On the Subject Reduction Property for Algebraic Type Systems

G. Barthe[1] and P-A. Melliès[2]

[1] CWI, PO Box 94079, 1090GB Amsterdam, The Netherlands.
Email: gilles@cwi.nl
[2] LFCS, University of Edinburgh, King's Buildings, Edinburgh EH9 3JZ, Scotland.
Email: paulm@dcs.ed.ac.uk

Abstract. *Algebraic type systems provide a general framework for the study of the interaction between typed λ-calculi and typed rewriting systems. A major problem in the development of a general theory for algebraic type systems is to prove that typing is preserved under reduction (Subject Reduction lemma). In this paper, we propose a general technique to prove Subject Reduction for a large class of algebraic type systems. The idea is to consider for every (functional) algebraic type system a labelled syntax for which Subject Reduction is easy to prove and then prove the equivalence between the labelled and standard syntaxes whenever the labelled system is strongly normalising. The equivalence can then be used to recover confluence, strong normalisation and subject reduction for the standard syntax.*

1 Introduction

λ-calculus and term-rewriting are two fundamental computational paradigms. When combined, they give rise to the class of algebraic-functional languages [5, 13, 14, 20]. Recently, H. Geuvers and the first author have proposed a general framework for the classification and study of algebraic-functional languages: *algebraic type systems* [11].

Subject Reduction, also known as Type Safeness, states that types are closed under reduction. It is an important property of a type system: for instance it implies that correctness is preserved under evaluation and is needed in most strong normalisation proofs. Unfortunately, it is unknown whether Subject Reduction holds for an arbitrary algebraic type system. Indeed, the reduction relation of algebraic type systems may not be confluent on pseudo-terms [13, 22] and as a result standard techniques to prove Subject Reduction [6, 19] cannot be used.

The problem. Let $\lambda\mathbf{S}$ be an algebraic type system. If $\Gamma \vdash M : A$ and $M \rightarrow_{\beta R} N$, then $\Gamma \vdash N : A$.

In this paper, we propose a general technique for proving Subject Reduction for a large class of algebraic type systems –and so provide a partial but useful answer to the problem. The central idea is to consider a labelled syntax for which Subject Reduction is easy to establish and then prove that, under suitable conditions, both syntaxes are equivalent. Our work, complemented with a generic proof of strong normalisation [8, 11, 25][3], provides a clear and widely applicable metatheory of algebraic type systems. A particular application of our work is a proof of strong normalisation of the algebraic λ-cube, see [8] for more detail. Another, perhaps more important, application is to contribute to a better understanding of the various presentations of type systems. Several presentations are used in the literature, each of which serves a specific purpose. For example, the labelled syntax we consider is best suited to give a semantics of type systems[2, 25, 27] while the standard syntax is best suited for programming and proof-checking. Our work establishes the equivalence between the two presentations for a large class of systems.

Contents of the paper In Section 2, we introduce the standard and labelled syntaxes of algebraic type systems. The Subject Reduction property for the labelled syntax is proved in Section 3 and the equivalence between the labelled and standard syntaxes is proved in Section 4. In Section 5, we consider an application of our results. Finally, we conclude in Section 6.

Preliminaries The paper assumes some basic familiarity with pure type systems –see for example [6, 19]– and term-rewriting –see for example [17, 23]. In order to fix notation and terminology, we briefly recall some fundamental notions. Throughout this subsection, we let X denote an arbitrary set and $\mathcal{R}$, $\mathcal{S}$ denote binary relations over X. Elements of X are called objects.

$\mathcal{R}.\mathcal{S}$ denotes the composition of $\mathcal{R}$ and $\mathcal{S}$. Moreover, we use the following notation (below R stands for reflexive, S for symmetric and T for transitive, C for closure):

Notion	RC	SC	TC	RTC	RSTC	Inverse	$\mathcal{R}^\omega \cdot (\mathcal{R}^{op})^\omega$
Notation	$\underline{\mathcal{R}}$	$\mathcal{R}^{\leftrightarrow}$	$\mathcal{R}^+$	$\mathcal{R}^\omega$	$=_\mathcal{R}$	$\mathcal{R}^{op}$	$\downarrow_\mathcal{R}$

Some of the relations will written as $\rightarrow_i$, in which case we use an ARS (abstract rewriting system) notation:

Usual notation	$\rightarrow_i^\omega$	$=_{\rightarrow_i}$	$\downarrow_{\rightarrow_i}$
ARS notation	$\twoheadrightarrow_i$	$=_i$	$\downarrow_i$

Definition 1 *A relation $\mathcal{R}$ is*

- locally confluent *if $\mathcal{R}^{op} \cdot \mathcal{R} \subseteq \downarrow_\mathcal{R}$.*
- confluent *if the relations $\downarrow_\mathcal{R}$ and $=_\mathcal{R}$ are equal.*
- Church-Rosser *on an object a if for every b, c such that $b\ (\mathcal{R}^\omega)^{op}\ a\ \mathcal{R}^\omega\ c$ there exists d such that $b\ \mathcal{R}^\omega\ d\ (\mathcal{R}^\omega)^{op}\ c$.*
- strongly normalising *on an object a if there is no infinite sequence*

$$a_0 \mathcal{R} a_1 \mathcal{R} a_2 \mathcal{R} \dots$$

[3] The last two proofs are concerned with a different syntax but may be adapted to that of this paper.

- canonical *on an object a if it is Church-Rosser and strongly normalising on a.*

Throughout the paper, we will make use of Newman's Lemma.

Lemma 2 (Newman's Lemma) *If $\mathcal{R}$ is locally confluent and strongly normalising on a, then $\mathcal{R}$ is Church-Rosser on a.*

2 Algebraic type systems

2.1 Motivation and background

Algebraic-functional languages combine typed λ-calculi and typed term-rewriting systems into an unified framework. These languages enhance traditional typed λ-calculi with a mechanism to define algebraic functions via rewrite rules. One of the simplest example of algebraic type system is obtained by extending the simply typed λ-calculus with a type of natural numbers nat, the constants $0 : \mathsf{nat}$, $s : \mathsf{nat} \rightarrow \mathsf{nat}$, $+ : \mathsf{nat} \rightarrow \mathsf{nat} \rightarrow \mathsf{nat}$ and the rewrite rules

$$\begin{aligned} 0 + x &\rightarrow x \\ x + 0 &\rightarrow x \\ sx + y &\rightarrow s(x + y) \\ x + sy &\rightarrow s(x + y) \end{aligned}$$

Algebraic-functional languages were originally designed as programming languages and built upon non-dependent calculi such as simply typed λ-calculus, second-order λ-calculus or higher-order λ-calculus [4, 3, 13, 14, 15, 20]. Recently, variants of algebraic-functional languages built upon systems of dependent types –typically the Extended Calculus of Constructions [24]– have been used for efficient proof-checking [9, 12, 10, 16]. In this approach, the typed term-rewriting system R is used in conjunction with a naming (or reflection) principle to decide efficiently whether two inhabitants of a given type A (typically a model of R) are related by the equality $=_A$ of A. Algebraic type systems with dependent types were first introduced –independently of the above mentioned application– in [5, 18] for the systems of the algebraic λ-cube and later in their general form in [11].

Remark In the above summary and throughout the paper, only first-order rewriting is discussed. However, some more complex form of term-rewriting can be considered. In fact, Jouannaud and Okada's original proposal for algebraic type systems [20], later followed in [4, 3, 5, 11, 18], is based on higher-order rewriting. Such a format allows for the definition of higher-order functions such as $\mathsf{maplist} : (\mathsf{nat} \rightarrow \mathsf{nat}) \rightarrow \mathsf{list} \rightarrow \mathsf{list}$ –where list is the type of lists of natural numbers:

$$\begin{aligned} \mathsf{maplist}(f, \mathsf{nil}) &\rightarrow \mathsf{nil} \\ \mathsf{maplist}(f, \mathsf{cons}(a, l)) &\rightarrow \mathsf{cons}(f\ a, \mathsf{maplist}(f, l)) \end{aligned}$$

The results of this paper apply to algebraic type systems with higher-order rewriting, the restriction to first-order rewriting being solely motivated by readability.

2.2 Algebraic type systems

In this subsection, we define algebraic type systems. The approach we follow is inspired from [5, 18] and is equivalent to that of [11].

Definition 3 *A* pre-specification *is a 6-tuple* $\lambda\mathbf{S} = (U, S, F, H, P, D)$ *where*

- U *is a set of* universes, S *is a set of* sorts *and* F *is a set of* function symbols;
- $H \subseteq (U \cup S) \times U$ *is a set of* axioms *s.t.* $\forall \tau \in S. \exists s \in U.\ (\tau, s) \in H$.
- $P \subseteq U \times U \times U$ *is a set of* rules;
- $D : F \to S^\star \times S$ *is a* declaration *function.*

For the sake of hygiene, we assume that U, S, F are pairwise disjoint. Throughout the rest of this paper, we let V be a fixed set of variables and let $\sigma, \tau, \ldots$ (resp. $f, g, \ldots$) range over sorts (resp. function symbols). Moreover we define the arity $\mathsf{ar}(f)$ of a function symbol $f \in F$ to be the length of the first component of $D(f)$. K is then defined as the set of function symbols of arity 0. To complete the specification of an ATS, we introduce algebraic reduction.

Definition 4 *Let* $\lambda\mathbf{S} = (U, S, F, H, P, D)$ *be a pre-specification.*

- *The set* L *of algebraic terms is given by the abstract syntax:*

$$L = V | f(L, \ldots, L)$$

  where in the last case the number of arguments applied to f *is* $\mathsf{ar}(f)$.
- *The set of variables of a term* t *is denoted by* $\mathsf{var}(t)$ *and is defined as usual.*
- *Let* $\xi : V \to S$. *The relation* $:_\xi \subseteq L \times S$ *is defined by the rules*

$$\frac{\xi(x) = \tau}{x :_\xi \tau} \qquad \frac{t_i :_\xi \tau_i \quad (1 \leq i \leq n)}{f(t_1, \ldots, t_n) :_\xi \sigma} \text{ if } D(f) = ((\tau_1\ \ldots\ \ldots \tau_n), \sigma)$$

- *A* rewrite rule *is a pair* (l, r) *of algebraic terms s.t.* $l \notin V$ *and* $\mathsf{var}(l) \subseteq \mathsf{var}(r)$ *and* $l, r :_\xi \tau$ *for some* $\xi : V \to S$ *and* $\tau \in S$.
- *A* rewrite system R *is a set of rewrite rules.*
- *Every rewrite system* R *may be seen as an unsorted rewrite system and thus induces a relation* $\to_{L(R)}$ *on* L.
- *Define the relation* $\to_{LL(R)}$ *by* $a \to_{LL(R)} b$ *if* $a \to_{L(R)} b$ *and* $a, b :_\xi \tau$ *for some* $\xi : V \to S$ *and* $\tau \in S$.

We can now define the notion of specification.

Definition 5 - *An* ATS specification *is a pair consisting of a pre-specification* $\lambda\mathbf{S} = (U, S, F, H, P, D)$ *and a rewrite system* R. *By abuse of notation, we write* $\lambda\mathbf{S} = (U, S, F, H, P, D, R)$.

- *Let* PROPERTY *be a property of relations (e.g. confluent or terminating). A specification* $\lambda\mathsf{S} = (U, S, F, A, P, D, R)$ *is* A-PROPERTY *if* $\rightarrow_{LL(R)}$ *is* PROPERTY.
- *A specification* $\lambda\mathsf{S} = (U, S, F, H, P, D, R)$ *is* functional *if* A *and* P *are partial maps.*

For the remaining of the paper, we assume:

Assumption 6 $\lambda\mathsf{S} = (U, S, F, H, P, D, R)$ *is an* ATS *specification.*

2.3 Standard syntax

The set T of *pseudo-terms* is defined by the abstract syntax:

$$T = V|U|S|TT|\Pi V : T.T|\lambda V : T.T|f(T, \ldots, T)$$

where in the last case, the number of arguments applied to f is $\mathsf{ar}(f)$. In other words, we only consider fully applied algebraic terms.

In order to provide a uniform framework to specify and compare the systems used in the literature, the rules for derivation, in Table 1, are parametrised by a binary relation $\mathcal{R}$ on pseudo-terms. For lack of space, only one deductive system $\vdash$ is considered here. The definition below makes use of contexts, substitutions and β-reduction. These are defined as usual.

Axiom	$\dfrac{}{\vdash_{\mathcal{R}} c : s}$	if $(c, s) \in H$
Function	$\dfrac{\Gamma \vdash_{\mathcal{R}} t_1 : \sigma_1 \quad \ldots \quad \Gamma \vdash_{\mathcal{R}} t_n : \sigma_n}{\Gamma \vdash_{\mathcal{R}} f(t_1, \ldots, t_n) : \tau}$	if $Df = ((\sigma_1 \ldots \sigma_n), \tau)$
Start	$\dfrac{\Gamma \vdash_{\mathcal{R}} A : s}{\Gamma, x : A \vdash_{\mathcal{R}} x : A}$	if $x \notin \Gamma$, $x \in V$
Weakening	$\dfrac{\Gamma \vdash_{\mathcal{R}} t : A \quad \Gamma \vdash_{\mathcal{R}} B : s}{\Gamma, x : B \vdash_{\mathcal{R}} t : A}$	if $x \notin \Gamma$ and $t \in S \cup U \cup V \cup K$
Product	$\dfrac{\Gamma \vdash_{\mathcal{R}} A : s_1 \quad \Gamma, x : A \vdash_{\mathcal{R}} B : s_2}{\Gamma \vdash_{\mathcal{R}} \Pi x : A.B : s_3}$	if $(s_1, s_2, s_3) \in P$
Application	$\dfrac{\Gamma \vdash_{\mathcal{R}} t : \Pi x : A.B \quad \Gamma \vdash_{\mathcal{R}} u : A}{\Gamma \vdash_{\mathcal{R}} t\, u : B[u/x]}$	
Abstraction	$\dfrac{\Gamma, x : A \vdash_{\mathcal{R}} t : B \quad \Gamma \vdash_{\mathcal{R}} (\Pi x : A.B) : s}{\Gamma \vdash_{\mathcal{R}} \lambda x : A.t : \Pi x : A.B}$	
Conversion	$\dfrac{\Gamma \vdash_{\mathcal{R}} u : A \quad \Gamma \vdash_{\mathcal{R}} B : s}{\Gamma \vdash_{\mathcal{R}} u : B}$	if $A\mathcal{R}B$

Table 1. $\mathcal{R}$-DEDUCTIVE SYSTEM FOR THE STANDARD SYNTAX

Definition 7 - $M \rightarrow_R N$ *if there exists a context* $C[.]$*, a rule* (l, r) *and a substitution* θ *s.t.* $M \equiv C[\theta l]$ *and* $N \equiv C[\theta r]$.

- $\rightarrow_{mix} \;=\; \rightarrow_\beta \cup \rightarrow_R$.
- $\vdash \;=\; \vdash_{\downarrow_{mix}}$.

One of the main obstacles in developing the meta-theory of algebraic type systems is the non-confluence of $\rightarrow_{mix}$.

Fact 8 **([22, 13])** $\rightarrow_{mix}$ *may not be confluent on pseudo-terms.*

Proof The following example is taken from [13]. Consider the term-rewriting system with one constant 0, one unary function s, a binary function minus and the rules:

$$\mathsf{minus}(sx, x) \rightarrow_R s0$$
$$\mathsf{minus}(x, x) \rightarrow_R 0$$

Let P be the fixed point of s, i.e. $P = (\lambda x : c.s(xx))(\lambda x : c.s(xx))$. We have $P \rightarrow_\beta sP$. Therefore

$$M \rightarrow_R 0$$
$$M \rightarrow_\beta \mathsf{minus}(sP, P) \rightarrow_R 1$$

Hence $\rightarrow_{mix}$ does not have unique normal forms; a fortiori it is not confluent. ■

As observed in [26], the failure of confluence is due to the non left-linearity of the rules.

2.4 Labelled syntax

The labelled syntax differs from the standard one by having labelled abstractions and labelled applications. The set T_e of labelled pseudo-terms is defined by the abstract syntax

$$T_e = V|U|S|\mathsf{app}^{\Pi V:T_e.T_e}(T_e, T_e)|\Pi V : T_e.T_e|\lambda^{\Pi V:T_e.T_e} V.T_e|f(T_e, \ldots, T_e)$$

where in the last case, the number of arguments applied to f is $\mathsf{ar}(f)$.

As for the standard syntax, we consider a class of deductive systems indexed by a binary relation $\mathcal{R}$ on (labelled) pseudo-terms. The rules for derivation are given in Table 2.

Definition 9 *A labelled pseudo-term M is* legal *w.r.t.* $\vdash^e_{\mathcal{R}}$ *if there is a context Γ and a pseudo-term A such that $\Gamma \vdash^e_{\mathcal{R}} M : A$. A labelled pseudo-context Γ is* legal *w.r.t.* $\vdash^e_{\mathcal{R}}$ *if there two pseudo-terms M and A such that $\Gamma \vdash^e_{\mathcal{R}} M : A$.*

In this paper two main deductive systems will be considered.

Definition 10 - Algebraic *reduction $\rightarrow_R$ is defined in the same way as for the standard syntax;*

- Tight *β-reduction $\rightarrow_{\beta_t}$ is defined as the compatible closure of*

$$\mathsf{app}^{\Pi x:A.B}(\lambda^{\Pi x:A.B}x.M, N) \rightarrow M[N/x]$$

Axiom	$\dfrac{}{\vdash^e_{\mathcal{R}} c : s}$	if $(c, s) \in H$
Function	$\dfrac{\Gamma \vdash^e_{\mathcal{R}} t_1 : \sigma_1 \quad \ldots \quad \Gamma \vdash^e_{\mathcal{R}} t_n : \sigma_n}{\Gamma \vdash^e_{\mathcal{R}} f(t_1, \ldots, t_n) : \tau}$	if $Df = ((\sigma_1 \ldots \sigma_n), \tau)$
Start	$\dfrac{\Gamma \vdash^e_{\mathcal{R}} A : s}{\Gamma, x : A \vdash^e_{\mathcal{R}} x : A}$	if $x \notin \Gamma$, $x \in V$
Weakening	$\dfrac{\Gamma \vdash^e_{\mathcal{R}} t : A \quad \Gamma \vdash^e_{\mathcal{R}} B : s}{\Gamma, x : B \vdash^e_{\mathcal{R}} t : A}$	if $x \notin \Gamma$ and $t \in S \cup U \cup V \cup K$
Product	$\dfrac{\Gamma \vdash^e_{\mathcal{R}} A : s_1 \quad \Gamma, x : A \vdash^e_{\mathcal{R}} B : s_2}{\Gamma \vdash^e_{\mathcal{R}} \Pi x : A.B : s_3}$	if $(s_1, s_2, s_3) \in P$
Application	$\dfrac{\Gamma \vdash^e_{\mathcal{R}} t : \Pi x : A.B \quad \Gamma \vdash^e_{\mathcal{R}} u : A}{\Gamma \vdash^e_{\mathcal{R}} \mathrm{app}^{\Pi x:A.B}(t, u) : B[u/x]}$	
Abstraction	$\dfrac{\Gamma, x : A \vdash^e_{\mathcal{R}} t : B \quad \Gamma \vdash^e_{\mathcal{R}} (\Pi x : A.B) : s}{\Gamma \vdash^e_{\mathcal{R}} \lambda^{\Pi x:A.B} x.t : \Pi x : A.B}$	
Conversion	$\dfrac{\Gamma \vdash^e_{\mathcal{R}} u : A \quad \Gamma \vdash^e_{\mathcal{R}} B : s}{\Gamma \vdash^e_{\mathcal{R}} u : B}$	if $A \mathcal{R} B$

Table 2. $\mathcal{R}$-DEDUCTIVE SYSTEM FOR THE LABELLED SYNTAX

- Loose *β-reduction $\rightarrow_{\beta_l}$ is defined as the compatible closure of*

$$\mathrm{app}^{\Pi x:A'.B'}(\lambda^{\Pi x:A.B} x.M, N) \rightarrow M[N/x]$$

- $\rightarrow_{mixt} \; = \; \rightarrow_R \cup \rightarrow_{\beta_t}$ *and* $\rightarrow_{mixl} \; = \; \rightarrow_R \cup \rightarrow_{\beta_l}$.
- $\vdash^e_t \; = \; \vdash^e_{\downarrow_{mixt}}$ *and* $\vdash^e_l \; = \; \vdash^e_{\downarrow_{mixl}}$.

The notions of tight reduction and loose reductions are due to T. Altenkirch [2]. Tight β-reduction requires the abstraction and application labels to match. In contrast, loose β-reduction does not impose any condition on labels.

Lemma 11 *1. $\rightarrow_{\beta_t}$ is locally confluent.*
2. If λS is A-confluent, then $\rightarrow_{mixt}$ is locally confluent.

Proof By induction on the structure of the terms. ∎

It is unclear whether tight β-reduction, which is not left-linear, is confluent.

2.5 Subject Reduction for the standard syntax

Before embarking on technicalities, let us analyze where the standard proof of subject reduction breaks down. The problem arises when trying to prove subject reduction for β-reduction: as noticed in [5, 18], one cannot prove subject reduction by induction on the length of the derivations. Indeed, the induction step

$$\frac{\Gamma \vdash \lambda x : A'.b : \Pi x : A.B \quad \Gamma \vdash a : A}{\Gamma \vdash (\lambda x : A'.b)\ a : B[a/x]}$$

fails if one wants to prove $\Gamma \vdash b[a/x] : B[a/x]$. If we follow the proof of subject reduction for pure type systems [6, 19], the induction step should be completed in four steps:

1. deduce from the generation lemma that $\Gamma, x : A' \vdash b : B'$ for some B' such that $\Pi x : A.B \downarrow_{mixt} C_1 \downarrow_{mixt} \dots \downarrow_{mixt} C_n \downarrow_{mixt} \Pi x : A'.B'$ (where the C_i's are legal);
2. use confluence to derive $A \downarrow_{mixt} A'$ and $B \downarrow_{mixt} B'$;
3. apply the conversion rule and substitution to get $\Gamma \vdash b[a/x] : B'[a/x]$;
4. apply the conversion rule once more to get $\Gamma \vdash b[a/x] : B[a/x]$.

However the induction step cannot be completed –at step 2– because of Fact 8. To circumvent this problem, we follow a different strategy to develop the meta-theory of $\vdash$ for functional, $\mathcal{A}$-confluent ATSs. We proceed in three steps:

1. prove subject reduction of $\rightarrow_{mixt}$ for a class of deductive systems $\vdash^{e}_{\mathcal{R}}$;
2. prove strong normalisation of the labelled syntax using subject reduction;
3. deduce from functionality and strong normalisation
   (a) the equivalence between labelled and unlabelled syntaxes
   (b) confluence, strong normalisation and subject reduction for the standard syntax.

The strategy was originally proposed by T. Altenkirch for the Calculus of Constructions [2] and was later applied to Pure Type Systems in [25]. In this paper, we treat Steps 1 and 3 for Algebraic Type Systems thoroughly. Step 2 is treated in [25] for Pure Type Systems and in [8] for Algebraic Type Systems.

3 The subject reduction property for the labelled syntax

Definition 12 *$\mathcal{S}$ has the Subject Reduction Property w.r.t $\mathcal{Q}$ ($\mathcal{Q}$-SR) if*

$$\Gamma \vdash^{e}_{\mathcal{S}} t : A \text{ and } t\mathcal{Q}u \Rightarrow \Gamma \vdash^{e}_{\mathcal{S}} u : A$$

In this section, we prove the Subject Reduction property w.r.t. $\rightarrow_{mixt}$ for a large class of $\mathcal{R}$-deductive systems.

The standard proof of subject reduction [6, 19] uses a *frontier* property: at each derivation step,

$$\frac{\Gamma_1 \vdash t_1 : A_1 \quad \cdots \quad \Gamma_k \vdash t_k : A_k}{\Delta \vdash u : B}$$

Δ and u can be constructed from the Γ_i's and the t_i's. Labelled systems do not fulfill this property because of the Application rule (where B appears in $\lambda^{\Pi x:A.B} x.t$). To recover this frontier property, we consider a variant $\Vdash_{\mathcal{R}}$ of the labelled syntax, where the Application rule is replaced by:

$$\text{Application+} \quad \frac{\Gamma \Vdash_{\mathcal{R}} t : \Pi x : A.B \quad \Gamma \Vdash_{\mathcal{R}} u : A \quad \Gamma \Vdash_{\mathcal{R}} \Pi x : A.B : s}{\Gamma \Vdash_{\mathcal{R}} \mathsf{app}^{\Pi x:A.B}(t, u) : B[u/x]}$$

Proposition 17 will show that this modification has in general no consequence on the set of derivable judgements. For now, we prove Subject Reduction for $\Vdash_{\mathcal{R}}$. Some preliminary closure results are needed.

Lemma 13 (Generation lemma) ($\mathbf{G}_c$) *if $\Gamma \Vdash_{\mathcal{R}} c : E$ and $c \in U \cup S$, there exists $s \in S$ such that $(c, s) \in H$ and $s\ \mathcal{R}^{\omega}\ E$;*

($\mathbf{G}_f$) *if $\Gamma \Vdash_{\mathcal{R}} f(t_1, \ldots, t_n) : E$ with $\mathsf{D}(f) = ((\sigma_1, \ldots, \sigma_n), \tau)$, then $\Gamma \Vdash_{\mathcal{R}} t_i : \sigma_i$ for $i = 1, \ldots, n$ and $\tau\ \mathcal{R}^{\omega}\ E$;*

($\mathbf{G}_x$) *if $\Gamma \Vdash_{\mathcal{R}} x : E$, then there exists B such that $(x : B) \in \Gamma$ and $B\ \mathcal{R}^{\omega}\ E$;*

($\mathbf{G}_{\mathsf{app}}$) *any derivation of $\Gamma \Vdash_{\mathcal{R}} \mathsf{app}^{\Pi x:A.B}(M, N) : E$ contains a derivation of $\Gamma \Vdash_{\mathcal{R}} M : \Pi x : A.B$ and $\Gamma \Vdash_{\mathcal{R}} N : A$ and $\Gamma \Vdash_{\mathcal{R}} \Pi x : A.B : s$ for some universe s. Moreover $B[N/x]\mathcal{R}^{\omega}E$.*

($\mathbf{G}_{\Pi}$) *any derivation of the judgement $\Gamma \Vdash_{\mathcal{R}} (\Pi x : A.B) : E$ contains derivations of $\Gamma \Vdash_{\mathcal{R}} A : s_1$ and $\Gamma, x : A \Vdash_{\mathcal{R}} B : s_2$ for some universes s_1, s_2. Moreover there exists $s_3 \in U$ such that $(s_1, s_2, s_3) \in P$ and $s_3\mathcal{R}^{\omega}E$.*

($\mathbf{G}_{\lambda}$) *any derivation of $\Gamma \Vdash_{\mathcal{R}} \lambda^{\Pi x:A.B}x.b : E$ contains a derivation of $\Gamma, x : A \Vdash_{\mathcal{R}} b : B$ and $\Gamma \Vdash_{\mathcal{R}} \Pi x : A.B : s$ for some universe s. Moreover $(\Pi x : A.B)\mathcal{R}^{\omega}E$.*

($\mathbf{G}_{\Gamma}$) *any derivation of $\Gamma, x : A \Vdash_{\mathcal{R}} M : B$ contains a derivation of $\Gamma \Vdash_{\mathcal{R}} A : s$ for some universe s.*

Lemma 14 (Substitution lemma) *Let $\Gamma_1, x : A, \Gamma_2$ be a context, let a, b, B be pseudo-terms. If $\mathcal{R}$ is closed under substitution then*

$$\left.\begin{array}{r}\Gamma_1, x : A, \Gamma_2 \Vdash_{\mathcal{R}} b : B \\ \Gamma_1 \Vdash_{\mathcal{R}} a : A\end{array}\right\} \Rightarrow \quad \Gamma_1, \Gamma_2[a/x] \Vdash_{\mathcal{R}} b[a/x] : B[a/x]$$

Lemma 15 (Correctness of Types) *(**C**) Suppose that $\mathcal{R}$ is closed under substitution. If $\Gamma \Vdash_{\mathcal{R}} a : A$ and $A \notin U$, then $\Gamma \Vdash_{\mathcal{R}} A : s$ for some universe s.*

Proof By induction on the structure of the derivation of $\Gamma \Vdash_{\mathcal{R}} a : A$. ■

The next result gives three general conditions for Subject Reduction to hold. $\mathbf{H}_1$ is needed to apply the above closure lemmas while $\mathbf{H}_2$ and $\mathbf{H}_3$ are needed to apply the induction hypothesis via a back-and-forth reasoning.

Theorem 16 (Subject Reduction Theorem) *Let $\mathcal{R}$ be a relation such that*

$\mathbf{H}_1$ *$\mathcal{R}$ is closed under substitution,*

$\mathbf{H}_2$ *if $Q_1 \rightarrow_{mixt} Q_2$ then $Q_1\ \mathcal{R}\ Q_2$*

$\mathbf{H}_3$ *if $Q_1 \rightarrow_{mixt} Q_2$ then $P[Q_2/x]\ \mathcal{R}\ P[Q_1/x]$ where P is any labelled pseudo-term.*

Assume $\Gamma \Vdash_{\mathcal{R}} M : A$ and $M \rightarrow_{mixt} M'$. Then $\Gamma \Vdash_{\mathcal{R}} M' : A$.

Proof See Appendix. ■

Proposition 17 *If $\mathcal{R}$ is closed under substitution then for every judgement (Γ, M, A):*

$$\Gamma \vdash^{e}_{\mathcal{R}} M : A \quad \Leftrightarrow \quad \Gamma \Vdash_{\mathcal{R}} M : A$$

Proof Both implications are proved by induction on the structure of derivations. The implication $(\Rightarrow)$ is proved using Correctness of Types. ■

Corollary 18 *If $\mathcal{R}$ verifies the hypotheses $\mathbf{H}_1$, $\mathbf{H}_2$ and $\mathbf{H}_3$, then it has the $\rightarrow_{mixt}$-SR property. In particular, $\vdash^e_l$ and $\vdash^e_t$ have the $\rightarrow_{mixt}$-SR property.*

4 Equivalence results

In this section, we establish under certain conditions an equivalence between

1. (a) labelled deductive systems;
2. (b) $\vdash^e_t$ and $\vdash$.

Only the most important equivalence results are stated here. There are further, more general, results which we omit for the lack of space.

4.1 A general equivalence result for labelled deductive systems

Throughout this subsection, $\mathcal{Q}$, $\mathcal{R}$ and $\mathcal{S}$ denote binary relations on labelled pseudo-terms.

Definition 19 - *$\mathcal{R} \sqsubseteq \mathcal{S}$ if for all judgements (Γ, M, A),*

$$\Gamma \vdash^e_{\mathcal{R}} M : A \Rightarrow \Gamma \vdash^e_{\mathcal{S}} M : A$$

- *$\mathcal{R} \simeq \mathcal{S}$ if $\mathcal{R} \sqsubseteq \mathcal{S} \sqsubseteq \mathcal{R}$.*
- *$\mathcal{R} < \mathcal{S}$ if for all judgements (Γ, M, A),*

$$(\Gamma \vdash^e_{\mathcal{S}} M : A \quad \text{and} \quad \Gamma \vdash^e_{\mathcal{S}} B : s \quad \text{and} \quad A\mathcal{R}B) \quad \Rightarrow \quad \Gamma \vdash^e_{\mathcal{S}} M : B$$

- *$\mathcal{R} \lessgtr \mathcal{S}$ if $\mathcal{R} < \mathcal{S} < \mathcal{R}$.*

Remark that $<$ is *not* transitive. Working at an abstract level, we show that all the labelled deductive systems satisfying certain properties are equivalent.

Proposition 20 *$\mathcal{R} < \mathcal{S} \Rightarrow \mathcal{R} \sqsubseteq \mathcal{S}$ and $\mathcal{R} \lessgtr \mathcal{S} \Leftrightarrow \mathcal{R} \simeq \mathcal{S}$.*

Proof See Appendix. ■

Proposition 21 *Assume that $\mathcal{S}$ has the $\mathcal{Q}$-SR property and is closed under substitutions.*

$$\mathcal{Q} < \mathcal{S} \text{ and } \mathcal{R} < \mathcal{S} \Rightarrow \mathcal{Q} \cdot \mathcal{R} < \mathcal{S}$$
$$\mathcal{R} < \mathcal{S} \text{ and } \mathcal{Q}^{op} < \mathcal{S} \Rightarrow \mathcal{R} \cdot \mathcal{Q}^{op} < \mathcal{S}$$
$$\mathcal{Q}^{\leftrightarrow} < \mathcal{S} \Rightarrow \mathcal{S} \simeq \mathcal{Q}^{\omega} \cdot \mathcal{S} \cdot (\mathcal{Q}^{op})^{\omega}$$

Proof See Appendix. ■

Corollary 22 (Equivalence Lemma) *Assume that $\mathcal{S}$ has the $\mathcal{Q}$-SR property and is closed under substitutions.*

$$\begin{aligned} \mathcal{Q}^{\leftrightarrow} < \mathcal{S} \quad &\Rightarrow \quad \downarrow_{\mathcal{Q}} \sqsubseteq \mathcal{S} \\ \mathcal{Q}^{\leftrightarrow} < \mathcal{S} < \downarrow_{\mathcal{Q}} \quad &\Rightarrow \quad \downarrow_{\mathcal{Q}} \simeq \mathcal{S} \end{aligned}$$

Proof See Appendix. ■

Theorem 23 (Equivalence theorem) *Let $\mathcal{R}$ verify the hypotheses $\mathbf{H}_1$, $\mathbf{H}_2$ and $\mathbf{H}_3$. Then $\downarrow_{mixt} \sqsubseteq \mathcal{R}$. Moreover $\mathcal{R} < \downarrow_{mixt} \quad \Rightarrow \quad \downarrow_{mixt} \simeq \mathcal{R}$.*

Proof See Appendix. ■

4.2 More labelled equivalences

In this subsection, we prove two further equivalence results for the labelled syntax. Both results will be used to prove the equivalence between the labelled and unlabelled syntaxes.

The first result is concerned with showing that under suitable conditions, $\vdash^{e}_{\downarrow_{\mathcal{R}}}$ is equivalent to another, easier to use, deductive system.

Definition 24 *Let $\mathcal{R}$ be a binary relation. The relation $\mathbf{T}(\mathcal{R})$ is defined by*

$$t \; \mathbf{T}(\mathcal{R}) \; u \quad \Longleftrightarrow \quad (t \; \mathcal{R} \; u \textit{ and } t \textit{ and } u \textit{ are legal w.r.t. } \vdash^{e}_{\downarrow_{\mathcal{R}}})$$

We have:

Lemma 25 *If $\mathcal{R}$ is closed under substitution and $\vdash^{e}_{\downarrow_{\mathcal{R}}}$ has $\mathcal{R}$-SR then $\downarrow_{\mathcal{R}} \simeq \downarrow_{\mathbf{T}(\mathcal{R})}$.*

Proof See Appendix. ■

We write $\to_{\mathbf{T}(mixt)}$ for $\mathbf{T}(\to_{mixt})$.

Corollary 26 $\downarrow_{\mathbf{T}(mixt)} \simeq \downarrow_{mixt}$.

An interesting point about $\to_{\mathbf{T}(mixt)}$ is that it is confluent when $\to_{mixt}$ is canonical on legal terms of $\vdash^{e}_{t}$.

The second result is concerned with an equivalence between $\vdash^{e}_{t}$ and $\vdash^{e}_{l}$. We start with a preliminary result:

Lemma 27 *Assume $\to_{\mathbf{T}(mixt)}$ is confluent. For every judgement $\Gamma \vdash^{e}_{t} M : A$ and labelled pseudo-term M',*

$$M \to_{mixl} M' \quad \Rightarrow \quad M \twoheadrightarrow^{+}_{mixt} M'$$

Proof See Appendix. ■

Proposition 28 *Assume $\to_{\mathbf{T}(mixt)}$ is confluent. Then $\downarrow_{mixt} \simeq \downarrow_{mixl}$.*

Proof The direct inclusion $\downarrow_{mixt} \sqsubseteq \downarrow_{mixl}$ follows from the inclusion $\downarrow_{mixt} \subseteq \downarrow_{mixl}$. The reverse inclusion $\downarrow_{mixl} \sqsubseteq \downarrow_{mixt}$ follows from Equivalence Lemma: Remark that we need Lemma 27 to show that $\vdash_{t}$ has the $\to_{mixl}$-SR property and that $(\downarrow_{mixl})^{\leftrightarrow} < \downarrow_{mixt}$. ■

4.3 Equivalence between labelled and unlabelled syntaxes

In this section, we establish the main equivalence result between labelled and unlabelled syntaxes. For the lack of space, we only consider the equivalence between $\vdash$ and $\vdash_t^e$.

There is an obvious translation from T_e to T which erases labels:

Definition 29 (The translation) *The map $||\cdot|| : T_e \to T$ is defined inductively as follows:*

- $||x|| = x$
- $||s|| = s,$
- $||\tau|| = \tau,$
- $||\Pi x : A.B|| = \Pi x : ||A||.||B||,$
- $||f(t_1, \ldots, t_n)|| = f(||t_1||, \ldots, ||t_n||),$
- $||\lambda^{\Pi x:A.B} x.M|| = \lambda x : ||A||.||M||,$
- $||\mathsf{app}^{\Pi x:A.B}(M, N)|| = ||M||\ ||N||.$

Erasure preserves typing.

Lemma 30 $\Gamma \vdash_t^e M : A \quad \Rightarrow \quad ||\Gamma|| \vdash ||M|| : ||A||$

Proof By an easy structural induction on the derivation of $\Gamma \vdash_t^e M : A$. ■

The fundamental fact about labels is that, under suitable conditions, every derivable judgement can be labelled without losing derivability. Throughout this subsection, we assume:

Assumption 31 *λS is a functional algebraic type system. Moreover $\to_{mixt}$ is canonical on legal terms of $\vdash_t^e$.*

We start with some preliminary results.

Proposition 32 (Unicity of types) *Assume $\Gamma \vdash_t^e M : A$ and $\Gamma \vdash_t^e M : B$. Then $A =_{T(mixt)} B$.*

Proof By induction on the structure of the derivation of $\Gamma \vdash_t^e M : A$. ■

Corollary 33 *Assume that $\Gamma \vdash_t^e M : A$ and $\Delta \vdash_t^e N : B$. If $\Gamma =_{\mathbf{T}(mixt)} \Delta$ and $M =_{\mathbf{T}(mixt)} N$, then $A =_{\mathbf{T}(mixt)} B$.*

Proof By confluence of $\to_{T(mixt)}$, there exists Ξ and P such that $\Gamma, \Delta \twoheadrightarrow_{T(mixt)} \Xi$ and $M, N \twoheadrightarrow_{T(mixt)} P$. By Subject Reduction, $\Xi \vdash_t^e P : A$ and $\Xi \vdash_t^e P : B$. By Unicity of Types, $A =_{\mathbf{T}(mixt)} B$. ■

Next we define for each legal term its canonical form.

Definition 34 (canonical forms) *Let M be legal. We define M^{can} as:*

- $x^{\mathsf{can}} = x$
- $s^{\mathsf{can}} = s$
- $\tau^{\mathsf{can}} = \tau$

- $(\Pi x : A.B)^{\mathsf{can}} = \Pi x : A^{\mathsf{can}}.B^{\mathsf{can}}$
- $(f(t_1, \ldots, t_n))^{\mathsf{can}} = f(t_1^{\mathsf{can}}, \ldots, t_n^{\mathsf{can}})$
- $(\lambda^{\Pi x:A.B} x.M)^{\mathsf{can}} = \lambda^{\Pi x:A^{\mathsf{can}}.B^{\mathsf{nf}}} x.M^{\mathsf{can}}$
- $(\mathsf{app}^{\Pi x:A.B}(M, N))^{\mathsf{can}} = \mathsf{app}^{\Pi x:A^{\mathsf{nf}}.B^{\mathsf{nf}}}(M^{\mathsf{can}}, N^{\mathsf{can}})$

where A^{nf} denotes the normal form of A w.r.t $\rightarrow_{mixt}$.

We remark that $||M|| = ||M^{\mathsf{can}}||$. One important property of $.^{\mathsf{can}}$ is that it identifies terms which have equal erasures.

Lemma 35 (Unicity of the canonical translation)

1. *for every legal contexts Γ and Δ, $||\Delta|| \equiv ||\Gamma|| \Rightarrow \Delta^{\mathsf{can}} \equiv \Gamma^{\mathsf{can}}$.*
2. *for every derivations $\Gamma \vdash_t^e M : C$ and $\Delta \vdash_t^e N : D$,*

$$(||\Gamma|| \equiv ||\Delta||) \wedge (||M|| \equiv ||N||) \Rightarrow M^{\mathsf{can}} \equiv N^{\mathsf{can}}$$

Proof See Appendix. ■

In order to be able to prove the equivalence between $\vdash$ and $\vdash_t^e$, it is necessary to show that standard reductions may be lifted to labelled ones. The following result is also useful to deduce subject reduction and strong normalisation from labelled subject reduction and strong normalisation.

Lemma 36 *Assume $\Gamma \vdash_t^e M : A$.*

1. *If $||M|| \rightarrow_{mix} N$ then there exists N' such that $M \twoheadrightarrow_{mixl}^{+} N'$ and $||N'|| \equiv N$.*
2. *If $||M|| \twoheadrightarrow_{mix}^{+} N$ then there exists N' such that $M \twoheadrightarrow_{mixt}^{+} N'$ and $||N'|| \equiv N$.*
3. *$\rightarrow_{mix}$ is confluent and strongly normalising on $||M||$.*
4. *If $\Gamma \vdash_t^e N : B$ with $||M|| \downarrow_{mix} ||N||$, then $M \downarrow_{mixt} N$.*

Proof See Appendix. ■

Collecting the previous results, we get:

Proposition 37 *If $\Gamma \vdash M : A$ is derivable then there exists a derivable judgement $\Gamma_\bullet \vdash_t^e M_\bullet : A_\bullet$ such that $\Gamma_\bullet, M_\bullet$ and $A_\bullet$ are canonical and*

$$||\Gamma_\bullet|| \equiv \Gamma \text{ and } ||M_\bullet|| \equiv M \text{ and } ||A_\bullet|| \equiv A$$

Proof See Appendix. ■

4.4 Aside: deductive systems with one-step conversion rule

Algebraic type systems are often defined with a deductive system using one-step conversion [5, 11, 18]. Unfortunately, it seems unclear how to prove Subject Reduction or an equivalence result for those systems[4]. Yet there is a slightly bigger deductive system for which Subject Reduction holds.

[4] For example, Subject Reduction for algebraic reduction cannot be proved by induction on the structure of derivations. The proof breaks down in the application case when the reduction takes place in the second argument of the application.

Definition 38 *$\to_{mix1}$ is the smallest reduction relation on labelled pseudo-terms such that for every $M, N, N' \in T_e$ and $x \in \mathsf{FV}(M)$, $N \to_{mixt} N' \Rightarrow M[N/x] \to_{mix1} M[N'/x]$. The relation* 1step *is defined as the symmetric closure of mix1.*

We have

Corollary 39 1step *has the $\to_{mixt}$-SR property and* 1step $\simeq \downarrow_{mixt}$.

Proof The first part follows from Theorem 16; the second part from the first and Theorem 23. ■

5 Application

5.1 The main result

Under suitable conditions, subject reduction for $\vdash$ can be deduced from subject reduction of $\vdash^e_t$.

Proposition 40 *Let $\lambda\mathsf{S}$ be a functional* ATS. *Assume $\to_{mixt}$ is canonical on legal terms of $\vdash^e_t$. Then $\vdash$ has the Subject Reduction property w.r.t. $\to_{mix}$. Moreover $\to_{mix}$ is canonical on legal terms of $\vdash$.*

Proof Assume $\Gamma \vdash M : A$. By Proposition 37, there exists a derivation $\Gamma_\bullet \vdash^e_t M_\bullet : A_\bullet$ with the expected translation property. By Lemma 36, there exists $N_\bullet$ such that $M_\bullet \twoheadrightarrow_{mixt} N_\bullet$ with $||N_\bullet|| \equiv N$. By Subject Reduction, $\Gamma_\bullet \vdash^e_t N_\bullet : A_\bullet$ and by translation $\Gamma \vdash N : A$. The second part of the proposition follows from Lemma 36. ■

5.2 The algebraic λ-cube

As a corollary of Proposition 40, we get strong normalisation of the algebraic λ-cube.

Corollary 41 *A-canonical systems of the algebraic λ-cube are strongly normalising.*

Proof By Proposition 40, it is enough to prove strong normalisation of $\to_{mixt}$ on legal terms of $\vdash^e_t$. See [8] for such a proof. ■

5.3 Aside: labelled strong normalisation from standard strong normalisation

It is important to know that the introduction of labels does not complicate the strong normalisation proofs. This can be made precise by considering an extension of algebraic type systems with the K-combinator [7]. We briefly describe the extension. At the level of pseudo-terms, a new clause is introduced in the abstract syntax to define the pseudo-term $\mathsf{K}\ t_1\ t_2$ for every pseudo-terms t_1 and

t_2. Moreover, a new reduction relation $\rightarrow_k$ is introduced; $\rightarrow_k$ is defined as the compatible closure of $\mathsf{K}\ x\ y \rightarrow x$ (hence the name K-combinator). At the level of the deductive system, one can define a K-deductive system $\vdash_K$ with the same rules as $\vdash$ and the extra rule

$$\frac{\Gamma \vdash_K a : A \quad \Gamma \vdash_K b : B}{\Gamma \vdash_K \mathsf{K}\ a\ b : A}$$

and extending the conversion rule into

$$\frac{\Gamma \vdash_K u : A \quad \Gamma \vdash_K B : s}{\Gamma \vdash u : B} \text{if } A \downarrow_{mixk} B$$

where $\rightarrow_{mixk} = \rightarrow_{mix} \cup \rightarrow_k$.

Proposition 42 *If $\vdash_k$ is strongly normalising w.r.t. $\rightarrow_{mixk}$, then $\vdash_t^e$ is strongly normalising w.r.t. $\rightarrow_{mixt}$.*

Proof By defining a reduction-preserving and derivation-preserving translation $\lceil . \rceil$ from labelled pseudo-terms to pseudo-terms with the K-combinator. The idea is that $\lceil . \rceil$ should be defined by

$$\begin{aligned} \lceil \mathsf{app}^{\Pi x:A.B}(M, N) \rceil &= \mathsf{K}\ (\lceil M \rceil\ \lceil N \rceil)\ \lceil \Pi x : A.B \rceil \\ \lceil \lambda^{\Pi x:A.B} x.b \rceil &= \lambda x : \lceil A \rceil . \mathsf{K} \lceil b \rceil\ \lceil B \rceil \end{aligned}$$

The other cases are defined in the obvious way. It is easy to check that this translation preserves reductions and derivations. ■

6 Conclusion

Proving the equivalence between the various formulations of algebraic or pure type systems is a vital exercise. It contributes to a better understanding of type systems and allows to derive results from one formalism to another. The main technical contribution of this paper is a proof of subject reduction for functional, A-confluent algebraic type systems which are strongly normalising for the labelled syntax. Although we have been unable to prove Subject Reduction for an arbitrary algebraic type system, our result is interesting because it is based on a simple technique and applies to an important class of algebraic type systems. Moreover, the technique in itself is interesting as it is very general and may be used in other type-theoretic frameworks where the reduction relation is not confluent on pseudo-terms. These include:

- pure and algebraic type systems with η-reduction [19],
- pure type systems with congruence types [10];
- abstract data type systems [21].

As such, it constitutes the first general technique to prove subject reduction for –unlabelled– type systems with a non-confluent reduction relation.

Acknowledgements

The two authors would like to thank H. Geuvers and R. Pollack for discussions on (algebraic) type systems. We would also like to thank the anonymous referees for suggesting many improvements to the paper.

References

1. S. Abramsky, D. Gabbay, and T. Maibaum, editors. *Handbook of Logic in Computer Science.* Oxford Science Publications, 1992.
2. T. Altenkirch. *Constructions, inductive types and strong normalisation.* PhD thesis, Laboratory for the Foundations of Computer Science, University of Edinburgh, 1994.
3. F. Barbanera and M. Fernández. Combining first and higher order rewrite systems with type assignment systems. In M.Bezem and J.-F. Groote, editors, *Proceedings of TLCA'93*, volume 664 of *Lecture Notes in Computer Science*, pages 60–74. Springer-Verlag, 1993.
4. F. Barbanera and M. Fernández. Modularity of termination and confluence in combinations of rewrite systems with λ_ω. In A. Lingas, R. Karlsson, and S. Karlsson, editors, *Proceedings of ICALP'93*, volume 700 of *Lecture Notes in Computer Science*, pages 657–668. Springer-Verlag, 1993.
5. F. Barbanera, M. Fernández, and H. Geuvers. Modularity of strong normalisation and confluence in the algebraic λ-cube. In *Proceedings of LICS'94*, pages 406–415. IEEE Computer Society Press, 1994.
6. H. Barendregt. Lambda calculi with types. In Abramsky et al. [1], pages 117–309. Volume 2.
7. G. Barthe. Extensions of pure type systems. In M. Dezani-Ciancaglini and G. Plotkin, editors, *Proceedings of TLCA'95*, volume 902 of *Lecture Notes in Computer Science*, pages 16–31. Springer-Verlag, April 1995.
8. G. Barthe. On strong normalisation of algebraic type systems. In preparation, 1997.
9. G. Barthe and H. Elbers. Towards lean proof checking. In J. Calmet and C. Limongelli, editors, *Proceedings of DISCO'96*, volume 1128 of *Lecture Notes in Computer Science*, pages 61–62. Springer-Verlag, 1996.
10. G. Barthe and H. Geuvers. Congruence types. In H. Kleine Buening, editor, *Proceedings of CSL'95*, volume 1092 of *Lecture Notes in Computer Science*, pages 36–51. Springer-Verlag, 1996.
11. G. Barthe and H. Geuvers. Modular properties of algebraic type systems. In G. Dowek, J. Heering, K. Meinke, and B. Möller, editors, *Proceedings of HOA'95*, volume 1074 of *Lecture Notes in Computer Science*, pages 37–56. Springer-Verlag, 1996.
12. G. Barthe, M. Ruys, and H. Barendregt. A two-level approach towards lean proof-checking. In S. Berardi and M. Coppo, editors, *Proceedings of TYPES'95*, volume 1158 of *Lecture Notes in Computer Science*, pages 16–35. Springer-Verlag, 1996.
13. V. Breazu-Tannen. Combining algebra and higher-order types. In *Proceedings of LICS'88*, pages 82–90. IEEE Computer Society Press, 1988.
14. V. Breazu-Tannen and J. Gallier. Polymorphic rewriting conserves algebraic strong normalisation. *Theoretical Computer Science*, 83:3–28, 1990.

15. V. Breazu-Tannen and J. Gallier. Polymorphic rewriting conserves algebraic confluence. *Information and Computation*, 114:1–29, 1994.
16. T. Coquand. Pattern matching in type theory. In B. Nordström, editor, *Informal proceedings of LF'92*, pages 66–79, 1992. Available from http://www.dcs.ed.ac.uk/lfcsinfo/research/types-bra/proc/index.html.
17. N. Dershowitz and J-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Formal models and semantics. Handbook of Theoretical Computer Science*, volume B, pages 243–320. Elsevier, 1990.
18. M. Fernandez. *Modèles de calcul multiparadigmes fondés sur la réécriture.* PhD thesis, Université Paris-Sud Orsay, 1993.
19. H. Geuvers. *Logics and type systems.* PhD thesis, University of Nijmegen, 1993.
20. J.-P. Jouannaud and M. Okada. Executable higher-order algebraic specification languages. In *Proceedings of LICS'91*, pages 350–361. IEEE Computer Society Press, 1991.
21. J.-P. Jouannaud and M. Okada. Abstract data type systems. *Theoretical Computer Science*, 173(2):349–391, 28 February 1997.
22. J.W. Klop. *Combinatory reduction systems.* Number 127 in Mathematical Centre Tracts. CWI, 1980.
23. J.W. Klop. Term-rewriting systems. In Abramsky et al. [1], pages 1–116. Volume 2.
24. Z. Luo. *Computation and Reasoning: A Type Theory for Computer Science.* Number 11 in International Series of Monographs on Computer Science. Oxford University Press, 1994.
25. P-A. Melliès and B. Werner. A generic proof of strong normalisation for pure type systems. Manuscript, 1996.
26. F. Müller. Confluence of the lambda calculus with left-linear algebraic rewriting. *Information Processing Letters*, 41:293–299, 1992.
27. Th. Streicher. *Correctness and Completeness of a Categorical Semantics of the Calculus of Constructions.* PhD thesis, University of Passau, 1989. Appeared as technical report MIP - 8913.

Appendix: proofs

Proof of Theorem 16 the proof proceeds along the same lines as in [6]. The following two facts are proved by simultaneous induction on the structure of derivations:

1. If $\Gamma \Vdash_{\mathcal{R}} M : C$ and $M \rightarrow_{mixt} M'$, then $\Gamma \Vdash_{\mathcal{R}} M' : C$.
2. If $\Gamma \Vdash_{\mathcal{R}} M : C$ and $\Gamma \rightarrow_{mixt} \Gamma'$, then $\Gamma' \Vdash_{\mathcal{R}} M : C$.

We treat the cases where the last rule is an abstraction, an application or a function rule:

- *abstraction rule:* assume $M \equiv \lambda^{\Pi x:A.B} x.t$ and $C \equiv \Pi x : A.B$. 2 follows from the induction hypothesis. As for 1, the interesting case is when the reduction occurs in A or in B, i.e. $M' \equiv \lambda^{\Pi x:A'.B} x.t$ or $M' \equiv \lambda^{\Pi x:A.B'} x.t$.

  Subcase 1: $M' \equiv \lambda^{\Pi x:A'.B} x.t$. In this case, we use the induction hypothesis to conclude $\Gamma, x : A' \Vdash_{\mathcal{R}} t : B$ and $\Gamma \Vdash_{\mathcal{R}} \Pi x : A'.B : s$. We may then apply the abstraction rule to get $\Gamma \Vdash_{\mathcal{R}} \lambda^{\Pi x:A'.B} x.t : \Pi x : A'.B$. By $\mathbf{H}_3$, we may apply the conversion rule to get $\Gamma \Vdash_{\mathcal{R}} \lambda^{\Pi x:A'.B} x.t : \Pi x : A.B$.

Subcase 2: $M' \equiv \lambda^{\Pi x:A.B'} x.t$. By induction hypothesis, $\Gamma \Vdash_{\mathcal{R}} \Pi x : A.B' : s$. By $\mathbf{G}_\Pi$, $\Gamma, x : A \Vdash_{\mathcal{R}} B' : s'$ for a universe s'. We apply the conversion rule thanks to $\mathbf{H}_2$ to deduce $\Gamma, x : A \Vdash_{\mathcal{R}} M : B'$. By abstraction, $\Gamma \Vdash_{\mathcal{R}} \lambda^{\Pi x:A.B'} x.M : (\Pi x : A.B')$. We next apply the conversion rule thanks to $\mathbf{H}_3$ to deduce $\Gamma \Vdash_{\mathcal{R}} \lambda^{\Pi x:A.B'} x.M : (\Pi x : A.B)$.

- *application rule:* assume $M \equiv \mathsf{app}^{\Pi x:A.B}\ t\ u$ and $C \equiv B[u/x]$. It is easy to prove 2. As for 1, we treat four subcases:

Subcase 1: top reduction: $M \equiv \mathsf{app}^{\Pi x:A.B}(\lambda^{\Pi x:A.B} x.b, u)$ and $M' \equiv b[u/x]$. By $\mathbf{G}_\lambda$: $\Gamma, x : A \Vdash_{\mathcal{R}} b : B$. By the Substitution Lemma, $\Gamma \Vdash_{\mathcal{R}} b[u/x] : B[u/x]$.

Subcase 2: inside u: $M \equiv \mathsf{app}^{\Pi x:A.B}(t, u)$ and $M' \equiv \mathsf{app}^{\Pi x:A.B}(t, u')$ with $u \rightarrow_{mixt} u'$. By induction hypothesis, $\Gamma \Vdash_{\mathcal{R}} u' : A$. By construction, $\Gamma \Vdash_{\mathcal{R}} (\Pi x : A.B) : s$. By application+, $\Gamma \Vdash_{\mathcal{R}} \mathsf{app}^{\Pi x:A.B}(t, u') : B[u'/x]$. By $\mathbf{G}_\Pi$: $\Gamma, x : A \Vdash_{\mathcal{R}} B : s'$ for some universe s'. By substitution lemma, $\Gamma \Vdash_{\mathcal{R}} B[u/x] : s'$ We apply a conversion rule thanks to $\mathbf{H}_3$ to deduce $\Gamma \Vdash_{\mathcal{R}} \mathsf{app}^{\Pi x:A.B}(t, u') : B[u/x]$.

Subcase 3: inside B: just like Subcase 2 of the abstraction rule: if $M \equiv \mathsf{app}^{\Pi x:A.B}(t, u)$ and $M' \equiv \mathsf{app}^{\Pi x:A.B'}(t, u)$ with $B \rightarrow_{mixt} B'$ then by induction hypothesis on the premise $\Gamma \Vdash_{\mathcal{R}} (\Pi x : A.B) : s$ we deduce $\Gamma \Vdash_{\mathcal{R}} (\Pi x : A.B') : s$. By conversion thanks to $\mathbf{H}_2$ we deduce $\Gamma \Vdash_{\mathcal{R}} t : (\Pi x : A.B')$. By application+, $\Gamma \Vdash_{\mathcal{R}} \mathsf{app}^{\Pi x:A.B'}(t, u) : B'[u/x]$. By $\mathbf{G}_\Pi$, $\Gamma, x : A \Vdash_{\mathcal{R}} B : s'$ for some universe s'. By substitution lemma, $\Gamma \Vdash_{\mathcal{R}} B[u/x] : s'$. By conversion and $\mathbf{H}_3$, $\Gamma \Vdash_{\mathcal{R}} \mathsf{app}^{\Pi x:A.B'}(t, u) : B[u/x]$.

Subcase 4: inside A: simpler than the preceding case because A does not appear in the type of $\mathsf{app}^{\Pi x:A.B}(t, u)$.

- *function rule:* if $M \equiv f(t_1, \ldots, t_n)$ with $\mathsf{D}(f) = ((\sigma_1, \ldots, \sigma_n), \tau)$, then $A \equiv \tau$. The only interesting case here is when M is a redex, i.e. when $M \equiv f(t_1, \ldots, t_n)$ is matched to a rewrite rule $l \rightarrow r$ (of sort τ) by some substitution θ. So let $M \equiv \theta l$ and $M' \equiv \theta r$.

Fact 43 *Assume M is an algebraic term of sort τ. Assume $\mathsf{FV}(M) = \{x_1, \ldots, x_n\}$ with $x_i \in V_{\sigma_i}$ for $i = 1, \ldots, n$. Then $\Gamma \equiv x_1 : \sigma_1, \ldots, x_n : \sigma_n \Vdash_{\mathcal{R}} M : \tau$.*

So we know $\Delta \Vdash_{\mathcal{R}} l : \tau$ and $\Delta \Vdash_{\mathcal{R}} r : \tau$ for the canonical context Δ associated to l. Moreover, $\Gamma \Vdash_{\mathcal{R}} \theta x_i : \tau_i$ for every $(x_i : \tau_i) \in \Delta$. By substitution, $\Gamma \Vdash_{\mathcal{R}} M' : \tau$.

■

Proof of Proposition 20 the first statement is proved by structural induction on the derivations of $\vdash^e_{\mathcal{R}}$. The direct implication of the second statement follows immediately. As for the reverse implication of the second statement, suppose that $\mathcal{R} \simeq \mathcal{S}$; we show $\mathcal{R} < \mathcal{S}$. Assume

$$\Gamma \vdash^e_{\mathcal{S}} M : A \quad \Gamma \vdash^e_{\mathcal{S}} B : s \quad A \mathcal{R} B$$

implies thanks to $\mathcal{S} \sqsubseteq \mathcal{R}$ that

$$\Gamma \vdash^e_{\mathcal{R}} M : A \quad \Gamma \vdash^e_{\mathcal{R}} B : s \quad A\mathcal{R}B$$

By $\mathcal{R}$-conversion, $\Gamma \vdash^e_{\mathcal{R}} M : B$. By $\mathcal{R} \simeq \mathcal{S}$: $\Gamma \vdash^e_{\mathcal{S}} M : B$. Henceforth $\mathcal{R} < \mathcal{S}$ and symmetrically $\mathcal{R} \lessgtr \mathcal{S}$. Remark that $\mathcal{R} \lessgtr \mathcal{S}$ implies $\mathcal{R} \simeq \mathcal{S}$ with the first statement. So we are done. ■

Proof of Proposition 21 We prove the first statement. Suppose that

$$\Gamma \vdash^e_{\mathcal{S}} M : A \quad \Gamma \vdash^e_{\mathcal{S}} B : s \quad A(\mathcal{Q} \cdot \mathcal{R})B$$

There exists a pseudo-term C such that $A\mathcal{Q}C\mathcal{R}B$. We use that $\mathcal{S}$ is closed by substitution: $\Gamma \vdash^e_{\mathcal{S}} A : s'$ for some universe s' follows by Correctness of Types. By $\mathcal{Q}$-SR: $\Gamma \vdash^e_{\mathcal{S}} C : s'$. By $\mathcal{Q} < \mathcal{S}$: $\Gamma \vdash^e_{\mathcal{S}} M : C$. By $\mathcal{R} < \mathcal{S}$: $\Gamma \vdash^e_{\mathcal{S}} M : B$.

The proof of the second statement is (nearly) dual. Suppose that:

$$\Gamma \vdash^e_{\mathcal{S}} M : A \quad \Gamma \vdash^e_{\mathcal{S}} B : s \quad A(\mathcal{R} \cdot \mathcal{Q}^{op})B$$

There is a pseudo-term C such that $A\mathcal{R}C\mathcal{Q}^{op}B$. By $\mathcal{Q}$-SR: $\Gamma \vdash^e_{\mathcal{S}} C : s$. By $\mathcal{R} < \mathcal{S}$: $\Gamma \vdash^e_{\mathcal{S}} M : C$. By $\mathcal{Q}^{op} < \mathcal{S}$: $\Gamma \vdash^e_{\mathcal{S}} M : B$. We are done.

The proof of the third statement: Note that $\mathcal{S} < \mathcal{S}$. Hence, we may apply the first and second statement as many times as wished. By continuity of $<$:

$$\mathcal{Q}^{\omega} \cdot \mathcal{S} \cdot (\mathcal{Q}^{op})^{\omega} < \mathcal{S}$$

We deduce from $\mathcal{S} \subseteq \mathcal{Q}^{\omega} \cdot \mathcal{S} \cdot (\mathcal{Q}^{op})^{\omega}$ that $\mathcal{S} \sqsubseteq \mathcal{Q}^{\omega} \cdot \mathcal{S} \cdot (\mathcal{Q}^{op})^{\omega}$. We are done with the first statement of proposition 20. ■

Proof of Corollary 22 the last statement is easy to prove with proposition 20. As for the first statement, we prove the following sequence of inequalities

$$\downarrow_{\mathcal{Q}} \equiv \mathcal{Q}^{\omega} \cdot (\mathcal{Q}^{op})^{\omega} \sqsubseteq \mathcal{Q}^{\omega} \cdot \underline{\mathcal{S}} \cdot (\mathcal{Q}^{op})^{\omega} \simeq \underline{\mathcal{S}} \simeq \mathcal{S}$$

We proceed in reverse order. $\underline{\mathcal{S}} \simeq \mathcal{S}$ is easy. It follows that $\underline{\mathcal{S}}$ has the $\mathcal{Q}$-SR property and that $\mathcal{Q} < \underline{\mathcal{S}}$. We apply Proposition 21 to get $\underline{\mathcal{S}} \simeq \mathcal{Q}^{\omega} \cdot \underline{\mathcal{S}} \cdot (\mathcal{Q}^{op})^{\omega}$. The last inequality follows $\mathcal{Q}^{\omega} \cdot (\mathcal{Q}^{op})^{\omega} \subseteq \mathcal{Q}^{\omega} \cdot \underline{\mathcal{S}} \cdot (\mathcal{Q}^{op})^{\omega}$. ■

Proof of Theorem 23 we only prove the first part as the second part is easy. Let $\mathcal{Q}$ be $\rightarrow_{mixt}$. It follows from $\mathbf{H}_2$ and $\mathbf{H}_3$ that $\mathcal{Q}^{\leftrightarrow} \subseteq \mathcal{R}$ therefore $\mathcal{Q}^{\leftrightarrow} < \mathcal{R}$. On the other hand Theorem 16 shows that $\mathcal{R}$ has the $\mathcal{Q}$-SR property. Hence we can apply corollary 22 to get $\downarrow_{mixt} \sqsubseteq \mathcal{R}$. ■

Proof of Lemma 25 the direction $\downarrow_{\mathbf{T}(\mathcal{R})} \sqsubseteq \downarrow_{\mathcal{R}}$ is the consequence of $\mathbf{T}(\mathcal{R}) \subset \mathcal{R}$. The reverse direction is a nice application of lemma 20. To prove that $\mathcal{R} < \downarrow_{\mathbf{T}(\mathcal{R})}$ suppose that $\Gamma \vdash^e_{\downarrow\mathbf{T}(\mathcal{R})} M : A$ and $\Gamma \vdash^e_{\downarrow\mathbf{T}(\mathcal{R})} B : s$ and $A \downarrow_{\mathcal{R}} B$. The following properties induce $A \downarrow_{\mathbf{T}(\mathcal{R})} B$:

1. the relation $\downarrow_{\mathcal{R}}$ is closed under substitution, so A is legal w.r.t. $\vdash^e_{\downarrow_{\mathcal{R}}}$ by Correctness of Types,
2. B is legal, $A \downarrow_{\mathcal{R}} B$ and $\vdash^e_{\downarrow_{\mathcal{R}}}$ has $\mathcal{R}$-SR.

Hence, the Conversion rule can be applied in $\vdash^{e}_{\downarrow_{\mathbf{T}(\mathcal{R})}}$ in order to get $\Gamma \vdash^{e}_{\downarrow_{\mathbf{T}(\mathcal{R})}} M : B$. We conclude that $\downarrow_{\mathcal{R}} < \downarrow_{\mathbf{T}(\mathcal{R})}$ and so $\downarrow_{\mathcal{R}} \sqsubseteq \downarrow_{\mathbf{T}(\mathcal{R})}$. ■

Proof of Lemma 27 by induction on the length of the derivation. Note that we only have to prove the result for $M \rightarrow_{\beta_l} M'$ as $\rightarrow_{R} \subseteq \rightarrow_{mixt}$. The only interesting case is when the last rule is an application rule and the subject of the judgement is a redex w.r.t. $\rightarrow_{\beta_l}$. So assume the last rule is

$$\frac{\Gamma \vdash^{e}_{t} \lambda^{\Pi x:A.B} x.t : (\Pi x : A'.B') \qquad \Gamma \vdash^{e}_{t} u : A'}{\Gamma \vdash^{e}_{t} \mathsf{app}^{\Pi x:A'.B'}(\lambda^{\Pi x:A.B} x.t, u) : B'[u/x]} \quad \cdot$$

with $M \equiv \mathsf{app}^{\Pi x:A'.B'}(\lambda^{\Pi x:A.B} x.t, u)$ and $M' \equiv t[u/x]$. To show that $M \twoheadrightarrow_{mixt} M'$. We use the fact that $\downarrow_{\mathbf{T}(mixt)} \simeq \downarrow_{mixt}$. By generation on $\vdash^{e}_{\downarrow_{\mathbf{T}(mixt)}}$ (which is equivalent to $\vdash^{e}_{t}$), $\Pi x : A.B \ =_{\mathbf{T}(mixt)} \ \Pi x : A'.B'$. By confluence of $\rightarrow_{\mathbf{T}(mixt)}$, there exists A'' and B'' such that $A, A' \twoheadrightarrow_{mixt} A''$ and $B, B' \twoheadrightarrow_{mixt} B''$. Therefore

$$M \twoheadrightarrow_{mixt} \mathsf{app}^{\Pi x:A''.B''}(\lambda^{\Pi x:A''.B''} x.t, u) \rightarrow_{\beta_t} t[u/x]$$

and we are done. ■

Proof of Lemma 35 by induction on the derivation of $\Gamma \vdash^{e}_{t} M : C$. We treat the case where the last rule is an application, an abstraction or a weakening:

- *application:* assume the last step is

  $$\frac{\Gamma \vdash^{e}_{t} t : \Pi x : A.B \qquad \Gamma \vdash^{e}_{t} u : A}{\Gamma \vdash^{e}_{t} \mathsf{app}^{\Pi x:A.B}(t, u) : B[u/x]}$$

  with $M \equiv \mathsf{app}^{\Pi x:A.B}(t, u)$. 1 is easy to prove. As for 2, assume $N \equiv \mathsf{app}^{\Pi x:A'.B'}(t', u')$, $||\Gamma|| \equiv ||\Delta||$ and $||M|| \equiv ||N||$. We show $M^{\mathsf{can}} \equiv N^{\mathsf{can}}$, i.e.

  $$\mathsf{app}^{\Pi x:A^{\mathsf{nf}}.B^{\mathsf{nf}}}(t^{\mathsf{can}}, u^{\mathsf{can}}) \equiv \mathsf{app}^{\Pi x:A'^{\mathsf{nf}}.B'^{\mathsf{nf}}}(t'^{\mathsf{can}}, u'^{\mathsf{can}})$$

  By $\mathbf{G}_{\mathsf{app}}$, $\Delta \vdash^{e}_{t} t' : \Pi x : A'.B'$ and $\Delta \vdash^{e}_{t} u' : A'$. Note that $||M|| \equiv ||N|| \Rightarrow (||t|| \equiv ||t'||$ and $||u|| \equiv ||u'||)$. We can use the induction hypothesis on the premises $\Gamma \vdash^{e}_{t} t : \Pi x : A.B$ and $\Gamma \vdash^{e}_{t} u : A$ and deduce that $\Gamma^{\mathsf{can}} \equiv \Delta^{\mathsf{can}}$, $t^{\mathsf{can}} \equiv t'^{\mathsf{can}}$ and $u^{\mathsf{can}} \equiv u'^{\mathsf{can}}$. By $\rightarrow_{mixt}$-SR, $\Gamma \downarrow_{\mathbf{T}(mixt)} \Delta$ and $t \downarrow_{\mathbf{T}(mixt)} t'$. By corollary 33 applied on

  $$\Gamma \vdash^{e}_{t} t : (\Pi x : A.B) \text{ and } \Delta \vdash^{e}_{t} t' : (\Pi x : A'.B')$$

  we deduce $(\Pi x : A.B) =_{\mathbf{T}(mixt)} (\Pi x : A'.B')$. By confluence, $(\Pi x : A.B) \downarrow_{\mathbf{T}(mixt)} (\Pi x : A'.B')$. Hence $A \downarrow_{mixt} A'$ and $B \downarrow_{mixt} B'$. By Correctness of Types, $\Gamma \vdash^{e}_{t} \Pi x : A.B : s$ and $\Delta \vdash^{e}_{t} \Pi x : A'.B' : s'$ and hence $\Pi x : A.B$ and $\Pi x : A'.B'$ are strongly normalising. So $A^{\mathsf{nf}} \equiv A'^{\mathsf{nf}}$ and $B^{\mathsf{nf}} \equiv B'^{\mathsf{nf}}$, and we are done.
- *abstraction:* assume the last step is

$$\frac{\Gamma, x : A \vdash^e_t t : B \quad \Gamma \vdash^e_t \Pi x : A.B : s}{\Gamma \vdash^e_t \lambda^{\Pi x:A.B} x.t : \Pi x : A.B}$$

with $M \equiv \lambda^{\Pi x:A.B} x.t$. 1 is easy to prove. As for 2, assume $N \equiv \lambda^{\Pi x:A'.B'} x.t'$, $||\Gamma|| \equiv ||\Delta||$ and $||M|| \equiv ||N||$. To show $M^{\mathsf{can}} \equiv N^{\mathsf{can}}$, i.e.

$$\lambda^{\Pi x:A^{\mathsf{can}}.B^{\mathsf{nf}}} x.t^{\mathsf{can}} \equiv \lambda^{\Pi x:A'^{\mathsf{can}}.B'^{\mathsf{nf}}} x.t'^{\mathsf{can}}$$

By $\mathbf{G}_\lambda$, $\Delta, x : A' \vdash^e_t t' : B'$. Note that $||M|| \equiv ||N||$ implies that $||A|| \equiv ||A'||$, hence $||\Gamma, x : A|| \equiv ||\Delta, x : A'||$. We can use the induction hypothesis on $\Gamma, x : A \vdash^e_t t : B$ and deduce $t^{\mathsf{can}} \equiv t'^{\mathsf{can}}$ and $A^{\mathsf{can}} \equiv A'^{\mathsf{can}}$.
We are left to show $B^{\mathsf{nf}} \equiv B'^{\mathsf{nf}}$. By corollary 33, $B =_{\mathbf{T}(mixt)} B'$. By confluence, $B \downarrow_{\mathbf{T}(mixt)} B'$. By Correctness of Types, either B is a sort or $\Gamma, x : A \vdash^e_t B : s_0$. Similarly, either B' is a sort or $\Delta, x : A' \vdash^e_t B' : s_1$. In all cases both B and B' are strongly normalising. Hence $B^{\mathsf{nf}} \equiv B'^{\mathsf{nf}}$ and we are done.

- *weakening:* assume the last step is

$$\frac{\Gamma \vdash^e_t M : B \quad \Gamma \vdash^e_t A : s}{\Gamma, x : A \vdash^e_t M : B}$$

with M a variable or a sort or an universe. Assume $\Delta, x : A'$ is a legal context with $||\Gamma, x : A|| \equiv ||\Delta, x : A'||$. Then $||\Gamma|| \equiv ||\Delta||$ and $||A|| \equiv ||A'||$. Necessarily, $\Delta \vdash^e_t A' : s'$ so we may apply the induction hypothesis on $\Gamma \vdash^e_t A : s$ to conclude $\Delta^{\mathsf{can}} \equiv \Gamma^{\mathsf{can}}$ and $A^{\mathsf{can}} \equiv A'^{\mathsf{can}}$. This proves 1. As for 2, assume $||N|| \equiv ||M||$. Then $N \equiv M$ because M a variable or a sort or an universe. So we are done.

■

Proof of Lemma 36

– first note that it is not true for an arbitrary M because algebraic rewrite rules might not be left-linear. Indeed, consider the rewrite rule

$$f(x, x) \to x$$

If A and A' have no common reduct, then the term

$$f(\mathsf{app}^{\Pi x:A.B}(x, y), \mathsf{app}^{\Pi x:A'.B}(x, y))$$

is in normal form while we have the reduction

$$||f(\mathsf{app}^{\Pi x:A.B}(x, y), \mathsf{app}^{\Pi x:A'.B}(x, y))|| \equiv f(x\ y, x\ y) \to x\ y$$

The lemma is proved by structural induction on the derivation of $\Gamma \vdash^e_t M : A$. We treat the cases where the last rule of the derivation is (function) or (application).

- *function:* then $M \equiv f(t_1, \dots, t_n)$. The only interesting case is when $||M||$ itself is a redex, i.e. when there exists a rule $l \to r$ and a substitution θ with domain $\mathsf{FV}(l)$ such that $\theta l \equiv f(||t_1||, \dots, ||t_n||)$ and $\theta r \equiv N$. Take l_0 linear with $\mathsf{FV}(l) \cap \mathsf{FV}(l_0) = \emptyset$ and ρ a renaming with domain $\mathsf{FV}(l_0)$ (it may rename two distinct variables with the same name) such that $\rho l_0 \equiv l$. There exists a labelled substitution θ' with domain $\mathsf{FV}(l_0)$ such that $\theta' l_0 \equiv M$. We know that for every $x \in \mathsf{FV}(l_0)$, we have $||\theta' x|| \equiv \theta \circ \rho(x)$. Hence for every $x, y \in \mathsf{FV}(l_0)$, $\rho x \equiv \rho y \Rightarrow ||\theta' x|| \equiv ||\theta' y||$. By Lemma 35, it follows $(\theta' x)^{\mathsf{can}} \equiv (\theta' y)^{\mathsf{can}}$. Define a labelled substitution θ'' with domain $\mathsf{FV}(l_0)$ by $\theta'' x \equiv (\theta' x)^{\mathsf{can}}$. There exists a substitution θ_0 with domain $\mathsf{FV}(l)$ such that $\theta''(x) \equiv \theta_0 \circ \rho(x)$ for every $x \in \mathsf{FV}(l_0)$. Define $N' \equiv \theta_0 r$. Then $M \twoheadrightarrow_{mixl} N'$.

  To show $||N'|| \equiv N$. Let $x \in \mathsf{FV}(l)$. There exists y such that $\rho y \equiv x$. We have

  $$||\theta_0 x|| \equiv ||\theta_0(\rho y)|| \equiv ||\theta'' y|| \equiv ||(\theta' y)^{\mathsf{can}}|| \equiv ||\theta' y|| \equiv \theta(\rho y) \equiv \theta x$$

  Hence $||\theta_0 x|| \equiv \theta x$ for every $x \in \mathsf{FV}(l)$ and we are done.
- *application:* let $M \equiv \mathsf{app}^{\Pi x:C.D}(t, u)$ and $||M|| \equiv ||t|| ||u|| \to_{mix} N'$. We use the induction hypothesis if the reduction occurs in $||t||$ or $||u||$. When $||M||$ itself is a β-redex then $t \equiv \lambda^{\Pi x:C'.D'} x.t'$ and $N \equiv ||t'||[||u||/x]$. The loose head reduction of M leads to $N' \equiv t'[u/x]$. We are done with the following equality:

  $$||N'|| \equiv \ ||t'[u/x]|| \equiv ||t'||[||u||/x] \ \equiv N$$

- it is proved by induction on the length of the reduction sequence $||M|| \twoheadrightarrow^+_{mix} N$. Assume $||M|| \to_{mix} P$. By 1. there exists P' such that $M \twoheadrightarrow^+_{mixl} P'$ and $||P'|| \equiv P$. By Lemma 27, $M \twoheadrightarrow^+_{mixt} P'$. By Subject Reduction, P' is legal. So we can apply the induction hypothesis on P'.
- the strong normalisation part is proved by induction on the length of the longest $\to_{mixt}$-reduction sequence starting from M. Assume that $||M|| \to_{mix} N$. By 2 there exists N' such that $M \twoheadrightarrow^+_{mixt} N'$ and $||N'|| \equiv N$. By Subject Reduction, N' is legal so we can apply the induction hypothesis. Hence $\to_{mix}$ is strongly normalising on $N \equiv ||N'||$. The property is true for any N such that $||M|| \to_{mix} N$. Henceforth $\to_{mix}$ is strongly normalising on $||M||$. As for the Church-Rosser property, assume $||M|| \twoheadrightarrow_{mix} N_1$ and $||M|| \twoheadrightarrow_{mix} N_2$. Then there exist N_1' and N_2' such that $M \twoheadrightarrow_{mixt} N_1'$ and $M \twoheadrightarrow_{mixt} N_2'$ with $||N_1'|| \equiv N_1$ and $||N_2'|| \equiv N_2$. By confluence of $\to_{mixt}$, there exists a labelled pseudo-term P with $N_1' \twoheadrightarrow_{mixt} P$ and $N_2' \twoheadrightarrow_{mixt} P$. We can translate $N_i' \twoheadrightarrow_{mixt} P$ into $||N_i'|| \twoheadrightarrow_{mix} ||P||$. It follows that $N_1 \twoheadrightarrow_{mix} ||P||$ and $N_2 \twoheadrightarrow_{mix} ||P||$.
- assume $||M|| \downarrow_{mix} ||N||$. Hence there exists Q such that $||M||, ||N|| \twoheadrightarrow_{mix} Q$. By 2, there exists Q_1 and Q_2 such that $M \twoheadrightarrow_{mixt} Q_1$ and $M \twoheadrightarrow_{mixt} Q_2$. Moreover $||Q_1|| \equiv ||Q_2|| \equiv Q$. Hence $Q_1^{\mathsf{can}} \equiv Q_2^{\mathsf{can}} \equiv Q'$. Thus we have $M, N \twoheadrightarrow_{mixt} Q'$ i.e. $M \downarrow_{mixt} N$.

$\blacksquare$

Proof of Proposition 37 the proof proceeds by structural induction on the derivation of $\Gamma \vdash M : A$.

- *conversion:* suppose that $\Gamma \vdash M : B$ is derived from $\Gamma \vdash M : A$ and $\Gamma \vdash B : s$ with $A \downarrow_{mix} B$. It follows by induction that $\Gamma_\bullet \vdash^e_t M_\bullet : A_\bullet$ and $\Gamma_\circ \vdash^e_t B_\circ : s$ with

$$||\Gamma_\bullet|| \equiv \Gamma \equiv ||\Gamma_\circ|| \quad ||M_\bullet|| \equiv M \quad ||A_\bullet|| \equiv A \quad ||B_\circ|| \equiv B$$

By lemma 35, $\Gamma_\bullet \equiv \Gamma_\circ$. By Correctness of Types, $\Gamma_\bullet \vdash^e_t A_\bullet : s'$ for a given universe s'. The last statement of Lemma 36 deduces $A \downarrow_{mixt} B$ from $\Gamma_\bullet \equiv \Gamma_\circ$ and $||A|| \downarrow_{mix} ||B||$. By conversion

$$\Gamma_\bullet \vdash M_\bullet : B_\circ$$

with the required translation (and canonicity) features:

$$||\Gamma_\bullet|| \equiv \Gamma \text{ and } ||M_\bullet|| \equiv M \text{ and } ||B_\circ|| \equiv N$$

- *application:* suppose that $\Gamma \vdash tu : B[u/x]$ is derived from $\Gamma \vdash t : (\Pi x : A.B)$ and $\Gamma \vdash u : A$ with application rule. By induction there exists two derivable judgements

$$\Gamma_\bullet \vdash^e_t t_\bullet : (\Pi x : A.B)_\bullet \text{ and } \Gamma_\circ \vdash^e_t u_\circ : A_\circ \quad (1)$$

with the good translation properties. Define $A_\bullet, B_\bullet$ as $(\Pi x : A.B)_\bullet \equiv \Pi x : A_\bullet.B_\bullet$; by G_Π and Correctness of Types, $\Gamma_\bullet \vdash^e_t A_\bullet : s'$ and $\Gamma_\circ \vdash^e_t A_\circ : s''$ for some universes s', s''. By lemma 35

$$\Gamma_\bullet \equiv \Gamma_\circ \text{ and } A_\bullet \equiv A_\circ$$

because $A_\circ$ is canonical. By (1) and an application rule it follows that:

$$\Gamma_\bullet \vdash^e_t \mathsf{app}^{\Pi x : A_\bullet.B_\bullet}(t_\bullet, u_\circ) : B_\bullet[u_\circ/x]$$

By the Subject Reduction Property and conversion:

$$\Gamma_\bullet \vdash^e_t \mathsf{app}^{\Pi x : A_\bullet^{\mathsf{nf}}.B_\bullet^{\mathsf{nf}}}(t_\bullet, u_\circ) : (B_\bullet[u_\circ/x])^{\mathsf{can}}$$

We easily check the three equalities

$$||\Gamma_\bullet|| \equiv \Gamma \text{ and } ||\mathsf{app}^{\Pi x : A_\bullet^{\mathsf{nf}}.B_\bullet^{\mathsf{nf}}}(t_\bullet, u_\circ)|| \equiv tu \text{ and } ||(B_\bullet[u_\circ/x])^{\mathsf{can}}|| \equiv B[u/x]$$

and the canonicity of $\Gamma_\bullet$, $\mathsf{app}^{\Pi x : A_\bullet^{\mathsf{nf}}.B_\bullet^{\mathsf{nf}}}(t_\bullet, u_\circ)$ and $(B_\bullet[u_\circ/x])^{\mathsf{can}}$.

- *abstraction:* assume the last step is

$$\frac{\Gamma, x : A \vdash t : B \qquad \Gamma \vdash \Pi x : A.B : s}{\Gamma \vdash \lambda x : A.t : \Pi x : A.B}$$

and $M \equiv \lambda x : A.t$. By induction hypothesis:

$$\Gamma_\bullet, x : A_\bullet \vdash^e_t t_\bullet : B_\bullet \qquad \Gamma_\circ \vdash^e_t (\Pi x : A.B)_\circ : s \qquad (2)$$

for some canonical contexts $\Gamma_\bullet, \Gamma_\circ$. By Correctness of Types, $\Gamma_\bullet, x : A_\bullet \vdash^e_t B_\bullet : s'''$ for some universe s'. Let $A_\circ$ and $B_\circ$ be defined as $(\Pi x : A.B)_\circ \equiv \Pi x : A_\circ.B_\circ$. By G_Π:

$$\Gamma_\circ \vdash^e_t A_\circ : s_1 \qquad \Gamma_\circ, x : A_\circ \vdash^e_t B_\circ : s_2$$

for some universes s_1, s_2. By lemma 35

$$\Gamma_\bullet \equiv \Gamma_\circ \quad A_\bullet \equiv A_\circ \quad B_\bullet \equiv B_\circ$$

By abstraction and (2):

$$\Gamma_\bullet \vdash^e_t \lambda^{\Pi x:A_\bullet.B_\bullet} x.t_\bullet : \Pi x : A_\bullet.B_\bullet$$

By the Subject Reduction Property and a conversion rule:

$$\Gamma_\bullet \vdash^e_t \lambda^{\Pi x:A_\bullet^{\mathsf{can}}.B_\bullet^{\mathsf{nf}}} x.t_\bullet : (\Pi x : A_\bullet.B_\bullet)^{\mathsf{can}}$$

We easily check the three equalities

$$||\Gamma_\bullet|| \equiv \Gamma \quad ||\lambda^{\Pi x:A_\bullet^{\mathsf{can}}.B_\bullet^{\mathsf{nf}}} x.t_\bullet|| \equiv \lambda x : A.t \quad ||(\Pi x : A_\bullet.B_\bullet)^{\mathsf{can}}|| \equiv \Pi x : A.B$$

and the respective canonicity properties.

- *weakening:* assume the last step is:

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash B : s}{\Gamma, x : B \vdash t : A} \text{ if } x \notin \Gamma \text{ and } t \in S \cup U \cup V \cup K$$

By induction there exists $\Gamma_\bullet \vdash^e_t t_\bullet : A_\bullet$ and $\Gamma_\circ \vdash^e_t B_\circ : s$. By lemma 35 $\Gamma_\circ \equiv \Gamma_\bullet$ and henceforth $\Gamma_\bullet, x : B_\circ \vdash^e_t t_\bullet : A_\bullet$ since $x \notin \Gamma$ and $t \in S \cup U \cup V \cup K$.

■