Journal of Functional Programming 1 (2): 155–189, April 1991

Modular proof of strong normalization for the calculus of constructions

HERMAN GEUVERS* AND MARK-JAN NEDERHOF
Faculty of Mathematics and Computer Science, University of Nijmegen, The Netherlands

Abstract

We present a modular proof of strong normalization for the Calculus of Constructions of Coquand and Huet (1985, 1988). This result was first proved by Coquand (1986), but our proof is more perspicious. The method consists of a little juggling with some systems in the cube of Barendregt (1989), which provides a fine structure of the calculus of constructions. It is proved that the strong normalization of the calculus of constructions is equivalent with the strong normalization of $F\omega$.

In order to give the proof, we first establish some properties of various type systems. Therefore, we present a general framework of typed lambda calculi, including many well-known ones.

Capsule review

The calculus of constructions due to Coquand and Huet (1985, 1988) is a very popular subject among those interested in computer science-oriented aspects of intuitionistic type theory. The strong normalization theorem for it, stating that all computation sequences terminate, is one of the most basic results. The authors present a new proof of strong normalization obtained essentially by combining ideas from several previous proofs of similar results for different systems, allowing them to divide up the difficulties and cope with them one at a time. The paper is self-contained and all proofs are given in detail.

1 Introduction

The strong normalization (SN) property for the calculus of constructions (Coquand and Huet, 1985, 1988) was proved by Coquand (1986). This proof is rather 'baroque', although it is reminiscent of other proofs of strong normalization for systems like the simply or polymorphically typed lambda calculus. Therefore, we looked for a conceptually more perspicious proof.

Barendregt (1989) gives a fine structure of the calculus of constructions. He defines a natural cube of eight type systems, ordered along the edges by inclusion, of which the smallest system is the simply typed lambda calculus $\lambda\!\rightarrow$, and the most complicated system is the calculus of constructions λC. Other systems in Barendregt's

* Partially supported by the EEC 'Project Stimulation ST2J/0374/C(EDB): Lambda Calcul Typé'.

cube include the second order (polymorphic) lambda calculus λ2, Girard's Fω system (called λω in the cube) and λP, a system related to the AUTOMATH system AUT-QE and studied by Harper *et al.* (1987) under the name LF.

The first step in our proof of strong normalization consists of defining a mapping from λC into λω such that reduction of terms is preserved. (This generalizes a method of Harper *et al.* (1987) mapping λP into λ→). Then, motivated by three kinds of abstractions possible in the various parts of the cube, reductions in λω are divided into three kinds. Two of these reductions turn out to be strongly normalizing. (For one of them we need that λ→ is SN.) To show that the third notion of reduction (even if mixed with the other two) is also SN, we map the terms of λω into the set Λ of typefree lambda terms. Then a higher order version of the argument used for λ2 (by Girard (1972), or Tait (1975)) shows that λω is SN.

The lemmas that are used in the proof of strong normalization are valid for a large class of systems not included in the cube. Therefore, we start this paper by giving a more general notion of type system, following definitions by Terlouw (1989*a*) and Berardi (1988), and prove the basic lemmas for this general notion. This puts the properties of the systems of the cube in a larger framework that might be useful for other arguments about typed lambda calculi. Berardi has shown that various logical systems also fit in the general notion of type system (see Barendregt, 1989, for some examples), which again stresses the relationship between typed lambda calculus and logic, usually expressed by the 'formulae-as-types' notion, discussed by Howard (1980).

2 Framework for type systems

A general notion of type system is presented, following definitions given by Terlouw (1989*a*, *b*) and Berardi (1988), generalizing Barendregt's cube. This notion will serve as a framework for reasoning about type systems and comparing various type systems with each other. One can prove a number of basic properties for this notion, which show how things work. We restrict ourselves to those properties that will be useful in the argument about strong normalization.

Definition 1
A Generalized Type System (*GTS*) is a system consisting of the following objects

(i) $\mathcal{C}$, a set of constants,
(ii) $\mathcal{S} \subseteq \mathcal{C}$, a set of sorts,
(iii) $\mathcal{A}$, a set of pairs (c, c'), with $c, c' \in \mathcal{C}$,
(iv) $\mathcal{R}$, a set of triples (s, s', s''), with $s, s', s'' \in \mathcal{S}$,
(v) for every $s \in \mathcal{S}$, a set $\mathcal{V}^s$ of variables.

The axiom pairs (c, c') will usually be denoted by c:c'. If $s' \equiv s''$ in a rule (s, s', s''), then the rule (s, s', s'') will be written as (s, s'). A GTS will be denoted by the quadruple $(\mathcal{C}, \mathcal{S}, \mathcal{A}, \mathcal{R})$. For the set of all variables, we shall write $\mathcal{V}$, so $\mathcal{V} = \bigcup_{s \in \mathcal{S}} \mathcal{V}^s$.

The idea of the GTS definition is that the sorts are the universes of the type system, where the axioms give the hierarchical structure between them. The constants are special objects of the type system, belonging to a universe or another constant. The rules restrict the formation of the Π-type, the collection of (possibly dependent) functions from one type to another. In Chapter 3 it can be seen how the GTS definition works in the case of well-known systems such as simply and polymorphically typed lambda calculus and the calculus of constructions.

Definition 2

(i) A GTS (*CONS*, *SORT*, *AXIOM*, *RULE*) is *functional* iff

- $AXIOM \subseteq CONS \times CONS$ is a *function*, i.e., $\forall c, c', c'' \in CONS.$ $[c:c', c:c'' \in AXIOM \Rightarrow c' \equiv c'']$,
- $RULE \subseteq (SORT \times SORT) \times SORT$ is a *function*, i.e., $\forall s_1, s_2, s_3, s_3' \in SORT.$ $[(s_1, s_2, s_3), (s_1, s_2, s_3') \in RULE \Rightarrow s_3 \equiv s_3']$,

(ii) A GTS (*SORT*, *CONS*, *AXIOM*, *RULE*) is *injective* iff

- *AXIOM* is *injective* on $SORT \times SORT$, i.e., $\forall s, s' \in SORT, c \in CONS.$ $[s:c, s':c \in AXIOM \Rightarrow s \equiv s']$,
- *RULE* is *injective in its second argument*, i.e., $\forall s_1, s_2, s_2', s_3 \in SORT.$ $[(s_1, s_2, s_3), (s_1, s_2', s_3) \in RULE \Rightarrow s_2 \equiv s_2']$.

A motivation for these two definitions is that type systems which are functional will turn out to have the so called 'uniqueness of assignment' property; the type that can be assigned to a certain object is unique up to β-equality. (Even without knowing the inference rules for GTSs, it will be clear that if *AXIOM* is not injective, a constant might be typed with two different sorts which are not β-equal.) Further, the properties of functionality and injectivity give rise to nice classifications of objects in a type system. It is not worth explaining the interest of these two definitions at this point. They will only become clear in view of the specific properties for functional and injective GTSs that are given at the end of this chapter.

Let in the following $\zeta = (SORT, CONS, AXIOM, RULE)$ be a GTS.

Definition 3

(i) The collection of *pseudoterms of* ζ, $PST(\zeta)$, is defined by

$$t ::= VAR \mid CONS \mid (t\,t) \mid (\lambda VAR{:}t.\,t) \mid (\Pi VAR{:}t.\,t).$$

(ii) If a term is one of the last three forms, it will be called a *composed* term.

The notions of *bound variable* and *free variable* of a pseudoterm are defined as usual, λ and Π bind variables.

The *substitution operator* works as in the untyped lambda calculus; $t[x := t']$ denotes the substitution of t' for x in the term t. Substitution is only allowed if no free variables become bound.

Just as in untyped lambda calculus, terms that only differ from each other in their bound variables will be identified; we work *modulo α-conversion*. If the terms t and

t' are α-convertible, this is denoted by $t \equiv t'$. In general, it will be assumed that the bound variables in a term differ from the free ones.

Also, the notion of *subterm* is directly copied from the untyped case. (For precise definitions, we refer to Barendregt, 1984).

Definition 4

(i) a *redex* is a pseudoterm of the form $(\lambda x{:}t.t')\,t''$.

(ii) *one-step* β*-reduction*, $\to_\beta$ (or just $\to$), is defined by $(\lambda x{:}t.t')\,t'' \to_\beta t'[x := t'']$, and if $t' \to_\beta t''$, then $t\,t' \to_\beta t\,t''$, $t'\,t \to_\beta t''t$, $\lambda x{:}t.t' \to_\beta \lambda x{:}t.t''$, $\lambda x{:}t'.t \to_\beta \lambda x{:}t''.t$, $\Pi x{:}t.t' \to_\beta \Pi x{:}t.t''$ and $\Pi x{:}t'.t \to_\beta \Pi x{:}t''.t$, for all $t, t', t'' \in \mathrm{PST}(\zeta)$.

(iii) β*-reduction*, $\twoheadrightarrow$, is the transitive reflexive closure of $\to$.

(iv) β*-conversion*, $=$, is the least equivalence relation generated by $\twoheadrightarrow$.

Theorem 5

The reduction relation $\twoheadrightarrow$ *on* $PST(\zeta)$ *satisfies the* Church–Rosser property, *i.e.* $\forall M, N \in PST(\zeta).\ [M = N \Rightarrow \exists P \in PST(\zeta).\ M \twoheadrightarrow P\ \&\ N \twoheadrightarrow P]$.

Proof. The proof of the Church–Rosser property for pseudoterms can be given in the same way as the proof of Church–Rosser for the untyped lambda calculus. We do not give it here. (For details see Barendregt and Dekkers, 1990). □

Definition 6

(i) A ζ*-assignment* is an expression of the form $A{:}B$, with $A, B \in \mathrm{PST}(\zeta)$,

(ii) A ζ*-declaration* is an expression of the form $x{:}A$, with $A \in \mathrm{PST}(\zeta)$, $x \in V_{AR}$,

(iii) A ζ*-pseudocontext* is a finite sequence of ζ-declarations,

(iv) A ζ*-statement* is an expression of the form $\Gamma \vdash A{:}B$, with Γ a ζ-pseudocontext, $A{:}B$ a ζ-assignment.

Definition 7

Let Γ' and $\Gamma = x_1{:}A_1, \ldots, x_n{:}A_n$ be pseudocontexts.

(i) The *domain of* Γ, *dom*(Γ), is the set $\{x_1, x_2, \ldots, x_n\}$,

(ii) For $i \leqslant n$, $\Gamma \lceil i$ (Γ *restricted to* i) is the pseudocontext $x_1{:}A_1, \ldots, x_i{:}A_i$,

(iii) For $i \leqslant n$, $\Gamma \backslash (x_i{:}A_i) = x_1{:}A_1, \ldots, x_{i-1}{:}A_{i-1}, x_{i+1}{:}A_{i+1}, \ldots, x_n{:}A_n$,

(iv) $\Gamma' \leqslant \Gamma$ (Γ' *is prefix of* Γ) iff $\Gamma' = \Gamma \lceil i$ for some $i \leqslant n$,

(v) $\Gamma' \subseteq \Gamma$ (Γ' *is subcontext of* Γ) iff $x{:}A$ in $\Gamma' \Rightarrow x{:}A$ in Γ,

(vi) $\Gamma \twoheadrightarrow \Gamma'$ iff $\Gamma' = x_1{:}A_1', \ldots, x_n{:}A_n'$ and $A_i \twoheadrightarrow A_i'$ for all $i \leqslant n$.

We now define for a pseudocontext Γ and pseudoterms A and B, the notion '$\Gamma \vdash A{:}B$ is true'. This definition picks the 'legal' terms out of the set of pseudoterms and the 'legal' contexts out of the set of pseudocontexts. Instead of '$\Gamma \vdash A{:}B$ is true', we shall just write $\Gamma \vdash A{:}B$, in words Γ *proves* $A{:}B$, or Γ *assigns* B *to* A. The notion of $\Gamma \vdash A{:}B$ is generated by the axioms, and inference rules of the system ζ. In case the system ζ in which we are working is not clear from the context, we write $\Gamma \vdash_\zeta A{:}B$.

Definition 8

(i) The *axioms of* ζ are the statements $\vdash c:c'$, with $c:c' \in \mathit{AXIOM}$,

(ii) The *inference rules of* ζ are the rules of the following six forms

$$\text{(Start)}\ \frac{\Gamma \vdash A:s}{\Gamma, x:A \vdash x:A} \qquad \text{with } s \in \mathit{SORT},\ x \in \mathit{VAR}^s,\ x \notin \mathrm{dom}(\Gamma),$$

$$\text{(Weakening)}\ \frac{\Gamma \vdash A:s \quad \Gamma \vdash B:C}{\Gamma, x:A \vdash B:C} \qquad \text{with } s \in \mathit{SORT},\ x \in \mathit{VAR}^s,\ x \notin \mathrm{dom}(\Gamma),$$

$$(\Pi\text{-rule})\ \frac{\Gamma \vdash B_1:s_1 \quad \Gamma, x:B_1 \vdash B_2:s_2}{\Gamma \vdash \Pi x:B_1.B_2:s_3} \qquad \text{with } (s_1, s_2, s_3) \in \mathit{RULE},$$

$$(\lambda\text{-rule})\ \frac{\Gamma \vdash B_1:s_1 \quad \Gamma, x:B_1 \vdash B_2:s_2 \quad \Gamma, x:B_1 \vdash C:B_2}{\Gamma \vdash \lambda x:B_1.C:\Pi x:B_1.B_2}$$

with $(s_1, s_2, s_3) \in \mathit{RULE}$ for some $s_3 \in \mathit{SORT}$,

$$\text{(Application)}\ \frac{\Gamma \vdash B_1:\Pi x:C_1.C_2 \quad \Gamma \vdash B_2:C_1}{\Gamma \vdash B_1 B_2:C_2[x:=B_2]},$$

$$\text{(Conversion)}\ \frac{\Gamma \vdash B:C \quad \Gamma \vdash C':s}{\Gamma \vdash B:C'} \qquad \text{with } C = C',\ s \in \mathit{SORT}.$$

The statements above the line will be called *premises*, the statements below the line the *conclusions* of a rule.

Definition 9

A *derivation* in the system ζ is a finite well-founded tree with

(i) each leaf of the tree is an axiom of ζ,

(ii) each node of the tree which is not a leaf is a conclusion of an inference rule, such that the successors of the node are exactly the premises of the inference rule.

For Δ and Δ' derivations in ζ, $\Delta < \Delta'$ (Δ subderivation of Δ') is defined as usual.

Definition 10

(I) For a derivation Δ, the *length of* Δ, $lh(\Delta)$, is inductively defined by

(i) If Δ consists only of an axiom, then $lh(\Delta) = 0$,

(ii) If the premises of the conclusion of Δ are $F_1, \ldots, F_n$, with derivations $\Delta_1, \ldots, \Delta_n$, then $lh(\Delta) = max\{lh(\Delta_i) \mid 1 \leqslant i \leqslant n\} + 1$.

(II) For a derivation Δ, the *trace of* Δ is the path in Δ that

(i) starts with the root and ends with a leaf,

(ii) takes the left path in case of (application) or (conversion),

(iii) takes the right path in case of (weakening), (Π-rule) or (λ-rule).

Definition 11
Let Γ be a pseudocontext, $A, B \in \mathrm{PST}(\zeta)$, Δ a derivation in ζ.

(i) Δ *is a derivation of* $\Gamma \vdash A:B$ or $\Delta:(\Gamma \vdash A:B)$ iff $\Gamma \vdash A:B$ is the root of the derivation Δ,
(ii) $\Gamma \vdash A:B$ *is true* or $\Gamma \vdash A:B$ iff $\Delta:(\Gamma \vdash A:B)$ for some derivation Δ in ζ.

Notation 12. $\Gamma \vdash A:B:C$ iff $\Gamma \vdash A:B$ and $\Gamma \vdash B:C$.

Definition 13
Let $s \in \mathcal{SORT}$.

(i) $\mathit{Context}(\zeta) = \{\Gamma \mid \Gamma \vdash A:B \text{ for some } A, B \in \mathrm{PST}(\zeta)\}$,
(ii) $\Gamma\text{-}\mathit{Term}(\zeta) = \{A \mid \Gamma \vdash A:B \text{ or } \Gamma \vdash B:A \text{ for some } B \in \mathrm{PST}(\zeta)\}$,
(iii) $\Gamma\text{-}s\text{-}\mathit{Term}(\zeta) = \{A \mid \Gamma \vdash A:s\}$,
(iv) $\Gamma\text{-}s\text{-}\mathit{Elt}(\zeta) = \{A \mid \Gamma \vdash A:B:s \text{ for some } B \in \mathrm{PST}(\zeta)\}$,
(v) $\mathit{Term}(\zeta) = \bigcup_{\Gamma \in \mathit{Context}(\zeta)} \Gamma\text{-}\mathit{Term}(\zeta)$,
(vi) $s\text{-}\mathit{Term}(\zeta) = \bigcup_{\Gamma \in \mathit{Context}(\zeta)} \Gamma\text{-}s\text{-}\mathit{Term}(\zeta)$,
(vii) $s\text{-}\mathit{Elt}(\zeta) = \bigcup_{\Gamma \in \mathit{Context}(\zeta)} \Gamma\text{-}s\text{-}\mathit{Elt}(\zeta)$.

Definition 14
Let ζ be a GTS, $M \in \mathit{Term}(\zeta)$, $n \in \mathbb{N}$.

(i) n *is an upperbound to the reductions starting from* M iff

$$\forall M_1, M_2, \ldots, M_m \in \mathit{Term}(\zeta).\, [M \to M_1 \to \ldots \to M_{m-1} \to M_m \Rightarrow m \leqslant n],$$

(ii) M *is strongly normalizable*, or $\mathrm{SN}(M)$, iff $\exists n \in \mathbb{N}.\, [n$ is an upperbound to the reductions starting from $M]$,
(iii) ζ *satisfies the strong normalization property*, or $\zeta \models \mathrm{SN}$, iff $\forall M \in \mathit{Term}(\zeta).\, \mathrm{SN}(M)$.

The fact that all terms of a set X are strongly normalizable will be denoted by $\mathrm{SN}(X)$.

In the following, the double negated version of $\zeta \models \mathrm{SN}$ will sometimes be used: $\neg\exists M_1 \in \mathit{Term}(\zeta).\, \forall m \in \mathbb{N}.\, \exists M_2, M_3, \ldots, M_m \in \mathit{Term}(\zeta).\, \forall i \leqslant m-1.\, [M_i \to M_{i+1}]$, stating that there are no infinite reduction sequences in ζ. The proofs given below are therefore not all constructive. Analysing the proofs, one can see, however, that the proof of equivalence of $\lambda\omega \models \mathrm{SN}$ and $\lambda P\omega \models \mathrm{SN}$ can be done in first order Heyting arithmetic.

From the axioms and inference rules, it will be clear that if a constant does not occur in any of the axioms or on the third place of a rule, then it does not occur in any statement of ζ. In the following, we shall therefore assume that

$$c \in \mathcal{CONS} \Rightarrow \exists c'.\, c:c' \in \mathcal{AXIOM} \vee c':c \in \mathcal{AXIOM}$$
$$\vee\, \exists s, s'.\, (s, s', c) \in \mathcal{RULE}.$$

The rest of this chapter will consist of lemmas and proofs for the generalized type systems. For examples we refer to Chapter 3, in which the systems of Barendregt's

cube are defined as GTSs. Some more exotic examples can be found in Barendregt (1989), e.g., the definition of predicate logics as GTSs. Barendregt also shows why, in general, we want the set *RULE* to consist of triples (s_1, s_2, s_3), and not just pairs (s_1, s_2).

Lemma 15 (*Free variables*)
For $\Gamma = x_1:A_1, x_2:A_2, \ldots, x_n:A_n$ *and* $\Gamma \vdash B:C$,

(i) $FV(B:C) \subseteq \{x_1, \ldots, x_n\}$,
(ii) $\forall i,j \leqslant n . x_i \equiv x_j \Rightarrow i = j$.

Proof. Induction on the length of the derivation of $\Gamma \vdash B:C$, distinguishing cases according to the last applied inference rule. □

Lemma 16
For $\Gamma = x_1:A_1, x_2:A_2, \ldots, x_n:A_n \in Context(\zeta)$,

(i) $\Gamma \vdash c:c'$ *for all* $c:c' \in AXIOM$,
(ii) $\Gamma \vdash x_i:A_i$ *for all* $i \leqslant n$.

Proof. The proof of (i) is by easy induction on the length of the tree that proves $\Gamma \in Context(\zeta)$.

For the proof of (ii), let $\Delta:(\Gamma \vdash B:C)$. The proof of $\Gamma \vdash x_i:A_i$ for all $i \leqslant n$ is by induction on the length of Δ, distinguishing cases according to the last applied rule. The two interesting cases are when this is (start) or (weakening). We only treat the case for the last rule being (weakening), as the other case is quite similar.

If the last rule is (weakening), then

$$\frac{x_1:A_1, \ldots, x_{n-1}:A_{n-1} \vdash A_n:s \quad x_1:A_1, \ldots, x_{n-1}:A_{n-1} \vdash B:C}{x_1:A_1, \ldots, x_n:A_n \vdash B:C}.$$

With one application of (start) we find that $\Gamma \vdash x_n:A_n$. By induction hypothesis $\Gamma\lceil n-1 \vdash x_i:A_i$ for all $i \leqslant n-1$, so with one application of (weakening): $\Gamma \vdash x_i:A_i$ for all $i \leqslant n-1$. □

Lemma 17 (*Substitution*)
For Γ_1 *and* $\Gamma_1, y:A, \Gamma_2 \in Context(\zeta)$, $A, B, C, D \in Term(\zeta)$, $y \in VAR$

$$\Gamma_1, y:A, \Gamma_2 \vdash B:C \;\&\; \Gamma_1 \vdash D:A \Rightarrow \Gamma_1, \Gamma_2[y:=D] \vdash B[y:=D]:C[y:=D].$$

Proof. By induction on the length of the derivation of $\Gamma_1, y:A, \Gamma_2 \vdash B:C$, assuming that $\Gamma_1 \vdash D:A$. We distinguish cases according to the last applied inference rule. If this rule is (start) or (weakening), we distinguish subcases $\Gamma_2 = \varnothing$ and $\Gamma_2 \neq \varnothing$.

If the last rule is (Π-rule), (λ-rule) or (conversion), or $\Gamma_2 \neq \varnothing$ and the last rule is (start) or (weakening), then the statement follows immediately from the induction hypothesis and an application of the rule.

If the last rule is (start) and $\Gamma_2 = \varnothing$, then $B \equiv y$ and $C \equiv A$. Now $y[y:=D] \equiv D$ and $A[y:=D] \equiv A$, so we are done by the assumption $\Gamma_1 \vdash D:A$.

If the last rule is (weakening) and $\Gamma_2 = \varnothing$, then

$$\frac{\Gamma_1 \vdash A:s \quad \Gamma_1 \vdash B:C}{\Gamma_1, y:A \vdash B:C}.$$

Now $y \notin FV(B:C)$, so $B[y:=D] \equiv B$ and $C[y:=D] \equiv C$, and we are done.

If the last rule is (application), then $B \equiv B_1 B_2$, $C \equiv C_2[x:=B_2]$ and

$$\frac{\Gamma_1, y:A, \Gamma_2 \vdash B_1:\Pi x:C_1.C_2 \quad \Gamma_1, y:A, \Gamma_2 \vdash B_2:C_1}{\Gamma_1, y:A, \Gamma_2 \vdash B_1 B_2:C_2[x:=B_2]}.$$

Now by induction hypothesis and (application)

$$\Gamma_1, \Gamma_2[y:=D] \vdash B_1 B_2[y:=D]:C_2[y:=D][x:=B_2[y:=D]].$$

We may assume that $x \notin FV(\Gamma_1, y:A, \Gamma_2)$, so $x \neq y$ and $x \notin FV(D)$. It follows that $C_2[y:=D][x:=B_2[y:=D]] \equiv C_2[x:=B_2][y:=D]$, and we are done. □

Lemma 18 (*Thinning*)
For $\Gamma, \Gamma' \in Context(\zeta)$, $B, C \in Term(\zeta)$

$$\Gamma \vdash B:C \,\&\, \Gamma \subseteq \Gamma' \Rightarrow \Gamma' \vdash B:C.$$

Proof. By induction on the length of the derivation of $\Gamma \vdash B:C$, distinguishing cases according to the last applied rule.

If $\Gamma \vdash B:C$ is an axiom or the last rule was (start), we are done by lemma 2.16. If the last rule is (weakening), we are done by the induction hypothesis. If the last rule is (application) or (conversion), the statement follows from the induction hypothesis and an application of the rule.

The argument for the cases when the last rule is (Π-rule) or (λ-rule) is similar. We treat the case for (Π-rule).

Let $\Gamma' \supseteq \Gamma$, $B \equiv \Pi x:B_1.B_2$, $C \equiv s$, and

$$\frac{\Gamma \vdash B_1:s_1 \quad \Gamma, x:B_1 \vdash B_2:s_2}{\Gamma \vdash \Pi x:B_1.B_2:s}.$$

Then we may assume that $x \notin \mathrm{dom}(\Gamma')$. By induction hypothesis $\Gamma' \vdash B_1:s_1$, so $\Gamma', x:B_1 \in Context(\zeta)$. $\Gamma', x:B_1 \supseteq \Gamma, x:B_1$, so by induction hypothesis $\Gamma', x:B_1 \vdash B_2:s_2$. With one application of (Π-rule) $\Gamma' \vdash \Pi x:B_1.B_2:s$. □

Lemma 19 (*Stripping*)
For $\Gamma = x_1:A_1, \ldots, x_n:A_n \in Context(\zeta)$, $M, N, P, R \in Term(\zeta)$

(i) $\Gamma \vdash c:R$, *with* $c \in \textsc{Cons} \Rightarrow \exists c' \in \textsc{Cons}.[R = c' \,\&\, c:c' \in \textsc{Axiom}]$,

(ii) $\Gamma \vdash x:R$, *with* $x \in \textsc{Var} \Rightarrow \exists i \leqslant n \exists s \in \textsc{Sort}.[\; x \equiv x_i \in \textsc{Var}^s \,\&\, R = A_i \,\&\, \Gamma\lceil i-1 \vdash A_i:s]$,

(iii) $\Gamma \vdash \Pi x{:}M.N{:}R \Rightarrow \Gamma \vdash M{:}s_1$ &
$\Gamma, x{:}M \vdash N{:}s_2$ &
$R = s_3$,
for some $(s_1, s_2, s_3) \in \mathcal{RULE}$,

(iv) $\Gamma \vdash \lambda x{:}M.N{:}R \Rightarrow \Gamma \vdash M{:}s_1$ &
$\Gamma, x{:}M \vdash B{:}s_2$ &
$\Gamma, x{:}M \vdash N{:}B$ &
$\Gamma \vdash \Pi x{:}M.B{:}s_3$ &
$R = \Pi x{:}M.B$,
for some $(s_1, s_2, s_3) \in \mathcal{RULE}$, $B \in \mathcal{T}erm(\zeta)$,

(v) $\Gamma \vdash MN{:}R \Rightarrow \Gamma \vdash M{:}\Pi x{:}A.B$ &
$\Gamma \vdash N{:}A$ &
$R = B[x := N]$,
for some $A, B \in \mathcal{T}erm(\zeta)$, $x \in \mathcal{VAR}$,

(vi) $\Gamma \vdash P{:}R \Rightarrow \exists c \in \mathcal{CONS}.[R \equiv c \vee \Gamma \vdash R{:}c \,\&\, c \in \mathcal{SORT}]$.

Proof (*i*)–(*v*). Let $\Delta{:}(\Gamma \vdash P{:}R)$ in one of the first five cases above. When we follow the trace of Δ, we only pass applications of (weakening) and (conversion), which do not change the term P, until we hit upon a rule by which the term P is introduced. In case (i) this is an axiom, in case (ii) this is (start), in case (iii) the (Π-rule), in case (iv) the (λ-rule), and in case (v) the (application). In all cases, the conclusion of the rule is $\Gamma' \vdash P{:}R'$, with $\Gamma' \leqslant \Gamma$ and $R' = R$. The proof of the five cases above now follows immediately by taking a look at the rule by which P is introduced, thinning the context Γ' to Γ and converting R' to R.

Proof (*vi*). By induction on the structure of P. Following the trace up in the tree $\Delta{:}(\Gamma \vdash P{:}R)$, we only pass applications of (weakening) until we hit upon (conversion), or the rule by which P is introduced. In the first case we conclude that $\Gamma \vdash R{:}s$, for some $s \in \mathcal{SORT}$, and we are done. In the second case we distinguish subcases according to the structure of P (cases (i)–(v) above).

If $P \equiv c \in \mathcal{CONS}$, $P \equiv x \in \mathcal{VAR}$, $P \equiv \Pi x{:}M.N$ or $P \equiv \lambda x{:}M.N$, it is immediately clear that $R \equiv c'$ for some $c' \in \mathcal{CONS}$ or $\Gamma \vdash R{:}s$ for some $s \in \mathcal{SORT}$.

If $P \equiv MN$, then $R \equiv C_2[x := N]$, $\Gamma \vdash M{:}\Pi x{:}C_1.C_2$ and $\Gamma \vdash N{:}C_1$ for certain C_1, C_2. Applying the induction hypothesis to $\Gamma \vdash M{:}\Pi x{:}C_1.C_2$ and case (iii), we find that $\Gamma, x{:}C_1 \vdash C_2{:}s_2$. With the substitution lemma we obtain $\Gamma \vdash C_2[x := N]{:}s_2$. □

Lemma 20 (*Permutation*)
For Γ_1 *and* $\Gamma_1, x{:}A, y{:}D, \Gamma_2 \in \mathcal{C}ontext(\zeta)$, $B, C \in \mathcal{T}erm(\zeta)$

$$\Gamma_1, x{:}A, y{:}D, \Gamma_2 \vdash B{:}C \,\&\, \Gamma_1 \vdash D{:}s \Rightarrow \Gamma_1, y{:}D, x{:}A, \Gamma_2 \vdash B{:}C.$$

Proof. Remark that $\Gamma_1, y{:}D$ is a legal context and so, by $\Gamma_1 \vdash A{:}s'$ and thinning, $\Gamma_1, y{:}D, x{:}A$ is a legal context too. If $\Gamma_2 = z_1{:}E_1, \ldots, z_n{:}E_n$, we may conclude that $\Gamma_1, y{:}D, x{:}A \vdash E_1{:}s''$ for some sort s'', and so $\Gamma_1, y{:}D, x{:}A, z_1{:}E_1$ is legal. Proceeding in this way for all $i \leqslant n$, we find that $\Gamma_1, y{:}D, x{:}A, \Gamma_2$ is a legal context. Using thinning one concludes that $\Gamma_1, y{:}D, x{:}A, \Gamma_2 \vdash B{:}C$. □

Lemma 21 (*Terms*)

$$M \in Term(\zeta) \Leftrightarrow M \in CONS \vee \exists \Gamma \exists C.\, \Gamma \vdash M : C.$$

Proof. According to the definition of $Term(\zeta)$,

$$M \in Term(\zeta) \Leftrightarrow \exists \Gamma \exists C.\, [\Gamma \vdash M : C \vee \Gamma \vdash C : M].$$

If $\Gamma \vdash C : M$, we find with the stripping lemma 19 (vi) that $M \equiv c \in CONS$ or $\Gamma \vdash M : s$ with $s \in SORT \subseteq CONS$. □

Lemma 22 (*Subject reduction*)
For $\Gamma, \Gamma' \in Context(\zeta)$, $B, B', C \in Term(\zeta)$ *and* $\Gamma \vdash B : C$

(i) $B \twoheadrightarrow B' \Rightarrow \Gamma \vdash B' : C$,
(ii) $\Gamma \twoheadrightarrow \Gamma' \Rightarrow \Gamma' \vdash B : C$.

Proof. By simultaneous induction on the length of the derivation of $\Gamma \vdash B : C$, distinguishing cases according to the last applied inference rule. We prove the lemma for one step reductions, so $B \to B'$ or $\Gamma \to \Gamma'$.

Proof of (*i*). If the last rule is (start), there is no redex in B.

If the last rule is (weakening), (conversion), (Π-rule) or (λ-rule), we are done by the induction hypothesis. (For (Π-rule) and (λ-rule), use also the induction hypothesis on (ii).)

If the last rule is (application), we distinguish subcases $B \equiv B_1 B_2 \to B_1' B_2' \equiv B'$ (the reduction taking place inside B_1 or B_2), and $B \equiv (\lambda x{:}A_1.A_2) B_2 \to A_2[x := B_2] \equiv B'$. We treat the last case. Let $B \equiv (\lambda x{:}A_1.A_2) B_2 \to A_2[x := B_2]$ and

$$\frac{\Gamma \vdash \lambda x{:}A_1.A_2 : \Pi x{:}C_1.C_2 \quad \Gamma \vdash B_2 : C_1}{\Gamma \vdash (\lambda x{:}A_1.A_2) B_2 : C_2[x := B_2]}.$$

Apply the stripping lemma to $\Gamma \vdash \lambda x{:}A_1.A_2 : \Pi x{:}C_1.C_2 : s$ to find

$$\Gamma \vdash A_1 : s_1 \tag{1}$$

$$\Gamma, x{:}A_1 \vdash A_2 : C_2' \quad \text{for some } C_2' = C_2 \tag{2}$$

$$\Gamma, x{:}C_1 \vdash C_2 : s_2 \tag{3}$$

and $$A_1 = C_1.$$

Further we have $$\Gamma \vdash B_2 : C_1 \tag{4}$$

Applying (conversion) to (1) and (4), we find that

$$\Gamma \vdash B_2 : A_1. \tag{5}$$

With the substitution lemma we conclude from (2) and (5) that

$$\Gamma \vdash A_2[x := B_2] : C_2'[x := B_2]. \tag{6}$$

Again with the substitution lemma we conclude from (3) and (4) that

$$\Gamma \vdash C_2[x := B_2] : s_2 \tag{7}$$

and to (6) and (7) one can apply (conversion) to obtain $\Gamma \vdash A_2[x:=B_2]:C_2[x:=B_2]$, which was to be proved.

Proof of (ii). If the last rule is (Π-rule), (λ-rule), (application) or (conversion), we are immediately done by induction hypothesis.

If the last rule is (start) or (weakening), we distinguish cases according to whether or not the reduction took place in the last declaration of the context. If this is not so, we are done by induction hypothesis. If the reduction did take place in the last declaration of the context, apply the induction hypothesis on (i) and (conversion) (in case (start) was the last rule) and we are done. □

Corollary 23
For $\Gamma \in Context(\zeta)$, $B, C, C' \in Term(\Gamma)$

$$\Gamma \vdash B:C \,\&\, C \twoheadrightarrow C' \Rightarrow \Gamma \vdash B:C'.$$

Proof. This follows immediately from stripping (vi), applying subject reduction to the term C and (conversion). □

Lemma 24 (*Uniqueness of assignment for functional* GTSs)
Let ζ *be a functional* GTS. *For* $\Gamma \in Context(\zeta)$, $B, C, C' \in Term(\zeta)$

$$\Gamma \vdash B:C \,\&\, \Gamma \vdash B:C' \Rightarrow C = C'.$$

Proof. By induction on the structure of the pseudoterm B. As the proof is easy, we only treat one case, for $B \equiv \lambda x:B_1.B_2$. Then, $\Gamma, x:B_1 \vdash B_2:C_2$ and $\Gamma, x:B_1 \vdash B_2:C_2'$ for some terms C_2 and C_2' and $C = \Pi x:B_1.C_2$, $C' = \Pi x:B_1.C_2'$. By induction hypothesis $C_2' = C_2$. Hence $C = \Pi x:B_1.C_2 = C'$. □

Lemma 25 (*Strengthening modulo reduction for functional* GTSs)
For $\Gamma_1, x:A$, $\Gamma_2 \in Context(\zeta)$, $B, C \in Term(\zeta)$

$$\Gamma_1, x:A, \Gamma_2 \vdash B:C,\ x \notin FV(\Gamma_2) \cup FV(B)$$
$$\Rightarrow \exists C' \in Term(\zeta).[C \twoheadrightarrow C' \,\&\, \Gamma_1, \Gamma_2 \vdash B:C'].$$

Proof. By induction on the length of the derivation of $\Gamma_1, x:A, \Gamma_2 \vdash B:C$, distinguishing cases according to the last applied rule.

If the last rule is (start) or (Π-rule), we are done by the induction hypothesis and one application of the rule.

If the last rule is (weakening), we are done by the induction hypothesis (distinguish between $\Gamma_2 = \varnothing$ and $\Gamma_2 \neq \varnothing$.)

If the last rule is (conversion), we are done by induction hypothesis, Church–Rosser property and Corollary 23.

If $B \equiv \lambda z:B_1.B_2$, $C \equiv \Pi z:B_1.C_2$ and the last applied rule is (λ-rule), then $\Gamma_1, x:A, \Gamma_2 \vdash B_1:s_1$ and $\Gamma_1, x:A, \Gamma_2, z:B_1 \vdash B_2:C_2:s_2$. By induction hypothesis $\Gamma_1, \Gamma_2 \vdash B_1:s_1$ and $\Gamma_1, \Gamma_2, z:B_1 \vdash B_2:C_2'$ for some C_2' with $C_2 \twoheadrightarrow C_2'$.

Now, $\Gamma_1, \Gamma_2, z:B_1 \vdash C_2':s_2$ (By stripping (vi): $C_2' \equiv c \in \mathcal{C}$ and $C_2':s_2$ is an axiom or $\Gamma_1, \Gamma_2, z:B_1 \vdash C_2':s \in \mathcal{S}$ and $s \equiv s_2$ by uniqueness of assignment.) With one

application of (λ-rule), we conclude that $\Gamma_1, \Gamma_2 \vdash \lambda z{:}B_1.B_2{:}\Pi z{:}B_1.C_2'$, where $C \twoheadrightarrow \Pi z{:}B_1.C_2'$.

If $B \equiv B_1 B_2$, $C \equiv C_2[x{:=}B_2]$, and the last applied rule is (application), then $\Gamma_1, x{:}A, \Gamma_2 \vdash B_1{:}\Pi z{:}C_1.C_2$ and $\Gamma_1, x{:}A, \Gamma_2 \vdash B_2{:}C_1$. By induction hypothesis $\Gamma_1, \Gamma_2 \vdash B_1{:}\Pi z{:}C_1'.C_2'$ and $\Gamma_1, \Gamma_2 \vdash B_2{:}C_1''$, for some C_1', C_1'', C_2' with $C_1 \twoheadrightarrow C_1'$, $C_1 \twoheadrightarrow C_1''$ and $C_2 \twoheadrightarrow C_2'$. Take C_1''' such that $C_1' \twoheadrightarrow C_1'''$ and $C_1'' \twoheadrightarrow C_1'''$. With Corollary 23 and (application), we find $\Gamma_1, \Gamma_2 \vdash B_1 B_2{:}C_2'[x{:=}B_2]$ with $C \twoheadrightarrow C_2'[x{:=}B_2]$.

Corollary 26 (*Strengthening for functional* GTSs)
Let ζ *be a functional* GTS. *For* $\Gamma_1, x{:}A, \Gamma_2 \in \mathit{Context}(\zeta)$, $B, C \in \mathit{Term}(\zeta)$

$$\Gamma_1, x{:}A, \Gamma_2 \vdash B{:}C,\ x \notin FV(\Gamma_2) \cup FV(B{:}C) \Rightarrow \Gamma_1, \Gamma_2 \vdash B{:}C.$$

Proof. By Lemma 25, we know that $\Gamma_1, \Gamma_2 \vdash B{:}C'$, with $C \twoheadrightarrow C'$. By stripping (vi), there are two possibilities, $C \in \mathit{CONS}$ or $\Gamma_1, x{:}A, \Gamma_2 \vdash C{:}s \in \mathit{SORT}$. In the first case $C' \equiv C$, and we are done. In the second case, $\Gamma_1, \Gamma_2 \vdash C{:}s$ by Lemma 25 so with one application of (conversion): $\Gamma_1, \Gamma_2 \vdash B{:}C$. □

The idea of proving the previous corollary using Lemma 25 is due to Luo (1988), who has used it in a proof of strong normalization for an extended calculus of constructions.

Definition 27
Let $\Gamma \in \mathit{Context}(\zeta)$, $\Pi x{:}B_1.B_2$, $\lambda x{:}B_1.B_2$, $B_1 B_2 \in \mathit{Term}(\zeta)$.

(i) $\Pi x{:}B_1.B_2$ *is formed by* (s_1, s_2, s_3) *in* Γ iff
$(s_1, s_2, s_3) \in \mathit{RULE}$ &
$\Gamma \vdash B_1{:}s_1$ &
$\Gamma, x{:}B_1 \vdash B_2{:}s_2$.

(ii) $\lambda x{:}B_1.B_2$ *is formed by* (s_1, s_2, s_3) *in* Γ iff
$\exists C_1, C_2 \in \mathit{Term}(\zeta)$. $(s_1, s_2, s_3) \in \mathit{RULE}$ &
$\Gamma \vdash \lambda x{:}B_1.B_2{:}\Pi x{:}C_1.C_2$ &
$\Pi x{:}C_1.C_2$ *is formed by* (s_1, s_2, s_3) in Γ.

(iii) $B_1 B_2$ *is formed by* (s_1, s_2, s_3) *in* Γ iff
$\exists C_1, C_2 \in \mathit{Term}(\zeta)$. $(s_1, s_2, s_3) \in \mathit{RULE}$ &
$\Gamma \vdash B_1{:}\Pi x{:}C_1.C_2$ &
$\Pi x{:}C_1.C_2$ is formed by (s_1, s_2, s_3) in Γ.

Remark 28. By this definition, all composed terms are formed by a certain rule. For the first two cases, this follows immediately from the stripping-lemma. For the third case, this follows from the stripping-lemma 19 (v) and (vi) and case (i).

Lemma 29 (*Uniqueness of formation for functional* GTSs)
Let ζ *be a functional* GTS. *For* $\Gamma \in \mathit{Context}(\zeta)$, $B \in \mathit{Term}(\zeta)$, B *composed*

$$B \text{ formed by } (s_1, s_2, s_3) \text{ in } \Gamma \ \&\ B \text{ formed by } (s_1', s_2', s_3') \text{ in } \Gamma \Rightarrow s_1 \equiv s_1', s_2 \equiv s_2', s_3 \equiv s_3'.$$

Proof. We are done if we prove the following property. If $\Pi x{:}B_1.B_2$ formed by (s_1,s_2,s_3) in Γ, $\Pi x{:}B_1'.B_2'$ formed by (s_1',s_2',s_3') in Γ and $\Pi x{:}B_1.B_2 \twoheadrightarrow \Pi x{:}B_1'.B_2'$, then $s_1 \equiv s_1'$, $s_2 \equiv s_2'$ and $s_3 \equiv s_3'$.

For the cases $M \equiv \lambda x{:}B_1.B_2$ and $M \equiv B_1 B_2$, the proof then follows by the uniqueness of assignment. Namely, if $\Gamma \vdash \lambda x{:}B_1.B_2{:}\Pi x{:}C_1.C_2$ and $\Gamma \vdash \lambda x{:}B_1.B_2{:}\Pi x{:}C_1'.C_2'$, respectively $\Gamma \vdash B_1{:}\Pi x{:}C_1.C_2$ and $\Gamma \vdash B_1{:}\Pi x{:}C_1'.C_2'$, with $\Pi x{:}C_1.C_2$ formed by (s_1,s_2,s_3) in Γ and $\Pi x{:}C_1'.C_2'$ formed by (s_1',s_2',s_3') in Γ, then $\Pi x{:}C_1.C_2 = \Pi x{:}C_1'.C_2'$. So, take (with Church–Rosser) $\Pi x{:}C_1''.C_2''$, such that $\Pi x{:}C_1'.C_2' \twoheadrightarrow \Pi x{:}C_1''.C_2''$ and $\Pi x{:}C_1.C_2 \twoheadrightarrow \Pi x{:}C_1''.C_2''$. Then $\Pi x{:}C_1''.C_2''$ is formed by (s_1,s_2,s_3) and (s_1',s_2',s_3') in Γ, so $s_1 \equiv s_1'$, $s_2 \equiv s_2'$ and $s_3 \equiv s_3'$.

The proof of the property runs as follows. Let $\Pi x{:}B_1.B_2$ be formed by (s_1,s_2,s_3) in Γ, $\Pi x{:}B_1'.B_2'$ formed by (s_1',s_2',s_3') in Γ and $\Pi x{:}B_1.B_2 \twoheadrightarrow \Pi x{:}B_1'.B_2'$. Then $\Gamma \vdash B_1{:}s_1$, $\Gamma, x{:}B_1 \vdash B_2{:}s_2$ and $\Gamma \vdash B_1'{:}s_1'$, $\Gamma, x{:}B_1' \vdash B_2'{:}s_2'$. By subject reduction and uniqueness of assignment $s_1 \equiv s_1'$ and $s_2 \equiv s_2'$. So $s_3 \equiv s_3'$. □

Lemma 30 (*Classification for injective* GTSs)
Let ζ *be an injective* GTS. *For* $s,s' \in \mathcal{SORT}$, $s \not\equiv s'$,

(i) $s\text{-}Term(\zeta) \cap s'\text{-}Term(\zeta) = \varnothing$
(ii) $s\text{-}Elt(\zeta) \cap s'\text{-}Elt(\zeta) = \varnothing$.

Proof. The proof of (i) and (ii) is simultaneous, by induction on the structure of pseudoterms. We only treat the induction step for variables and for terms B of the form $B_1 B_2$. For the other induction steps, the statement follows immediately from the induction hypothesis using the injectivity properties.

Let $\Gamma \vdash x{:}B{:}s$ and $\Gamma' \vdash x{:}B'{:}s'$. Then $x{:}A \in \Gamma$ for certain A with $A = B$, and $\Gamma \vdash A{:}s_0$ and $x{:}A' \in \Gamma'$ for certain A' with $A' = B'$ and $\Gamma' \vdash A'{:}s_0$. (Where s_0 is the sort for which $x \in \mathcal{VAR}^{s_0}$). By Church–Rosser, subject reduction and uniqueness of assignment $s_0 \equiv s$ and $s_0 \equiv s'$. For $\Gamma \vdash x{:}s$ and $\Gamma' \vdash x{:}s'$ the argument is similar.

Let $\Gamma \vdash B_1 B_2{:}C$ and $\Gamma' \vdash B_1 B_2{:}C'$. Then $\Gamma \vdash B_1{:}\Pi x{:}C_1.C_2{:}s_3$, $\Gamma \vdash B_2{:}C_1{:}s_1$, $\Gamma, x{:}C_1 \vdash C_2{:}s_2$ and $\Gamma' \vdash B_1{:}\Pi x{:}C_1'.C_2'{:}s_3'$, $\Gamma' \vdash B_2{:}C_1'{:}s_1'$, $\Gamma', x{:}C_1' \vdash C_2'{:}s_2'$, for certain terms C_1, C_2, C_1', C_2' and $(s_1,s_2,s_3), (s_1',s_2',s_3') \in \mathcal{RULE}$ with $C_2[x{:=}B_2] = C$ and $C_2'[x{:=}B_2] = C'$.

By induction hypothesis $s_1 \equiv s_1'$, $s_3 \equiv s_3'$ and so $s_2 \equiv s_2'$.

By substitution $\Gamma \vdash C_2[x{:=}B_2]{:}s_2$ and $\Gamma' \vdash C_2'[x{:=}B_2]{:}s_2'$.

If now $\Gamma \vdash C{:}s$ and $\Gamma' \vdash C'{:}s'$, then by Church–Rosser, subject reduction and uniqueness of assignment, $s \equiv s_2$ and $s' \equiv s_2'$, so $s \equiv s'$.

If $C \equiv s$ and $C \equiv s'$, then by subject reduction and uniqueness of assignment $s{:}s_2$ and $s'{:}s_2$ are axioms, so $s \equiv s'$. □

The previous lemma motivates terminologies such as 't is a *s-Term*', or 't is a *s-Elt*'. These notions are not ambiguous in an injective GTS. That the lemma does not hold in general for systems which are not injective is shown by the following example.

Example 31
Let ζ be given by

$$\mathit{SORT} = \{*, *', \nabla, \nabla', \Delta, \Delta', \Box\},$$

$$\mathit{AXIOM} = \{*:*', \nabla:\nabla', \Delta:\Delta'\},$$

$$\mathit{RULE} = \{(*, \nabla, \Box), (*, \Delta, \Box)\},$$

$$\alpha \in \mathit{VAR}^{*'}, \beta \in \mathit{VAR}^{\Delta'}, \gamma \in \mathit{VAR}^{\nabla'}, x \in \mathit{VAR}^{*}, f \in \mathit{VAR}^{\Box},$$

then ζ is not injective and

$$\alpha:*, \beta:\Delta, f:\alpha \to \beta, x:\alpha \vdash fx:\beta:\Delta,$$

$$\alpha:*, \gamma:\nabla, f:\alpha \to \gamma, x:\alpha \vdash fx:\gamma:\nabla,$$

so $fx \in \Delta\text{-}\mathit{Elt}(\zeta)$ and $fx \in \nabla\text{-}\mathit{Elt}(\zeta)$.

Corollary 32 (Uniqueness of formation for injective GTSs)
Let ζ be an injective GTS. *For $\Gamma, \Gamma' \in \mathit{Context}(\zeta)$, $M \in \mathit{Term}(\zeta)$, M composed*

$$M \textit{ formed by } (s_1, s_2, s_3) \textit{ in } \Gamma \;\&\; M \textit{ formed by } (s_1', s_2', s_3') \textit{ in } \Gamma'$$
$$\Rightarrow s_1 \equiv s_1', s_2 \equiv s_2', s_3 \equiv s_3'.$$

Proof. Using the classification lemma for injective ζ, the proof runs just like the proof of the uniqueness of formation lemma for functional ζ. □

This corollary allows us to use the terminology 'formed by' without mentioning the context Γ, in case of an injective GTS. That the corollary is not true for just functional GTS is shown by the following example.

Example 33
Let the system ζ be given by

$$\mathit{SORT} = \{s_1, s_2, s_2', s_3\},$$

$$\mathit{AXIOM} = \{s_1:s_3, s_2:s_3, s_2':s_3\},$$

$$\mathit{RULE} = \{(s_1, s_2, s_3), (s_1, s_2', s_3)\},$$

$$y, z \in \mathit{VAR}^{s_3}, x \in \mathit{VAR}^{s_1}.$$

Then $\Pi x{:}y.z$ is formed by (s_1, s_2, s_3) in the context $y:s_1, z:s_2$,

$\Pi x{:}y.z$ is formed by (s_1, s_2', s_3) in the context $y:s_1, z:s_2'$.

3 Barendregt's cube of typed lambda calculi

Barendregt's cube consists of a coherent collection of eight type systems, each one corresponding with a vertex of a cube such that there is an inclusion relation along the edges of the cube. The systems of the cube will be defined by giving for each of