) &:= \mathcal{V}(A) \times \mathcal{V}(B), \text{ if } A, B{:}\Box, \\ \mathcal{V}(\Sigma\alpha{:}A.\tau) &:= \mathcal{V}(A), \text{ if } A{:}\Box \text{ and } \tau{:}\star, \\ \mathcal{V}(\Sigma x{:}\sigma.B) &:= \mathcal{V}(B), \text{ if } B{:}\Box \text{ and } \sigma{:}\star. \end{aligned}$$

The notion of $\xi \models^\Box \Gamma$ (the constructor valuation ξ satisfies Γ) is as before in Definition 3.6.

5.12. Definition. The extension of $\llbracket - \rrbracket_\xi$ (definition 3.7) to CC^Σ is done by changing the clauses for $\star$ and Π-kinds and by adding clauses for Σ-types and its constructors as follows.

$$\begin{aligned} \llbracket \star \rrbracket_\xi &= \boldsymbol{\lambda} X \in \mathrm{SAT}.\mathsf{SN}, \\ \llbracket \Pi x{:}\sigma.B \rrbracket_\xi &= \boldsymbol{\lambda} b \in \mathcal{V}(B).\llbracket \sigma \rrbracket_\xi \to \llbracket B \rrbracket_\xi(b), \\ \llbracket \Pi \alpha{:}A.B \rrbracket_\xi &= \boldsymbol{\lambda} f \in \mathcal{V}(A) \to \mathcal{V}(B). \bigcap_{a \in \mathcal{V}(A)} \llbracket A \rrbracket_\xi(a) \to \llbracket B \rrbracket_{\xi(\alpha:=a)}(fa), \\ \llbracket \Sigma x{:}\sigma.B \rrbracket_\xi &= \boldsymbol{\lambda} b \in \mathcal{V}(B).\llbracket \sigma \rrbracket_\xi \times \llbracket B \rrbracket_\xi(b), \\ \llbracket \Sigma \alpha{:}A.B \rrbracket_\xi &= \boldsymbol{\lambda} p \in \mathcal{V}(A) \times \mathcal{V}(B).\llbracket A \rrbracket_\xi(\mathrm{fst}\,(p)) \times \llbracket B \rrbracket_{\xi(\alpha:=\mathrm{fst}\,(p))}(\mathrm{snd}\,(p)). \\ \llbracket \Sigma \alpha{:}A.\tau \rrbracket_\xi &= \boldsymbol{\lambda} a \in \mathcal{V}(A).\llbracket A \rrbracket_\xi(a) \times \llbracket \sigma \rrbracket_{\xi(\alpha:=a)}, \\ \llbracket \langle P, Q \rangle \rrbracket_\xi &= (\llbracket P \rrbracket_\xi, \llbracket Q \rrbracket_\xi), \\ \llbracket \langle P, q \rangle \rrbracket_\xi &= \llbracket P \rrbracket_\xi \\ \llbracket \langle p, Q \rangle \rrbracket_\xi &= \llbracket Q \rrbracket_\xi, \\ \llbracket \pi_1 Q \rrbracket_\xi &= \llbracket Q \rrbracket_\xi, \text{ if } Q : \Sigma\alpha{:}A.\tau \text{ with } \tau \text{ a'type}, \\ \llbracket \pi_1 Q \rrbracket_\xi &= \mathrm{fst}\,(\llbracket Q \rrbracket_\xi), \text{ if } Q : \Sigma\alpha{:}A.B \text{ with } B \text{ a kind}, \\ \llbracket \pi_2 Q \rrbracket_\xi &= \llbracket Q \rrbracket_\xi, \text{ if } Q : \Sigma x{:}\sigma.B \text{ with } \sigma \text{ a type}, \\ \llbracket \pi_2 Q \rrbracket_\xi &= \mathrm{snd}\,(\llbracket Q \rrbracket_\xi), \text{ if } Q : \Sigma\alpha : A.B \text{ with } A \text{ a kind}. \end{aligned}$$

Here, $\rightarrow$ denotes set-theoretic function space construction if it is in the subscript of a $\bigcap$; otherwise it denotes the function space on saturated sets. Furthermore, $(-,-)$ denotes pairing and fst and snd denote projections in set-theory. Remember that σ and τ stand for types, A and B stand for kinds, p and q stand for objects and P and Q stand for constructors.

It is now easy to verify the substitution property for $\llbracket - \rrbracket_\xi$ and to show that $\llbracket - \rrbracket_\xi$ preserves reduction (compare with Fact 3.9): $\llbracket M[Q/\alpha] \rrbracket_\xi = \llbracket M \rrbracket_{\xi(\alpha:=\llbracket Q \rrbracket_\xi)}$, $\llbracket M[q/x] \rrbracket_\xi = \llbracket M \rrbracket_\xi$ and if $M \longrightarrow_{\beta\pi} N$, then $\llbracket M \rrbracket_\xi = \llbracket N \rrbracket_\xi$, provided that M is a kind or a constructor.

Hence we can prove the following Soundness Lemma (compare with Lemma 3.8 and Lemma 5.8) by simultaneous induction on the derivation.

5.13. LEMMA (Soundness Lemma). *For Γ a context of* CC^Σ, $Q, A \in \mathsf{Term}(\mathrm{CC}^\Sigma)$ *and* $\xi \models^\square \Gamma$,

$$\Gamma \vdash Q : A(:\square) \Longrightarrow \llbracket Q \rrbracket_\xi \in \mathcal{V}(A),$$
$$\Gamma \vdash A : \square \Longrightarrow \llbracket A \rrbracket_\xi \in \mathcal{V}(A) \rightarrow SAT.$$

To define the interpretation $\llparenthesis - \rrparenthesis_\rho$, we have to say when a valuation ρ *satisfies Γ with respect to ξ* (notation $\rho, \xi \models \Gamma$; see also Definition 3.10).

5.14. DEFINITION. For $\rho;\mathsf{Var} \rightarrow \mathsf{T}$, we say that ρ *satisfies Γ with respect to ξ* (notation $\rho, \xi \models \Gamma$) when

$$x{:}\sigma \in \Gamma \Longrightarrow \rho(x) \in \llbracket \sigma \rrbracket_\xi,$$
$$\alpha{:}A \in \Gamma \Longrightarrow \rho(\alpha) \in \llbracket A \rrbracket_\xi(\xi(\alpha)).$$

The interpretation of objects, constructors and kinds of CC^Σ under $\llparenthesis - \rrparenthesis_\rho$ is done by extending the valyation ρ to a substitution $\llparenthesis - \rrparenthesis_\rho : \mathsf{T} \rightarrow \mathsf{T}$ (see Definition 3.11).

The notion of $\Gamma \models M : T$ (Γ satisfies that M is of type T) now takes the following form. (Compare with Definition 3.12.)

5.15. DEFINITION. For Γ a context and t an object, σ a type, P a constructor and A a kind of CC^Σ, we define

$$\Gamma \models t{:}\sigma \ \text{ iff } \ \forall \xi, \rho[\rho, \xi \models \Gamma \Longrightarrow \llparenthesis t \rrparenthesis_\rho \in \llbracket \sigma \rrbracket_\xi],$$
$$\Gamma \models P{:}A \ \text{ iff } \ \forall \xi, \rho[\rho, \xi \models \Gamma \Longrightarrow \llparenthesis P \rrparenthesis_\rho \in \llbracket A \rrbracket_\xi(\llbracket P \rrbracket_\xi)].$$

5.16. THEOREM (Soundness Theorem). *For Γ a context and M and T terms of* CC^Σ,

$$\Gamma \vdash M : T \Longrightarrow \Gamma \models M : T.$$

PROOF. The proof is by induction on the derivation. We treat a few cases.

- $M \equiv \langle P, t\rangle$ with $P{:}A$ and $t{:}\tau[P/\alpha]$. Then by IH, $\llparenthesis P\rrparenthesis_\rho \in \llbracket A\rrbracket_\xi(\llbracket P\rrbracket_\xi)$ and $\llparenthesis t\rrparenthesis_\rho \in \llbracket \tau[P/\alpha]\rrbracket_\xi (= \llbracket\tau\rrbracket_{\xi(\alpha:=\llbracket P\rrbracket_\xi)})$. Then $\llparenthesis\langle P,t\rangle\rrparenthesis_\rho \equiv \langle\llparenthesis P\rrparenthesis_\rho, \llparenthesis t\rrparenthesis_\rho\rangle \in$ $\llbracket A\rrbracket_\xi(\llbracket P\rrbracket_\xi) \times \llbracket\tau\rrbracket_{\xi(\alpha:=\llbracket P\rrbracket_\xi)} = \llbracket\Sigma\alpha{:}A.\tau\rrbracket_\xi(\llbracket P\rrbracket_\xi) = \llbracket\Sigma\alpha{:}A.\tau\rrbracket_\xi(\llbracket\langle P,t\rangle\rrbracket_\xi)$.
- $M \equiv \langle P, Q\rangle$ with $P{:}A$ and $Q{:}B[P/\alpha]$. Then by IH, $\llparenthesis P\rrparenthesis_\rho \in \llbracket A\rrbracket_\xi(\llbracket P\rrbracket_\xi)$ and $\llparenthesis Q\rrparenthesis_\rho \in \llbracket B[P/\alpha]\rrbracket_\xi(\llbracket Q\rrbracket_\xi)(= \llbracket B\rrbracket_{\xi(\alpha:=\llbracket P\rrbracket_\xi)}(\llbracket Q\rrbracket_\xi))$. Then, $\llparenthesis\langle P,Q\rangle\rrparenthesis_\rho \equiv$ $\langle\llparenthesis P\rrparenthesis_\rho, \llparenthesis Q\rrparenthesis_\rho\rangle \in \llbracket A\rrbracket_\xi(\llbracket P\rrbracket_\xi) \times \llbracket B\rrbracket_{\xi(\alpha:=\llbracket P\rrbracket_\xi)}(\llbracket Q\rrbracket_\xi) = \llbracket\Sigma\alpha{:}A.B\rrbracket_\xi(\llbracket\langle P,Q\rangle\rrbracket_\xi)$.
- $M \equiv \pi_1 P$ with $P{:}\Sigma\alpha{:}A.\tau$. Then by IH, $\llparenthesis P\rrparenthesis_\rho \in \llbracket\Sigma\alpha{:}A.\tau\rrbracket_\xi(\llbracket P\rrbracket_\xi)$, that is $\llparenthesis P\rrparenthesis_\rho \in \llbracket A\rrbracket_\xi(\llbracket P\rrbracket_\xi) \times \llbracket\sigma\rrbracket_{\xi(\alpha:=\llbracket P\rrbracket_\xi)}$. So, $\llparenthesis\pi_1 P\rrparenthesis_\rho = \pi_1\llparenthesis P\rrparenthesis_\rho \in \llbracket A\rrbracket_\xi(\llbracket P\rrbracket_\xi) =$ $\llbracket A\rrbracket_\xi(\llbracket\pi_1 P\rrbracket_\xi)$.
- $M \equiv \pi_2 P$ with $P{:}\Sigma\alpha{:}A.B$. Then by IH, $\llparenthesis P\rrparenthesis_\rho \in \llbracket\Sigma\alpha{:}A.B\rrbracket_\xi(\llbracket P\rrbracket_\xi)$, that is, $\llparenthesis P\rrparenthesis_\rho \in \llbracket A\rrbracket_\xi(\text{fst}\,\llbracket P\rrbracket_\xi) \times \llbracket B\rrbracket_{\xi(\alpha:=\text{fst}\,\llbracket P\rrbracket_\xi)}(\text{snd}\,\llbracket P\rrbracket_\xi)$. So, $\llparenthesis\pi_2 P\rrparenthesis_\rho = \pi_2\llparenthesis P\rrparenthesis_\rho \in$ $\llbracket B\rrbracket_{\xi(\alpha:=\llbracket\pi_1 P\rrbracket_\xi)}(\llbracket\pi_2 P\rrbracket_\xi) = \llbracket B[\pi_1 P/\alpha]\rrbracket_\xi(\llbracket\pi_2 P\rrbracket_\xi)$. □

The following is now an immediate consequence of the fact that we have for every context Γ a constructor valuation ξ such that $\xi \models^{\Box} \Gamma$ and furthermore, that for the identity valuation ρ_0, we have $\rho_0, \xi \models \Gamma$. (See the proof of 3.14 for details.)

5.17. Corollary (Strong Normalization for CC^{Σ}).

$$\forall M \in \mathsf{Term}(\mathrm{CC}^{\Sigma})[SN(M)].$$

The version of Σ-types that makes CC inconsistent is the one that lets $\Sigma\alpha{:}A.\tau : \star$ if $A : \Box$ and $\tau : \star$. It is instructive to see why this version of Σ-types does not fit into the proof of SN above. Suppose we let $\Sigma\alpha{:}A.\tau : \star$, with $A : \Box$ and $\tau : \star$. Then we do not define $\mathcal{V}(\Sigma\alpha{:}A.\tau)$, because this is not a kind. Furthermore, we can define $\llbracket\Sigma\alpha{:}A.\tau\rrbracket_\xi$ as before. The problem arises when we try to define $\llbracket\pi_1 t\rrbracket_\xi$ for $t : \Sigma\alpha{:}A.\tau : \star$, because $\llbracket\pi_1 t\rrbracket_\xi$ can not be defined in terms of $\llbracket t\rrbracket_\xi$, for the simple reason that $\llbracket t\rrbracket_\xi$ does not exist. (Note that t is an object and for objects $\llbracket - \rrbracket_\xi$ is not defined.)

5.3. CC with inductive kinds

The approach to proving strong normalization can be generalised to inductive kinds. We treat the example for natural numbers. In the following, note that our 'inductive types' are *kinds*, whereas in a system like Coq, they are *types*. Having the natural numbers on the kind-level conforms better with a more traditional view on logical systems, where the level of 'domains' and the level of 'formulas' are separated. We now give the syntactic rules for the kind Nat.

$$(\text{Nat}) \vdash \text{Nat} : \Box, \qquad (\text{Zero}) \vdash Z : \text{Nat}, \qquad (\text{Succ}) \vdash S : \text{Nat}{\rightarrow}\text{Nat},$$

$$(\text{Elim})\ \frac{\Gamma, \alpha{:}\text{Nat} \vdash T : \star/\Box \quad \Gamma \vdash M_1 : T[Z/\alpha] \quad \Gamma \vdash M_2 : \Pi\alpha{:}\text{Nat}.T{\rightarrow}T[S\alpha/\alpha]}{\Gamma, \alpha{:}\text{Nat} \vdash \text{Rec}M_1M_2\alpha : T}$$

with the reduction rules

$$\text{Rec}M_1M_2Z \longrightarrow_r M_1, \quad \text{Rec}M_1M_2(S\alpha) \longrightarrow_r M_2\alpha(\text{Rec}M_1M_2\alpha)$$

The system CC extended with this scheme for natural numbers will be denoted by CC^N. The interpretation of CC^N in the saturated sets framework is as follows.

5.18. Definition. Adapt the mappings $\mathcal{V}$, $\llbracket - \rrbracket_\xi$ and $\llparenthesis - \rrparenthesis_\rho$ by adding the following clauses. (The interpretation of Π-kinds and Π-types is as in Definition 5.6.)

$$\begin{aligned}
\mathcal{V}(\mathrm{Nat}) &:= \mathbb{N},\\
\llbracket Z \rrbracket_\xi &:= 0,\\
\llbracket S \rrbracket_\xi &:= \boldsymbol{\lambda} n \in \mathbb{N}.n+1,\\
\llbracket \mathrm{Rec} P_1 P_2 \rrbracket_\xi &:= \text{the function defined by primitive recursion from } \llbracket P_1 \rrbracket_\xi \text{ and } \llbracket P_2 \rrbracket_\xi,\\
\llbracket \mathrm{Nat} \rrbracket_\xi &:= \mathrm{lfp}(\Phi), \text{ where for } N \in \mathbb{N} \rightarrow \mathrm{SAT},\\
\Phi(N) &= \boldsymbol{\lambda} n \in \mathbb{N}. \bigcap_{X \in \mathbb{N} \rightarrow \mathrm{SAT}} X(0) \rightarrow (\bigcap_{m \in \mathbb{N}} N(m) \rightarrow X(m) \rightarrow X(m+1)) \rightarrow X(n)),\\
\llparenthesis Z \rrparenthesis_\rho &:= \lambda xy.x,\\
\llparenthesis S \rrparenthesis_\rho &:= \lambda zxy.yz((\lambda v.vxy)z),\\
\llparenthesis \mathrm{Rec} M_1 M_2 \rrparenthesis_\rho &:= \lambda z.z \llparenthesis M_1 \rrparenthesis_\rho \llparenthesis M_2 \rrparenthesis_\rho .
\end{aligned}$$

We ambiguously denote $\llbracket \mathrm{Nat} \rrbracket_\xi$ by Nat.

The function Φ, used in the definition of $\llbracket \mathrm{Nat} \rrbracket_\xi$, is a positive morphism from $\mathbb{N} \rightarrow \mathrm{SAT}$ to $\mathbb{N} \rightarrow \mathrm{SAT}$ and hence it has a least fixed point (lfp). (Compare with Definition 4.4 and Corollary 4.6.) A term $\mathrm{Rec} M_1 M_2$ can be a constructor (if T in the scheme is a kind) or an object (if T in the scheme is a type). In the second case it only has an interpretation under $\llparenthesis - \rrparenthesis_\rho$, in the first case it has two interpretations. If $\mathrm{Rec} P_1 P_2 : \Pi\alpha{:}\mathrm{Nat}.T$, with T a kind, then $\llbracket \mathrm{Rec} P_1 P_2 \rrbracket_\xi$ is the function $F : \mathbb{N} \rightarrow \mathcal{V}(T)$, defined by $F(0) = \llbracket P_1 \rrbracket_\xi$ and $F(n+1) = \llbracket P_2 \rrbracket_\xi (n)(F(n))$.

5.19. Lemma (Soundness Lemma). *For Γ a context of CC^N, $P, A \in \mathsf{Term}(\mathrm{CC}^N)$ and $\xi \models^\Box \Gamma$,*

$$\begin{aligned}
\Gamma \vdash P : A (:\Box) &\Longrightarrow \llbracket P \rrbracket_\xi \in \mathcal{V}(A),\\
\Gamma \vdash A : \Box &\Longrightarrow \llbracket A \rrbracket_\xi \in \mathcal{V}(A) \rightarrow SAT.
\end{aligned}$$

Proof. By induction on the derivation. The only interesting case is when the last rule was (Elim) and $P \equiv \mathrm{Rec} P_1 P_2$. Then, by IH, $\llbracket P_1 \rrbracket_\xi \in \mathcal{V}(T)$ and $\llbracket P_2 \rrbracket_\xi \in \mathbb{N} \rightarrow \mathcal{V}(T) \rightarrow \mathcal{V}(T)$. So, indeed $\llbracket \mathrm{Rec} P_1 P_2 \rrbracket_\xi \in \mathbb{N} \rightarrow \mathcal{V}(T)$ and we are done. □

The notion of *the valuation ρ satisfies Γ with respect to ξ* ($\rho, \xi \models \Gamma$), is defined as before:

$$\begin{aligned}
x{:}\sigma \in \Gamma &\Longrightarrow \rho(x) \in \llbracket \sigma \rrbracket_\xi,\\
\alpha{:}A \in \Gamma &\Longrightarrow \rho(\alpha) \in \llbracket A \rrbracket_\xi (\xi(\alpha)).
\end{aligned}$$

So is the notion of Γ satisfies $M : T$ ($\Gamma \models M : T$), which is the same as in Definition 5.15.

5.20. THEOREM (Soundness Theorem). *For Γ a context and M and T terms of* CC^N,

$$\Gamma \vdash M : T \Longrightarrow \Gamma \models M : T.$$

PROOF. By induction on the derivation.

- $M \equiv Z$. Let $X \in \mathbb{N}{\to}\mathrm{SAT}$. For all $M \in X(0)$ and all $N \in \bigcap_{m \in \mathrm{N}} \mathrm{Nat}(m){\to}X(m){\to}X(m+1)$, $(\lambda xy.x)MN \overset{k}{\twoheadrightarrow} M \in X(0)$. $(\lambda xy.x)MN$ is also SN, so $(\lambda xy.x)MN \in X(0)$ and hence $\lambda xy.x \in \mathrm{Nat}(0)$.
- $M \equiv S$. We have to prove that $\llparenthesis S \rrparenthesis_\rho \in \bigcap_{p \in \mathrm{N}} \mathrm{Nat}(p){\to}\mathrm{Nat}(p+1)$. Let $p \in \mathbb{N}$ and $P \in \mathrm{Nat}(p)$. Let also $X \in \mathbb{N}{\to}\mathrm{SAT}$, $M \in X(0)$ and $N \in \bigcap_{m \in \mathrm{N}} \mathrm{Nat}(m){\to}X(m){\to}X(m+1)$. Then $NP \in X(p){\to}X(p+1)$ and $(\lambda v.vMN)P \in X(p)$, so $NP((\lambda v.vMN)P) \in X(p+1)$. Hence, $(\lambda zxy.yz((\lambda v.vxy)z))PMN \in X(p+1)$, and so $\lambda zxy.yz((\lambda v.vxy)z) \cap_{p \in \mathrm{N}} \mathrm{Nat}(p){\to}\mathrm{Nat}(p+1)$.
- $M \equiv \mathrm{Rec}M_1M_2$. We have to prove that

  $$\llparenthesis \mathrm{Rec}M_1M_2 \rrparenthesis_\rho \lambda z.z\llparenthesis M_1 \rrparenthesis_\rho M_2 \in \bigcap_{n \in \mathrm{N}} \mathrm{Nat}(n){\to}\llbracket T \rrbracket_{\xi(\alpha:=n)}(\llbracket \mathrm{Rec}M_1M_2 \rrbracket_\xi(n)).$$

  By IH, $\llparenthesis M_1 \rrparenthesis_\rho \in \llbracket T[Z/\alpha] \rrbracket_\xi(\llbracket M_1 \rrbracket_\xi)$ and

  $$\llparenthesis M_2 \rrparenthesis_\rho \in \bigcap_{n \in \mathrm{N}} \mathrm{Nat}(n){\to} \bigcap_{t \in \mathcal{V}(T)} \llbracket T \rrbracket_{\xi(\alpha:=n)}(t){\to}\llbracket T[S\alpha/\alpha] \rrbracket_{\xi(\alpha:=n)}(\llbracket M_2 \rrbracket_\xi(n)(t)).$$

  Let $n \in \mathbb{N}$ and $N \in \mathrm{Nat}(n)$. Take for X the map $\boldsymbol{\lambda} m \in \mathbb{N}.\llbracket T \rrbracket_{\xi(\alpha:=m)}(\llbracket \mathrm{Rec}M_1M_2 \rrbracket_\xi(m))$. Then $\llparenthesis M_1 \rrparenthesis_\rho \in X(0)$ and

  $$\llparenthesis M_2 \rrparenthesis_\rho \in \bigcap_{n \in \mathrm{N}} \mathrm{Nat}(n){\to}X(n){\to}X(n+1),$$

  by taking t to be $\llbracket \mathrm{Rec}M_1M_2 \rrbracket_\xi(n)$. Hence we find that $N\llparenthesis M_1 \rrparenthesis_\rho \llparenthesis M_2 \rrparenthesis_\rho \in X(n) = \llbracket T \rrbracket_{\xi(\alpha:=n)}(\llbracket \mathrm{Rec}M_1M_2 \rrbracket_\xi(n))$. So, $\llparenthesis \mathrm{Rec}M_1M_2 \rrparenthesis_\rho = \lambda z.z\llparenthesis M_1 \rrparenthesis_\rho \llparenthesis M_2 \rrparenthesis_\rho \in \bigcap_{n \in \mathrm{N}} \mathrm{Nat}(n){\to}\llbracket T \rrbracket_{\xi(\alpha:=n)}(\llbracket \mathrm{Rec}M_1M_2 \rrbracket_\xi(n))$. □

5.21. COROLLARY (Strong Normalization for CC^N).

$$\forall M \in \mathsf{Term}(\mathrm{CC}^N)[SN(M)].$$

The Corollary follows in a standard way from the Theorem (see the proof of Theorem 3.14) by taking for ρ the identity valuation ρ_0 and by observing that, if $M \longrightarrow_r N$, then $\llparenthesis M \rrparenthesis_{\rho_0}$ reduces to $\llparenthesis N \rrparenthesis_{\rho_0}$ in at least one step. For the latter: $\llparenthesis \mathrm{Rec}M_1M_2(Sx) \rrparenthesis_\rho \equiv (\lambda z.z\llparenthesis M_1 \rrparenthesis_\rho \llparenthesis M_2 \rrparenthesis_\rho)((\lambda zpq.qz((\lambda v.vpq)z))\rho(x))$, which reduces to $\llparenthesis M_2 \rrparenthesis_\rho \rho(x)((\lambda v.v\llparenthesis M_1 \rrparenthesis_\rho \llparenthesis M_2 \rrparenthesis_\rho)\rho(x)) \equiv \llparenthesis M_2(\mathrm{Rec}M_1M_2x) \rrparenthesis_\rho$.

Concluding Remarks

We have given a short and flexible proof of Strong Normalization for the Calculus of Constructions. The flexiblity lies in the fact that the framework of saturated sets allows many basic constructions like function types, product types and W-types. (One can also include, e.g. positive recursive types, for which details have been left out because of lack of space.) A question that has not been addressed here is whether this construction can be extended to higher universes (adding a sort $\Box_1$ with $\Box : \Box_1$, etcetera). It seems that, in order to treat this extension, one first has to prove a kind of quasi-normalization theorem (as in [Luo 1990], for the Extended Calculus of Constructions) to have some restriction on the possible form of a kind.

We did look into the extension with inductive types: the example of the natural numbers strongly suggests a general procedure for other inductive types by (roughly) interpreting an inductive type T as the parametrized saturated set that corresponds with the elimination scheme of T. Note however, that, different from a system like Coq, the inductive types are in fact kinds here (or 'large types'). Our treatment of inductive types as kinds fits rather naturally with the approach that we have chosen for the strong normalization proof, where the interpretation of a type does not depend on the interpretation of an object. However, it looks like this approach puts some principle restriction to the extendibility of our proof to the case where inductive types are small types. Then we can form a constructor P such that $P0$ is convertible with $\Pi\alpha{:}\star.\alpha$ and $P1$ is convertible with $\Pi\alpha{:}\star.\alpha\to\alpha$. Type dependency can not anymore be ignored in this case (because $[\![P0]\!]_\xi \neq [\![P1]\!]_\xi$). Furthermore, the interpretation of $\Pi x{:}\sigma.\tau$ can not be $[\![\sigma]\!]_\xi \to [\![\tau]\!]_\xi$; instead $\Pi x{:}\sigma.\tau$ should be interpreted in a parametrized way (as we did for inductive kinds in the last section) or as a real dependent product of saturated sets.

Acknowledgements

I would like to thank Gilles Barthe for the suggestion of defining the saturated sets as sets of pseudoterms (instead of untyped λ terms). I would like to thank the anonymous referees for their enlightening remarks and for pointing out several mistakes in the first version of this paper.

References

[Altenkirch 1993a] Th. Altenkirch, Yet another Strong Normalization proof for the Calculus of Constructions, Laboratory for Foundations of Computer Science, Manuscript, 11 pp.

[Altenkirch 1993] Th. Altenkirch, Constructions, Inductive types and Strong Normalization proof, Ph. D. Thesis, University of Edinburgh, UK.

[Barbanera et al. 1995] F. Barbanera, M. Fernández, J.H. Geuvers, Modularity of Strong Normalization in the lambda-algebraic-cube, manuscript.

[Barendregt 1984] H.P. Barendregt, *The lambda calculus: its syntax and semantics*, revised edition. Studies in Logic and the Foundations of Mathematics, North

Holland.

[Barendregt 1992] H.P. Barendregt, Typed lambda calculi. In Abramski et al. (eds.), *Handbook of Logic in Computer Science*, Oxford Univ. Press.

[Berardi 1988] S. Berardi, Towards a mathematical analysis of the Coquand-Huet calculus of constructions and the other systems in Barendregt's cube. Dept. Computer Science, Carnegie-Mellon University and Dipartimento Matematica, Universita di Torino, Italy.

[Coquand 1985] Th. Coquand, Une théorie des constructions, Thèse de troisième cycle, Université Paris VII, France.

[Coquand 1990] Th. Coquand, Metamathematical investigations of a calculus of constructions. In *Logic and Computer Science*, ed. P.G. Odifreddi, APIC series, vol. 31, Academic Press, pp 91-122.

[Coquand and Gallier 1990] Th. Coquand and J. Gallier, A proof of Strong Normalization for the Theory of Constructions using a Kripke-like interpretation, In the Informal Proceedings of the Workshop on Logical Frameworks, Antibes, May 1990.

[Coquand and Huet 1988] Th. Coquand and G. Huet, The calculus of constructions, *Information and Computation*, 76, pp 95-120.

[Coquand and Mohring 1990] Th. Coquand and Ch. Paulin-Mohring Inductively defined types, In P. Martin-Löf and G. Mints editors. *COLOG-88 : International conference on computer logic, LNCS 417.*

[Geuvers and Nederhof 1991] J.H. Geuvers and M.J. Nederhof, A modular proof of strong normalisation for the calculus of constructions. *Journal of Functional Programming*, vol 1 (2), pp 155-189.

[Geuvers 1993] J.H. Geuvers, Logics and Type Systems, Ph. D. thesis, Universiteit Nijmegen, the Netherlands.

[Geuvers and Werner 1994] H. Geuvers and B. Werner, On the Church-Rosser property for Expressive Type Systems and its Consequences for their Metatheoretic Study, in *Proceedings of the Ninth Annual Symposium on Logic in Computer Science, Paris, France,* IEEE Computer Society, pp 320–329.

[Gallier 1990] On Girard's "Candidats de Reductibilité". In *Logic and Computer Science*, ed. P.G. Odifreddi, APIC series, vol. 31, Academic Press, pp 123-204.

[Girard 1972] J.-Y. Girard, Interprétation fonctionelle et élimination des coupures dans l'arithmétique d'ordre supérieur. Ph.D. thesis, Université Paris VII, France.

[Girard et al. 1989] J.-Y. Girard, Y. Lafont and P. Taylor, *Proofs and types*, Camb. Tracts in Theoretical Computer Science 7, Cambridge University Press.

[Goguen 1994] H. Goguen, A Typed Operational Semantics for Type Theory, PhD. thesis, University of Edinburgh, UK, 1994.

[Luo 1990] Z. Luo, An Extended Calculus of Constructions, Ph. D. Thesis, University of Edinburgh, UK.

[Luo 1989] Z. Luo, ECC: An extended Calculus of Constructions. *Proc. of the fourth ann. symp. on Logic in Comp. Science, Asilomar, Cal.* IEEE, pp 386-395.

[Martin-Löf 1984] P. Martin-Löf, *Intuitionistic Type Theory*, Studies in Proof theory, Bibliopolis, Napoli.

[Nordström et al. 1990] B. Nordström, K. Petersson, J.M. Smith, *Programming in Martin-Löf's Type Theory.* Oxford University Press.

[Ong and Ritter 1994] L. Ong and E. Ritter, A generic Strong Normalization argument: application to the Calculus of Constructions, University of Cambridge Computer Laboratory, Manuscript, 19 pp.

[Scedrov 1990] A guide to polymorphic types. In *Logic and Computer Science*, ed. P.G. Odifreddi, APIC series, vol. 31, Academic Press, pp 387-420.

[Tait 1965] W.W. Tait, Infinitely long terms of transfinite type. In *Formal Systems and Recursive Functions*, eds. J.N. Crossley and M.A.E. Dummett, North-Holland.

[Tait 1975] W.W. Tait, A realizability interpretation of the theory of species. In *Proceedings of Logic Colloquium*, ed. R. Parikh, LNM 453, pp 240–251.

[Terlouw 1993] J. Terlouw, Strong Normalization in type systems: a model theoretic approach, In the *Dirk van Dalen Festschrift,* Eds. H. Barendregt, M. Bezem and J.W. Klop, Department of Philosophy, Utrecht University, the Netherlands, pp 161-190.