Termination of Algebraic Type Systems: The Syntactic Approach

Gilles Barthe and Femke van Raamsdonk *

CWI
P.O. Box 94079, 1090 GB Amsterdam
The Netherlands
{gilles,femke}@cwi.nl

1 Introduction

Combinations of type theory and rewriting are of obvious interest for the study of higher-order programming and program transformation with algebraic data types specifications; somewhat more recently, they also found applications in proof-checking. A natural question in this field concerns the termination or strong normalisation of such systems and is as follows: given a terminating type system T and a terminating rewriting system R, is the combination of T and R terminating? It is not surprising that this question has already received considerable attention, see for instance [1, 2, 4, 7, 11, 12, 15–17, 19, 24]. However, the situation is in our opinion not yet satisfactory, since most of the proofs of termination of a combination of a type theory and a rewriting system consist basically in redoing the proof of termination of the type theory. Ideally, one would like to have a modular *proof* of these modularity *results*, i.e. a proof that *uses but does not re-prove* the facts that the type theory and the term rewriting system are terminating.

The question we embark on is hence to develop general methods that permit to derive termination of the combination of a type theory and a rewriting system from termination of those systems separately. We make the question precise in the framework of algebraic type systems [7] which combine pure type systems and term rewriting systems. The advantage of this setting is its generality; for instance the combination of the calculus of constructions with a term rewriting system, as defined e.g. in [4], is an algebraic type system.

The first method we present is called *termination by translation.* An algebraic type system $\mathcal{A}$ is terminating if there exists a map into a terminating algebraic type system $\mathcal{A}'$ such that derivable judgements in $\mathcal{A}$ are mapped to derivable judgements in $\mathcal{A}'$ and rewrite steps in $\mathcal{A}$ are mapped to non-empty rewrite sequences in $\mathcal{A}'$. This technique, which is well-known in type theory, is an elaboration of termination by translation in first-order term rewriting. Despite its extreme simplicity, it permits to obtain, in a very easy way, useful termination results for algebraic type systems.

* A part of this work has been carried out while the second author was at INRIA Sophia Antipolis, France, on a grant of the HCM Cooperation Network EXPRESS.

The second method, which we call *termination by stability*, has not only useful applications but is also interesting in itself. The method is inspired from [16] where Dougherty considers untyped λ-calculus with β-reduction in combination with a first-order, single-sorted, term rewriting system $\mathcal{R}$. In particular, Dougherty shows that the union of $\rightarrow_\beta$ and $\rightarrow_\mathcal{R}$ is terminating on a suitably defined subset $\mathsf{Stable}(\mathcal{R})$ of the set of β-strongly normalising terms. Elements of $\mathsf{Stable}(\mathcal{R})$ are called stable terms, after which the method is named. Our interest in Dougherty's method lies in the fact that termination of the combined reduction relation is *reduced* to termination of its components, i.e. β-reduction and $\mathcal{R}$-reduction. In Section 3, we extend Dougherty's method to algebraic type systems.

In Section 5, we apply the two methods to find easy proofs of well-known results. For example, we give easy proofs of termination for the combination of higher-order λ-calculus $\lambda\omega$ and a terminating term rewriting system. Furthermore, we derive from the two methods a number of new results:

- The combination of higher-order λ-calculus, $\lambda\omega$, with $\beta\eta$-reduction and a terminating term rewriting system is terminating.
- Under certain mild conditions, the combination of a terminating pure type system and a terminating non-collapsing term rewriting system is terminating. As a corollary, we obtain that the combination of a pure type system and a ground rewriting system is terminating, under some mild conditions.

Our methods are flexible and robust. Firstly, they apply to several notions of reductions such as β-reduction, η-reduction, and, we claim, η-expansion. Secondly, they carry over to variations of algebraic type systems such as type-assignment systems, domain-free pure type systems or pure type systems with Π-conversion. Thirdly, the method scales up when other type constructions, such as the ones for products and sums, are considered.

Moreover, our methods are simple. They can be carried out in a weak system of arithmetic. This is the case since we *reduce* termination of an algebraic type system to that of a pure type system and *do not prove* the latter, which of course may require a strong system of arithmetic.

Finally, our methods are informative. In particular, they shed some light on the logical status of algebraic type systems. For example, a simple application of derivation-preserving translations shows that the internal logic of an algebraic type system cannot distinguish between two distinct closed algebraic terms.

Related work. The problem of termination for combinations of λ-calculus and term rewriting systems has already received considerable attention in higher-order rewriting but here we limit ourselves to termination results for algebraic type systems.

One of the first termination results for algebraic type systems is due to Breazu-Tannen and Gallier. In [12], they prove that the combination of the polymorphic λ-calculus –system F– with the curried version of a terminating first-order term rewriting system is terminating. The proof makes use of the

'candidats de reductibilité' and, for an essential part, consists in redoing the proof of termination for system F.

The combination of polymorphic λ-calculus with higher-order term rewriting is studied by Jouannaud and Okada in [19]. Using a computability argument, they show that provided some conditions on the form of the rewrite rules are satisfied, the combination of the polymorphic λ-calculus and a terminating higher-order rewriting system is again terminating. This result is generalized by Barbanera, Fernández and Geuvers in a series of papers first to intersection type systems, then to higher-order λ-calculus and finally to the so-called algebraic λ-cube [1, 3, 4]. Termination of the algebraic λ-cube is proved along the lines of [18] by using a computability predicate and two reduction preserving translations.

In [7], Barthe and Geuvers introduce the notion of algebraic type system and provide a general criterion for termination of an algebraic type system. The criterion is proved by a model construction based on saturated sets –so it re-does the termination proof for the corresponding pure type system. Unfortunately, the criterion requires the algebraic type system to have the subject reduction property, a severe restriction in the current state of knowledge. The problem is partially overcome in [9] where Barthe and Melliès use a labelled syntax to prove termination and subject reduction of algebraic type systems. However, the approach is complicated and requires to redo the proof of termination of the underlying pure type system.

Using a completely different approach, van de Pol shows in [24] termination of the combination of simply typed λ-calculus and the curried version of a terminating first-order term rewriting system. This result is obtained in the framework of higher-order rewriting systems, so in particular simply typed λ-calculus is coded as a higher-order rewriting system. The proof consists of extending a termination model of the term rewriting system to a termination model for simply typed λ-calculus.

Finally, several authors have recently considered algebraic type systems with η-expansion [14, 15]. In a nutshell, two techniques are used to prove termination: reducibility candidates and simulation of η-expansion. The first one is not modular, since it involves doing the termination proof again. The second one consists in defining a translation from legal terms to legal terms so that every infinite reduction sequence with η-expansions is translated into an infinite reduction sequence without η-expansions. This approach is in a sense orthogonal to ours as we translate an infinite reduction with algebraic reduction steps into an infinite reduction sequence without algebraic reduction steps. It turns out that our approach yields shorter and conceptually simpler proofs.

Organisation of the paper. The paper is organised as follows: in Section 2, we introduce the framework of algebraic type systems. Section 3 presents the techniques of termination by translation and termination by stability. In Section 4 we consider algebraic rewriting of algebraic pure type systems. Section 5 contains several applications of the two techniques, yielding new proofs of old results as well as new results. We conclude in Section 6.

2 Algebraic Type Systems

In this section we present the definition of an algebraic type system as the combination of a pure type system and a typed term rewriting system. First we recall the definition of a pure type system and give a suitable definition of a typed term rewriting system.

2.1 Pure Type Systems

Pure type systems were introduced by Berardi and by Terlouw as a general framework to define and study typed λ-calculi. The definition we present is a slight modification of the one given in [5].

Definition 1. A pure type system $\mathcal{S}$ is specified by a triple (U, TA, TR) with

1. U a set of *universes*,
2. $TA \subseteq U \times U$ a set of *typing axioms*,
3. $TR \subseteq U \times U \times U$ is a set of *typing rules*.

We assume a set V of *variables*, written as $x, y, z, \ldots$. The set of *pseudo-terms* of a pure type system $\mathcal{S} = (U, TA, TR)$ is defined by

$$T ::= V \mid U \mid \Pi V : T.T \mid \lambda V : T.T \mid TT$$

An *environment* is an ordered list of pairs of the form $x : A$ with $x \in V$ and $A \in T$. A *judgement* is a triple of the form $\Gamma \vdash M : A$ with Γ an environment and M, A pseudo-terms. Intuitively, a judgement assigns in a given environment a type to a term. The meaningful judgements of a pure type system are defined by means of a set of rules. One of these rules, the conversion rule, makes use of a rewrite relation on the set of pseudo-terms of the pure type system. For the sake of uniformity, we present the rules parametrised over the rewrite relation used in the conversion rule. A judgement then takes the form $\Gamma \vdash_{\mathcal{C}} M : A$, where $\mathcal{C}$ is the rewrite relation used in the conversion rule.

Definition 2. Let $\rightarrow_{\mathcal{C}}$ be a rewrite relation on the set of pseudo-terms of a pure type system $\mathcal{S} = (U, TA, TR)$.

1. A judgement $\Gamma \vdash_{\mathcal{C}} M : A$ is said to be *derivable* if it can be derived using the rules given in Table 1.
2. A pseudo-term M is said to be *$\mathcal{C}$-legal* if for some Γ and some A the judgement $\Gamma \vdash_{\mathcal{C}} M : A$ is derivable. The set of $\mathcal{C}$-legal terms of the pure type system $\mathcal{S}$ is denoted by $\mathcal{L}(\mathcal{S}, \mathcal{C})$.

If we simply state $\Gamma \vdash_{\mathcal{C}} M : A$, we mean that $\Gamma \vdash_{\mathcal{C}} M : A$ is a derivable judgement.

The most important rewrite relation on the set of pseudo-terms of a pure type system is the β-reduction relation, denoted by $\rightarrow_\beta$, which is defined as the compatible closure of

$$(\lambda x : A.M)N \rightarrow M[x := N]$$

axiom	$\dfrac{}{\vdash_{\mathcal{C}} c : s}$	if $(c, s) \in TA$
start	$\dfrac{\Gamma \vdash_{\mathcal{C}} A : s}{\Gamma, x : A \vdash_{\mathcal{C}} x : A}$	if $x \notin \Gamma$, $x \in V$
weakening	$\dfrac{\Gamma \vdash_{\mathcal{C}} M : A \quad \Gamma \vdash_{\mathcal{C}} B : s}{\Gamma, x : B \vdash_{\mathcal{C}} M : A}$	if $x \notin \Gamma$, $x \in V$
product	$\dfrac{\Gamma \vdash_{\mathcal{C}} A : s_1 \quad \Gamma, x : A \vdash_{\mathcal{C}} B : s_2}{\Gamma \vdash_{\mathcal{C}} \Pi x : A.B : s_3}$	if $(s_1, s_2, s_3) \in TR$
application	$\dfrac{\Gamma \vdash_{\mathcal{C}} M : \Pi x : A.B \quad \Gamma \vdash_{\mathcal{C}} N : A}{\Gamma \vdash_{\mathcal{C}} M\ N : B[x := M]}$	
abstraction	$\dfrac{\Gamma, x : A \vdash_{\mathcal{C}} M : B \quad \Gamma \vdash_{\mathcal{C}} (\Pi x : A.B) : s}{\Gamma \vdash_{\mathcal{C}} \lambda x : A.M : \Pi x : A.B}$	
conversion	$\dfrac{\Gamma \vdash_{\mathcal{C}} u : A \quad \Gamma \vdash_{\mathcal{C}} B : s}{\Gamma \vdash_{\mathcal{C}} u : B}$	if $A \downarrow_{\mathcal{C}} B$

Table 1. PURE TYPE SYSTEMS

Often one considers the set of β-legal terms of a pure type system, equipped with the β-reduction relation, that is, the rewriting system $(\mathcal{L}(\mathcal{S}, \beta), \rightarrow_\beta)$.

An important example of pure type system is the calculus of constructions, defined by Coquand and Huet in [13]. Barendregt presents in [5] a fine-grain analysis of the calculus of constructions in terms of the λ-cube, a cube consisting of eight pure type systems. They all have $\{\star, \Box\}$ as the set of universes, and $\star : \Box$ as only typing axiom. Their sets of typing rules are as follows (here (s_1, s_2) abbreviates (s_1, s_2, s_2)):

$$\begin{array}{ll} \lambda{\rightarrow} : (\star,\star) & \lambda P \ : (\star,\star),(\star,\Box) \\ \lambda 2 \ : (\star,\star),(\Box,\star) & \lambda P2 : (\star,\star),(\Box,\star),(\star,\Box) \\ \lambda\underline{\omega} \ : (\star,\star),(\Box,\Box) & \lambda P\underline{\omega} : (\star,\star),(\Box,\Box),(\star,\Box) \\ \lambda\omega \ : (\star,\star),(\Box,\star),(\Box,\Box) & \lambda P\omega : (\star,\star),(\Box,\star),(\Box,\Box),(\star,\Box) \end{array}$$

Most of the systems of the λ-cube are of independent interest and appear in the literature, often in a variant form, see [5] for references. The calculus of constructions, $\lambda P\omega$, is the most complex system of the cube.

Morphisms of Pure Type Systems. Morphisms of pure type systems are maps between the sets of universes which preserve typing axioms and typing rules.

Definition 3. Let $\mathcal{A} = (U, TA, TR)$ and $\mathcal{A}' = (U', TA', TR')$ be pure type systems. A pure type system morphism between $\mathcal{A}$ and $\mathcal{A}'$ is a mapping

$$\phi : U \to U'$$

such that:

1. if $(u_1, u_2) \in TA$, then $(\phi(u_1), \phi(u_2)) \in TA'$,
2. if $(u_1, u_2, u_3) \in TR$, then $(\phi(u_1), \phi(u_2), \phi(u_3)) \in TR'$.

For a pure type morphism ϕ as in the previous definition, we write $\phi : \mathcal{A} \to \mathcal{A}'$ by a slight abuse of notation, also to denote the homomorphic extension of ϕ mapping pseudo-terms of $\mathcal{S}$ to pseudo-terms of $\mathcal{S}'$. A morphism as defined in this way is only concerned with the signature of a pure type system. If one is interested in a morphism that maps $\mathcal{C}$-legal terms in $\mathcal{S}$ to $\mathcal{C}'$-legal terms in $\mathcal{S}'$, then we should require in addition that $M \downarrow_{\mathcal{C}} N$ in $\mathcal{S}$ implies $\phi(M) \downarrow_{\mathcal{C}'} \phi(N)$ in $\mathcal{S}'$. Then it follows that $\mathcal{C}$-legal terms are mapped to $\mathcal{C}'$-legal terms by a straightforward induction on derivations.

2.2 Typed Term Rewriting Systems

In this subsection, we define typed term rewriting systems in such a way that they can be conveniently combined with pure type systems. We use the following notion of sorted signature.

Definition 4.

1. Let S be a set. An *algebraic type over* S is an expression of the form

$$s_1 \times \ldots \times s_n \to s$$

   with $n \geq 0$ and $s_1, \ldots, s_n, s \in S$. We write s instead of $\to s$.
2. A *sorted signature* is a pair (S, F) consisting of a set of sorts, written as $s, s', \ldots$, and a set of function symbols, written as $f, g, \ldots$, such that every function symbol $f \in F$ has a unique algebraic type over S.

Note that we simply assume every function symbol to have an algebraic type; we don't consider explicitly a function assigning algebraic types to function symbols. If a function symbol f has an algebraic type $s_1 \times \ldots \times s_n \to s$, then n is said to be the *arity* of f. The arity of a symbol f is denoted by $\mathsf{ar}(f)$.

In order to define terms, we assume a set V of *variables*, written as $x, y, z, \ldots$. We define an *environment* as an ordered list of type declarations of the form $x : s$, with $x \in V$ and s a sort. We say that a variable x is *declared* in Γ if $x : s \in \Gamma$ for some s and we assume that variables are declared at most once.

Definition 5. An expression M is a *typed algebraic term* (also simply called a *term*) over a sorted signature (S, F) if $\Gamma \vdash M : s$ is derivable for some environment Γ and sort s, using the rules given in Table 2.

$$\frac{}{\Gamma, x : s \vdash x : s} \quad \text{if } s \in S$$

$$\frac{\Gamma \vdash M_1 : s_1 \quad \ldots \quad \Gamma \vdash M_n : s_n}{\Gamma \vdash f(M_1, \ldots, M_n) : s} \quad \text{if } f : s_1 \times \ldots \times s_n \to s$$

Table 2. TYPED ALGEBRAIC TERMS OVER (S, F)

Rewrite rules of a typed term rewriting system are defined as follows. Here $var(M)$ denotes the set of variables occurring in M.

Definition 6. Let (S, F) be a sorted signature. A *rewrite rule over* (S, F) is a pair of terms over (S, F), written as $l \to r$, such that

1. for every environment Γ, we have $\Gamma \vdash l : A$ implies $\Gamma \vdash r : A$,
2. $l \notin V$,
3. $var(r) \subseteq var(l)$.

Now we have collected all ingredients for the definition of a typed term rewriting system.

Definition 7. A *typed term rewriting system* $\mathcal{R}$ is specified by a pair $((S, F), R)$ with

1. (S, F) a sorted signature,
2. R a set of rewrite rules over (S, F).

The *rewrite relation* $\to_{\mathcal{R}}$ of a typed term rewriting system $\mathcal{R} = ((S, F), R)$ is defined as follows. We have $M \to_{\mathcal{R}} N$ if M is a typed algebraic term and there is a context $C[\bullet]$, a substitution θ and a rewrite rule $l \to r \in R$ such that $M = C[l\theta]$ and $N = C[r\theta]$. Here a context is a term with a unique occurrence of a special constant $\bullet$, and $C[M]$ denotes the result of replacing $\bullet$ in $C[\bullet]$ by M. Substitutions are supposed to preserve the typing.

We will assume the reader to be familiar with well-known properties of (untyped) term rewriting system, which can be found in [22, 23]. In the following, we will often simply say 'term rewriting system' instead of 'typed term rewriting system'.

Morphisms of Sorted Signatures. In the sequel, we shall use mappings that preserve the structure of sorted signatures. These mappings are defined as follows.

Definition 8. Let (S, F) and (S', F') be sorted signatures. A sorted signature morphism between (S, F) and (S', F') is a pair of mappings $\psi = (\psi_1, \psi_2)$ with

$$\psi_1 : S \to S'$$
$$\psi_2 : F \to F'$$

such that for every $f : s_1 \times \ldots \times s_n \to s$ in (S, F), we have $\psi_2(f) : \psi_1(s_1) \times \ldots \times \psi_1(s_n) \to \psi_1(s)$ in (S', R').

2.3 Algebraic Type Systems

The definition we present in this subsection is equivalent to the one given in [9] and inspired from [4, 7]. An algebraic type system is a combination of a pure type system and a typed term rewriting system. In order to define the combination, we need to specify how sorts are embedded into universes. This is the purpose of the embedding axioms EA below.

Definition 9. An *algebraic type system* is specified by a triple $\mathcal{S} +_{EA} \mathcal{R}$ consisting of

1. a pure type system $\mathcal{S} = (U, TA, TR)$,
2. a typed term rewriting system $\mathcal{R} = ((S, F), R)$,
3. a set of embedding axioms $EA \subseteq S \times U$ such that for every $s \in S$ there exists a $u \in U$ such that $(s, u) \in EA$.

In the sequel, we let $\mathsf{codom}(EA)$ denote the set of $u \in U$ such that there exists an $s \in S$ with $(s, u) \in EA$. Usually, e.g. in the algebraic λ-cube, $\mathsf{codom}(EA)$ is a singleton, but it is not necessarily desirable that all sorts live in the same universe. For instance, one could have a typed term rewriting system with sorts nat for natural numbers and ord for ordinals, and relate them to universes set and class in a pure type system by declaring nat : set and ord : class.

In the remainder of this subsection consider an algebraic type system $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ with $\mathcal{S} = (U, TA, TR)$ and $\mathcal{R} = ((S, F), R)$.

Definition 10.

1. The set T of *pseudo-terms* of $\mathcal{A}$ is defined as follows:
$$T ::= V \mid U \mid S \mid \Pi V : T.T \mid \lambda V : T.T \mid TT \mid f(T, \ldots, T)$$
where $f \in F$ and the arity of f is respected.
2. The rules used to form derivable judgements of $\mathcal{A}$ are parametrised over a rewrite relation $\rightarrow_{\mathcal{C}}$ on the set of pseudo-terms of $\mathcal{A}$. They are given in Table 3.
3. A pseudo-term M is said to be a *$\mathcal{C}$-legal term* if there exist an environment Γ and a pseudo-term A such that $\Gamma \vdash_{\mathcal{C}} M : A$ is a derivable judgement. The set of $\mathcal{C}$-legal terms is denoted by $\mathcal{L}(\mathcal{A}, \mathcal{C})$.

Traditionally, the rewrite relation considered for algebraic type systems is the union of β-reduction with algebraic reduction. It is defined as follows.

Definition 11.

1. The β-rewrite relation $\rightarrow_\beta$ is defined as the compatible closure of
$$(\lambda x : A.M)N \rightarrow M[x := N]$$
2. The algebraic rewrite relation $\rightarrow_{\mathcal{R}}$ is defined by $M \rightarrow_{\mathcal{R}} N$ if there exists a context $C[\bullet]$, a substitution θ and a rewrite rule $l \rightarrow r$ such that $M = C[l\theta]$ and $N = C[r\theta]$.

axiom	$\dfrac{}{\vdash_{\mathcal{C}} s : s'}$	if $(s, s') \in TA \cup EA$
start	$\dfrac{\Gamma \vdash_{\mathcal{C}} A : s}{\Gamma, x : A \vdash_{\mathcal{C}} x : A}$	if $x \notin \Gamma$, $x \in V$
weakening	$\dfrac{\Gamma \vdash_{\mathcal{C}} t : A \quad \Gamma \vdash_{\mathcal{C}} B : s}{\Gamma, x : B \vdash_{\mathcal{C}} t : A}$	if $x \notin \Gamma$, $x \in V$
product	$\dfrac{\Gamma \vdash_{\mathcal{C}} A : s_1 \quad \Gamma, x : A \vdash_{\mathcal{C}} B : s_2}{\Gamma \vdash_{\mathcal{C}} \Pi x : A.B : s_3}$	if $(s_1, s_2, s_3) \in TR$
application	$\dfrac{\Gamma \vdash_{\mathcal{C}} t : \Pi x : A.B \quad \Gamma \vdash_{\mathcal{C}} u : A}{\Gamma \vdash_{\mathcal{C}} t\ u : B[u/x]}$	
abstraction	$\dfrac{\Gamma, x : A \vdash_{\mathcal{C}} t : B \quad \Gamma \vdash_{\mathcal{C}} (\Pi x : A.B) : s}{\Gamma \vdash_{\mathcal{C}} \lambda x : A.t : \Pi x : A.B}$	
function	$\dfrac{\Gamma \vdash_{\mathcal{C}} M_1 : \sigma_1 \quad \ldots \quad \Gamma \vdash_{\mathcal{C}} M_n : \sigma_n}{\Gamma \vdash_{\mathcal{C}} f(M_1, \ldots, M_n) : \tau}$	if $f : \sigma_1 \times \ldots \times \sigma_n \rightarrow \tau$
conversion	$\dfrac{\Gamma \vdash_{\mathcal{C}} u : A \quad \Gamma \vdash_{\mathcal{C}} B : s}{\Gamma \vdash_{\mathcal{C}} u : B}$	if $A \downarrow_{\mathcal{C}} B$

Table 3. ALGEBRAIC PURE TYPE SYSTEMS

3. The rewrite relation $\rightarrow_{mix}$ is defined as $\rightarrow_{mix} = \rightarrow_\beta \cup \rightarrow_{\mathcal{R}}$.

In the sequel, we will make use of the following well-known definition.

Definition 12. Let $\rightarrow_{\mathcal{C}}$ and $\rightarrow_{\mathcal{D}}$ be rewrite relations on the set of pseudo-terms of $\mathcal{A}$. Then $\mathcal{L}(\mathcal{A},\mathcal{C})$ is said to *have the subject reduction property for* $\rightarrow_{\mathcal{D}}$ is for every $M \in \mathcal{L}(\mathcal{A},\mathcal{C})$ we have that $\Gamma \vdash_{\mathcal{C}} M : A$ and $M \rightarrow_{\mathcal{D}} N$ implies $\Gamma \vdash_{\mathcal{C}} N : A$.

Morphisms of Algebraic Type Systems. We define a morphism between algebraic type systems as a pair consisting of a morphism of pure type systems and a morphism of signatures.

Definition 13. Let $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ and $\mathcal{A}' = \mathcal{S}' +_{EA'} \mathcal{R}'$ be algebraic type systems. An algebraic type system morphism between $\mathcal{A}$ and $\mathcal{A}'$ is a pair of mappings $\phi + \psi$ such that

1. $\phi : \mathcal{S} \rightarrow \mathcal{S}'$ is a pure type systems morphism,
2. $\psi = (\psi_1, \psi_2) : (S, F) \rightarrow (S', F')$ is a sorted signature morphism,
3. if $(s, u) \in EA$, then $(\psi_1(s), \phi(u)) \in EA'$.

Every morphism $\phi + \psi$ of algebraic type systems from $\mathcal{A}$ to $\mathcal{A}'$ can be extended homomorphically into a map from the set of $\mathcal{A}$-pseudo-terms into the set of $\mathcal{A}'$-pseudo-terms. By abuse of notation, we denote this map by $\phi + \psi$.

3 Techniques

In this section we present two techniques that can be used to derive termination of an algebraic type system $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ from termination of $\mathcal{S}$ and $\mathcal{R}$. First we briefly comment on which problems occur.

A first problem is caused by the fact that an algebraic type system might have more terms than its underlying pure type system because of the conversion rule. So if $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$, then $\mathcal{L}(\mathcal{A}, mix)$ is not necessarily contained in $\mathcal{L}(\mathcal{S}, \beta)$. As a consequence, we cannot immediately conclude termination of $\rightarrow_\beta$ on $\mathcal{L}(\mathcal{A}, mix)$ from termination of $\rightarrow_\beta$ on $\mathcal{L}(\mathcal{S}, \beta)$.

Second, a well-known result originally due to Klop [21] (see also [11]) states that the rewrite relation $\rightarrow_{mix}$ is not necessarily confluent on the set of pseudo-terms of an algebraic type system. As a consequence, the traditional proof of subject reduction of $\mathcal{L}(\mathcal{A}, mix)$ for $\rightarrow_\beta$ breaks down [4, 9].

A third problem is how to infer termination of $\rightarrow_{\mathcal{R}}$ on some set of terms of an algebraic type system from termination of $\mathcal{R}$. This is discussed in Section 4.

3.1 Termination by Translation

A well-known technique to show termination of a rewriting system $(A_1, \rightarrow_1)$ is to map it into a terminating rewriting system $(A_2, \rightarrow_2)$ such that one step in the former corresponds to at least one step in the latter. In this subsection, we extend this technique to the case of algebraic type systems. Then the mapping

should not only preserve the rewrite relation but should also map a derivable judgement to a derivable judgement. We have the following result; its proof is very easy and is omitted.

Proposition 14. *Let $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ and $\mathcal{A}' = \mathcal{S}' +_{EA'} \mathcal{R}'$ be algebraic type systems. Let $\to_C, \to_D$ be rewrite relations on the set of pseudo-terms of $\mathcal{A}$ and let $\to_{C'}, \to_{D'}$ be rewrite relations on the set of pseudo-terms of $\mathcal{A}'$. Let $\phi + \psi : \mathcal{A} \to \mathcal{A}'$ be a morphism of algebraic type systems.*

1. *Suppose that for all pseudo-terms M and N in $\mathcal{A}$, we have*
   (a) $M \downarrow_C N$ implies $(\phi+\psi)(M) \downarrow_{C'} (\phi+\psi)(N)$,
   (b) $M \to_D N$ implies $(\phi+\psi)(M) \to^+_{D'} (\phi+\psi)(N)$.
   Then termination of $(\mathcal{L}(\mathcal{A}', C'), \to_{D'})$ implies termination of $(\mathcal{L}(\mathcal{A}, C), \to_D)$.
2. *If in addition we have that for all pseudo-terms M and N in $\mathcal{A}$, $M \to_{\mathcal{E}} N$ implies $(\phi+\psi)(M) \to^*_{\mathcal{E}'} (\phi+\psi)(N)$, for a rewrite relation $\to_{\mathcal{E}}$ on the set of pseudo-terms of $\mathcal{A}$ and a rewrite relation $\to_{\mathcal{E}'}$ on the set of pseudo-terms of $\mathcal{A}'$, then we have that termination of $\mathcal{D}'$ relative to $\mathcal{E}'$ on $\mathcal{L}(\mathcal{A}', C')$ implies termination of $\mathcal{D}$ relative to $\mathcal{E}$ on $\mathcal{L}(\mathcal{A}, C)$*

We stress that one of the purposes of the present paper is not to show that termination by translation is a technique to infer termination, because this is of course well-known, but to show that Proposition 14 has useful applications.

3.2 Termination by Stability

In this subsection we present a second technique to infer termination of an algebraic type system: termination by stability. The principle of this technique is due to Dougherty. He shows in [16] that termination of $\to_\beta \cup \to_{\mathcal{R}}$ follows from termination of $\to_\beta$ and termination of $\to_{\mathcal{R}}$, provided that we restrict attention to a set of *stable terms*. Stability is in fact an abstract form of typing, and Dougherty's result is obtained for untyped λ-calculus. In this subsection we adapt Dougherty's result to the case of algebraic type systems. Instead of making use of a generalisation of the notion of stability, adapted to the case of algebraic type systems, we will make use of a similar notion which we call *preservation of sorts*. It is defined as follows.

Definition 15. Let $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ be an algebraic type system and let $\to_C$ be a rewrite relation on the set of pseudo-terms of $\mathcal{A}$.

1. Two pseudo-terms M and N are said to be *C-legally convertible* if there is an environment Γ and a sequence of terms $P_1, \ldots, P_n$ such that $\Gamma \vdash_C P_i : s$ for every $i \in \{1, \ldots, n\}$ and
$$M \downarrow_C P_1 \downarrow_C \ldots \downarrow_C P_n \downarrow_C N.$$
2. $\mathcal{L}(\mathcal{A}, C)$ has *preservation of sorts* if no two sorts are C-legally convertible and no sort is C-legally convertible with a pseudo-term of the form $\Pi x : A.B$.

Preservation of sorts plays a rôle similar to arity-checking in [16]. In [4], it is proved that an algebraic type system obtained by a λ-calculus from the λ-cube with a term rewriting system enjoys preservation of sorts.

Preservation of sorts is used to show that algebraic reduction preserves β-normal forms.

Lemma 16. *Let $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ be an algebraic type system that has preservation of sorts. Let M be a β-normal form with $M \to_{\mathcal{R}} M'$. Then M' is a β-normal form as well.*

Proof. First, let $l \to r$ be a rewrite rule of $\mathcal{R}$ and suppose that $l\theta$ is in β-normal form. We show that $r\theta$ is in β-normal form as well. To start with, since r consists only of function symbols of $\mathcal{R}$ and variables, there are no β-redexes in the r-part of $r\theta$. Further, there are no β-redexes in the θ-part of $r\theta$, since all variables occurring in r occur also in l and $l\theta$ is supposed to be in β-normal form. Finally, there are no β-redexes 'on the border between r and θ' in $r\theta$, since no sort s is β-convertible to a term of the form $\Pi y : A.B$, so neither l nor r has a subterm of the form xP.

Then we can proceed by induction on $C[\bullet]$ to prove that $C[l\theta]$ is in β-normal form implies that $C[r\theta]$ is in β-normal form. In the induction also preservation of types is used.

In the proof of the main result of this section, Theorem 19, we make use of a lemma concerning reduction diagrams. In these diagrams, we make use of complete developments of the set of all β-redexes in a term. The result of performing such a complete development in a term M is defined inductively as follows.

Definition 17. The term M^* is inductively defined as follows.

1. $(aM_1 \dots M_n)^* = aM_1^* \dots M_n^*$ with $a \in V \cup U \cup S$ and $n \geq 0$,
2. $(\lambda x : P.M)^* = \lambda x : P^*.M^*$,
3. $(\Pi x : P.Q)^* = \Pi x : P^*.Q^*$,
4. $((\lambda x : Q.M)NP_1 \dots P_n)^* = M^*[x := N^*]P_1^* \dots P_n^*$ with $n \geq 0$,
5. $f(M_1, \dots, M_n)^* = f(M_1^*, \dots, M_n^*)$ with $n \geq 0$.

We will make use of the following diagrams.

Lemma 18. *1. If $M \to_\beta N$, then $M^* \to_\beta^* N^*$. In a diagram:*

$$\begin{array}{ccc} M & \xrightarrow[\beta]{} & N \\ \downarrow & & \downarrow \\ M^* & \xrightarrow[\beta^*]{} & N^* \end{array}$$

2. If $M \to_{\mathcal{R}} N$, then $M \to_{\mathcal{R}}^* N^*$. In a diagram:

$$\begin{array}{ccc} M & \xrightarrow[\mathcal{R}]{} & N \\ \downarrow & & \downarrow \\ M^* & \xrightarrow[\mathcal{R}^*]{} & N^* \end{array}$$

If the step $M \to_{\mathcal{R}} N$ takes place at position ϵ, then $M' = N'$.

Now we present the theorem which is the core of the termination by stability method.

Theorem 19. *Let $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ be an algebraic type system. Suppose that*

1. *$\mathcal{L}(\mathcal{A}, \mathcal{C})$ has the subject reduction property for $\to_{mix}$,*
2. *$\mathcal{L}(\mathcal{A}, \mathcal{C})$ has preservation of sorts*
3. *$(\mathcal{L}(\mathcal{A}, \mathcal{C}), \to_\beta)$ is terminating,*
4. *$(\mathcal{L}(\mathcal{A}, \mathcal{C}), \to_{\mathcal{R}})$ is terminating.*

Then $(\mathcal{L}(\mathcal{A}, \mathcal{C}), \to_{mix})$ is terminating.

Proof. Let $M \in \mathcal{L}(\mathcal{A}, \mathcal{C})$. We prove that M is terminating with respect to $\to_{mix}$ by induction on the maximal length of a β-rewrite sequence starting in M, denoted by $\mathsf{maxred}_\beta(M)$.

1. $\mathsf{maxred}_\beta(M) = 0$. Let σ be a rewrite sequence starting in M. By Lemma 16 σ is of the form

$$\sigma : M = M_0 \to_{\mathcal{R}} M_1 \to_{\mathcal{R}} M_2 \to_{\mathcal{R}} \dots.$$

   By hypothesis 4, σ is finite.
2. $\mathsf{maxred}_\beta(M) > 0$. We proceed by induction on M, only the two difficult cases are treated here.
   (a) $M = (\lambda x : A.P)QQ_1 \dots Q_n$ with $n \geq 0$. Let $\sigma : M = M_0 \to_{mix} M_1 \to_{mix} M_2 \to_{mix} \dots$ be a rewrite sequence starting in M. Two cases are distinguished.
      i. Every term in σ is of the form $(\lambda x : A'.P')Q'Q'_1 \dots Q'_n$ with $A \to^*_{mix} A', P \to^*_{mix} P', Q \to^*_{mix} Q', Q_i \to^*_{mix} Q'_i$. Then σ is finite by the induction hypothesis on M.
      ii. There is a k such that $M_k = (\lambda x : A'.P')Q'Q'_1 \dots Q'_m$ and $M_{k+1} = P'[x := Q']Q'_1 \dots Q'_m$. Now M_{k+1} is a mix-reduct of $M' = P[x := Q]Q_1 \dots Q_m$. By the induction hypothesis on $\mathsf{maxred}_\beta(M)$, M' is terminating and hence σ is finite.
   (b) $M = f(M_1, \dots, M_n)$. Let $\sigma : M = M_0 \to_{mix} M_1 \to_{mix} M_2 \to_{mix} \dots$ be a rewrite sequence starting in M. Using Lemma 18 we build a rewrite sequence σ^* starting in M^* as follows:

$$\begin{array}{lccccccc} \sigma : & M_0 & \xrightarrow[mix]{} & M_1 & \xrightarrow[mix]{} & M_2 & \xrightarrow[mix]{} & \cdots \\ & \downarrow & & \downarrow & & \downarrow & & \\ \sigma^* & M_0^* & \xrightarrow[mix]{} & M_1^* & \xrightarrow[mix]{} & M_2^* & \xrightarrow[mix]{} & \cdots \end{array}$$

   Since clearly $\mathsf{maxred}_\beta(M^*) < \mathsf{maxred}_\beta(M)$ if M is not in β-normal form, the induction hypothesis on $\mathsf{maxred}_\beta(M)$ yields that σ^* must be finite. Therefore, there is a k such that for every $l \geq k$ every redex contracted in $M_l \to_{mix} M_{l+1}$ is in a subterm Q_1 of a subterm $(\lambda x : B.Q_0)Q_1$ of M_l. The subterm $(\lambda x : A.Q_0)Q_1$ is a subterm of a reduct of a subterm of M and hence by the induction hypothesis terminating.

Note that we need preservation of sorts in order to be able to apply Lemma 16. Further, we use the properties that a subterm of a term in $\mathcal{L}(\mathcal{A},\mathcal{C})$ is in $\mathcal{L}(\mathcal{A},\mathcal{C})$ and that a $\rightarrow_{mix}$-reduct of a term in $\mathcal{L}(\mathcal{A},\mathcal{C})$ is in $\mathcal{L}(\mathcal{A},\mathcal{C})$. Indeed, subject reduction of $\mathcal{L}(\mathcal{A},\mathcal{C})$ for $\rightarrow_{mix}$ is crucial.

4 Algebraic Reduction

In order to show termination of $(\mathcal{L}(\mathcal{A},mix),\rightarrow_{mix})$ for some algebraic type system $\mathcal{A}=\mathcal{S}+_{EA}\mathcal{R}$, we need to show in particular that $\rightarrow_{\mathcal{R}}$ is terminating on $\mathcal{L}(\mathcal{A},mix)$. As already mentioned in the introduction, this is not guaranteed by termination of $\mathcal{R}$ only. In this section we present two results concerning termination of $\rightarrow_{\mathcal{R}}$ on some set of terms in an algebraic type system.

If a typed term rewriting system $\mathcal{R}$ is terminating, then it is not necessarily the case that its untyped version $\mathcal{E}(\mathcal{R})$, which is obtained by erasing all information concerning the sorts, is terminating. A counterexample is for instance an adaptation of the counterexample by Toyama, showing that termination is not a modular property of term rewriting system, see [25]. Now the difficulties can simply be avoided by considering term rewriting systems that are persistently terminating. A typed term rewriting system $\mathcal{R}$ is said to be *persistently terminating* is its untyped version $\mathcal{E}(\mathcal{R})$ is also terminating.

It is quite easy to see that if a term rewriting system $\mathcal{R}$ is persistently terminating, meaning that also $\mathcal{E}(\mathcal{R})$ is terminating, then $\rightarrow_{\mathcal{R}}$ is terminating on the set of pseudo-terms of an algebraic type system of the form $\mathcal{A}=\mathcal{S}+_{EA}\mathcal{R}$, as follows. We assume that the single sorted term rewriting system $\mathcal{E}(\mathcal{R})$ is terminating. Now we extend the signature of $\mathcal{E}(\mathcal{R})$ with fresh symbols δ of arity 0, and $\underline{\Pi}$, $\underline{\lambda}$, Appl of arity 2. Note that this extension $\mathcal{E}(\mathcal{R})'$ is still terminating. All pseudo-terms of an algebraic type system $\mathcal{A}=\mathcal{S}+_{EA}\mathcal{R}$ can be mapped to terms of the only sort of $\mathcal{E}(\mathcal{R})$, say s, by means of the following mapping:

$$\begin{aligned}
|a| &= \delta && \text{for } a\in V\cup U\cup S\\
|M\ N| &= \mathsf{App}(|M|,|N|)\\
|\lambda x:A.M| &= \underline{\lambda}(|A|,|M|)\\
|\Pi x:A.B| &= \underline{\Pi}(|A|,|B|)\\
|f(t_1,\ldots,t_n)| &= f(|t_1|,\ldots,|t_n|) && \text{if } f\in F \text{ and } \mathsf{ar}(f)=n
\end{aligned}$$

Since this mapping preserves the one-step rewrite relation, it follows that $\rightarrow_{\mathcal{R}}$ is terminating on the set of pseudo-terms.

Now the question is which terminating term rewriting systems are persistently terminating. An answer to this question is given by Zantema, who shows in [25] that termination is a persistent property both for non-collapsing and for non-duplicating term rewriting systems. Using the observation above, we have the following corollary of Zantema's result. It will be used in Section 5 to show that under certain conditions the combination of a terminating pure type system and a terminating and non-collapsing term rewriting system is terminating.

Theorem 20. *Let $\mathcal{A}=\mathcal{S}+_{EA}\mathcal{R}$ be an algebraic type system such that:*

1. $\mathcal{R}$ *is terminating,*
2. $\mathcal{R}$ *is either non-collapsing or non-duplicating.*

Then $\to_{\mathcal{R}}$ *is terminating on the set of pseudo-terms of* $\mathcal{A}$.

Another way to obtain termination of $\to_{\mathcal{R}}$ on some set of terms of an algebraic type system $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ is by instead of imposing restrictions on $\mathcal{R}$ imposing restrictions on $\mathcal{A}$. Using the techniques sketched above, the following result can be obtained; for lack of space the proof is omitted.

Theorem 21. *Let* $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ *be an algebraic type system such that*

1. $\mathcal{L}(\mathcal{A}, \mathcal{C})$ *has preservation of sorts,*
2. $\mathcal{L}(\mathcal{A}, \mathcal{C})$ *has the subject reduction property for* $\to_{\mathcal{R}}$,
3. $\mathcal{R}$ *is terminating.*

Then $(\mathcal{L}(\mathcal{A}, \mathcal{C}), \to_{\mathcal{R}})$ *is terminating.*

This result will be used in Section 5 in order to show that the combination of a terminating pure type system without dependent types and a terminating term rewriting system is terminating.

5 Applications

In this section we apply the methods presented in Section 3 to several situations of interest.

5.1 Non-dependent Algebraic Type Systems

In this subsection we consider a restricted class of algebraic type systems where only β-reduction, not mix-reduction, is used in the conversion rule. So the set of legal terms we consider is of the form $\mathcal{L}(\mathcal{A}, \beta)$. If we have that the set of legal terms $\mathcal{L}(\mathcal{A}, \beta)$ has the subject reduction property for $\to_{\mathcal{R}}$, then the termination by stability method can be presented in a somewhat simpler form. This is expressed in the following proposition.

Proposition 22. *Let* $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ *be an algebraic type system and suppose that:*

1. $\mathcal{L}(\mathcal{A}, \beta)$ *has the subject reduction property for* $\to_{\mathcal{R}}$,
2. $(\mathcal{L}(\mathcal{S}, \beta), \to_{\beta})$ *is terminating,*
3. $\mathcal{R}$ *is terminating.*

Then $(\mathcal{L}(\mathcal{A}, \beta), \to_{mix})$ *is terminating.*

Proof. First, since $\mathcal{L}(\mathcal{S}, \beta)$ has the subject reduction property for $\to_{\beta}$, it follows easily that $\mathcal{L}(\mathcal{A}, \beta)$ has the subject reduction property for $\to_{\beta}$. Because moreover we have by hypothesis that $\mathcal{L}(\mathcal{A}, \beta)$ has the subject reduction property for $\to_{\mathcal{R}}$, we can conclude that $\mathcal{L}(\mathcal{A}, \beta)$ has the subject reduction property for $\to_{mix}$. Second, $\mathcal{L}(\mathcal{A}, \beta)$ has preservation of sorts. Third, termination of $(\mathcal{L}(\mathcal{A}, \beta), \to_{\beta})$ follows from termination of $(\mathcal{L}(\mathcal{S}, \beta), \to_{\beta})$. Fourth, we have by Theorem 21 that $(\mathcal{L}(\mathcal{A}, \beta), \to_{\mathcal{R}})$ is terminating.

Hence we can by Theorem 19 conclude that $(\mathcal{L}(\mathcal{A}, \beta), \to_{mix})$ is terminating.

Non-dependent λ-calculi. We obtain a useful corollary of Proposition 22 by applying it to the case that the pure type system of the algebraic type system is a λ-calculus with non-dependent types, for instance a λ-calculus in the left plane of the λ-cube, and all sorts are declared to live in $\star$.

Corollary 23. *Let $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ be an algebraic type system such that:*

1. *$\mathcal{S}$ is $\lambda\!\rightarrow$, $\lambda 2$ or $\lambda\omega$,*
2. *$\mathsf{codom}(EA) = \{\star\}$,*
3. *$\mathcal{R}$ is terminating.*

Then $(\mathcal{L}(\mathcal{A}, \beta), \rightarrow_{mix})$ is terminating.

Proof. Let $\mathcal{S}$ be $\lambda\!\rightarrow$, $\lambda 2$ or $\lambda\omega$. It can be shown that $\Gamma \vdash_\beta l\theta : A$ implies $\Gamma \vdash_\beta r\theta : A$. Then, since there are no rewrite steps in the types, and we have $s : \star$ for every sort s, we can show by induction on the context that $\Gamma \vdash_\beta C[l\theta] : A$ implies $\Gamma \vdash_\beta C[r\theta] : A$. This yields that $\mathcal{L}(\mathcal{A}, \beta)$ has the subject reduction property for $\rightarrow_{\mathcal{R}}$. termination of $(\mathcal{L}(\mathcal{A}, \beta), \rightarrow_\beta)$ follows from termination of $(\mathcal{L}(\mathcal{S}, \beta), \rightarrow_\beta)$. We can conclude by Proposition 22 that $(\mathcal{L}(\mathcal{A}, \beta), \rightarrow_{mix})$ is terminating.

η. An inspection of the proof of Corollary 23 yields that we have the same result for the combination of a λ-calculus in the left plane of the cube with $\beta\eta$-reduction, and a terminating term rewriting system. The η-reduction relation, denoted by $\rightarrow_\eta$, is defined as the smallest compatible closure of

$$\lambda x : A.Mx \rightarrow M$$

with the side-condition that x has no free occurrence in M. If $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$, then we denote by $\rightarrow_{mix_\eta}$ the rewrite relation $\rightarrow_\beta \cup \rightarrow_\eta \cup \rightarrow_{\mathcal{R}}$. We have the following result.

Corollary 24. *Let $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ be an algebraic type system such that:*

1. *$\mathcal{S}$ is $\lambda\!\rightarrow$, $\lambda 2$ or $\lambda\omega$,*
2. *$\mathsf{codom}(EA) = \{\star\}$,*
3. *$\mathcal{R}$ is terminating.*

Then $(\mathcal{L}(\mathcal{A}, \beta\eta), \rightarrow_{mix_\eta})$ is terminating.

For $\lambda\!\rightarrow$ and $\lambda 2$, a similar result for the union of β-reduction, η-expansion and algebraic reduction can be obtained.

5.2 Non-collapsing Term Rewrite Rules

In this subsection we consider combinations of a pure type system and term rewriting system without collapsing rules. Throughout this section, we assume an algebraic type system $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ with $\mathcal{R}$ a non-collapsing term rewriting system. Recall that a rewrite rule is said to be collapsing if it is of the form $l \rightarrow x$

with $x \in V$. We will suppose that for every sort s in a term rewriting system there is a distinguished constant c_s of sort s. This is not a serious restriction.

We denote by $\mathcal{R}'$ the term rewriting system $\mathcal{R}$ extended with a rewrite rule

$$f(x_1, \ldots, x_n) \to c_s$$

for every function symbol $f : s_1 \times \ldots \times s_n \to s$ in $\mathcal{R}$. Further, we denote by $\to_{mix'}$ the rewrite relation $\to_\beta \cup \to_{\mathcal{R}'}$ which is defined on the set of pseudo-terms of $\mathcal{A}$ and on the set of pseudo-terms of $\mathcal{S} +_{EA} \mathcal{R}'$.

Now the termination by stability method can be presented in a slightly more simple way. For the proof, we need $(\mathcal{L}(\mathcal{A}, mix'), \to_\beta)$ to be terminating. This follows from termination of $(\mathcal{L}(\mathcal{S}, \beta), \to_\beta)$ by the termination by translation method, provided that $\mathcal{S}$ has enough axioms and enough products, which can be enforced in a rather crude way by requiring $\mathcal{A}$ to be regular, a property that is defined as follows.

Definition 25. An algebraic type system $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ is said to be *regular* if the following two conditions are satisfied:

1. Universes are connected, that is: for every $u \in U$ there exists $u' \in U$ such that either $(u, u') \in TA$ or $(u', u) \in TA$.
2. Universes which contain an algebraic sort have products, that is: for all $s_1, s_2 \in \mathsf{codom}(EA)$ there exists $s_3 \in \mathsf{codom}(EA)$ such that $(s_1, s_2, s_3) \in TR$.

Regularity is closely related to the notion of fullness for pure type systems. We have the following result.

Proposition 26. *Let $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ be an algebraic type system such that:*

1. *$\mathcal{S}$ is regular,*
2. *$(\mathcal{L}(\mathcal{S}, \beta), \to_\beta)$ is terminating,*
3. *$\mathcal{R}$ is non-collapsing,*
4. *$\mathcal{R}$ is terminating.*

Then $(\mathcal{L}(\mathcal{A}, mix), \to_{mix})$ is terminating.

Proof. Let $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ be an algebraic type system and assume that $\mathcal{R}$ is a non-collapsing term rewriting system. We can show that the rewrite relation $\to_{mix'}$ is confluent on the set of pseudo-terms of $\mathcal{A}$, by projecting a rewrite sequence to one where the algebraic part is replaced by constants c_s. As a consequence, we obtain that $\mathcal{L}(\mathcal{A}, mix')$ has the subject reduction property for $\to_{mix}$, and that $\mathcal{L}(\mathcal{A}, mix')$ has preservation of sorts.

Since $\mathcal{R}$ is terminating and non-collapsing, we have by Theorem 20 that the rewrite relation $\to_{\mathcal{R}}$ is terminating on the set of pseudo-terms of $\mathcal{A}$.

Using regularity, we can show using similar techniques to the one presented in [8], that termination of $(\mathcal{L}(\mathcal{S}, \beta), \to_\beta)$ implies termination of $(\mathcal{L}(\mathcal{A}, mix'), \to_\beta)$.

It then follows by Theorem 19 that $(\mathcal{L}(\mathcal{A}, mix'), \to_{mix})$ is terminating. Hence also the subsystem $(\mathcal{L}(\mathcal{A}, mix), \to_{mix})$ is terminating.

Ground Rewriting. Proposition 26 can be applied to the case where we consider the ground rewriting relation (the rewrite relation restricted to terms without variables) of an arbitrary terminating rewriting system, since the ground rewriting relation can be considered as generated by the infinitely many ground instances of the rewrite rules. A ground instance of a rewrite rule is clearly a non-collapsing rewrite rule. If $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ is an algebraic type system, then we denote by $\rightarrow_{\mathcal{R}_g}$ the ground rewrite relation of $\mathcal{R}$, and by $\rightarrow_{mix_g}$ the rewrite relation $\rightarrow_\beta \cup \rightarrow_{\mathcal{R}_g}$. We have the following corollary of Proposition 26.

Corollary 27. *Let $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ be an algebraic type system such that:*

1. *$\mathcal{S}$ is regular,*
2. *$(\mathcal{L}(\mathcal{S}, \beta), \rightarrow_\beta)$ is terminating,*
3. *$\mathcal{R}$ is terminating on ground terms.*

Then $(\mathcal{L}(\mathcal{A}, mix_g), \rightarrow_{mix_g})$ is terminating.

η. Again, the results presented in this subsection can easily be adapted to the case of an algebraic type system with $\beta\eta$-reduction.

5.3 Non-duplicating Term Rewrite Rules

If we consider an algebraic type system $\mathcal{A} = \mathcal{S} +_{EA} \mathcal{R}$ such that $\mathcal{R}$ is non-duplicating, then termination by translation may be applied to obtain several results. Indeed, we can define for every algebraic type system 'universal' algebraic type system such that β-strong normalisation of the latter imply mix-strong normalisation of the latter. Then, using termination of translation and postponement techniques, one can prove, provided $\mathcal{R}$ is terminating and non-duplicating, that termination of $\rightarrow_\beta$ in the 'universal' algebraic type system implies termination of $\rightarrow_{mix}$ in $\mathcal{A}$.

This illustrates that, despite its extreme simplicity, Proposition 14 can take us quite some way in the study of termination for algebraic type systems.

6 Concluding Remarks

We have developed purely syntactic methods to prove termination of algebraic type systems. Although we do not establish termination as a modular property of algebraic type systems, our methods yield simple proofs of well-known results as well as new results. Moreover they lead to a better understanding of the interaction between a type system and a rewriting system. In addition, the methods developed in this paper, especially termination by stability, may also be adapted to yield similar results for confluence, extending the 'confluence by stability' result in [16].

The most outstanding question left unanswered is termination of β-reduction for 'universal' algebraic type systems. A positive answer to that question would

be a definite step towards modular proofs of strong normalization for algebraic type systems. It would also be interesting to see whether our methods can be adapted to algebraic type systems with higher-order term rewriting à la Jouannaud-Okada [19] or to typed λ-calculi with pattern matching [20].

Finally, the technique of derivation-preserving translations reveals the impossibility of distinguishing in the internal logic of the algebraic type system between two closed algebraic terms of the same type. This is clearly a weakness of algebraic type systems for dependent type theories. Some possible ways to fix this are described in [6, 20] and [10].

Acknowledgements. The diagrams in this paper are designed using the package Xy-pic of Kristoffer H. Rose. We thank the referees for their help in improving the presentation of the paper.

References

1. 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.
2. 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.
3. F. Barbanera and M. Fernández. Intersection type assignment systems with higher-order algebraic rewriting. *Theoretical Computer Science*, 170(1–2):173–207, 15 December 1996.
4. 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.
5. H. Barendregt. Lambda calculi with types. In S. Abramsky, D. M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, pages 117–309. Oxford Science Publications, 1992. Volume 2.
6. 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.
7. 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.
8. G. Barthe, J. Hatcliff, and M.H. Sørensen. Weak Normalization implies Strong Normalization in Generalized Non-Dependent Pure Type Systems. Draft, 1997.
9. G. Barthe and P.-A. Melliès. On the subject reduction property for algebraic type systems. Proceedings of CSL'96. To appear as LNCS, 1996.
10. 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.

11. V. Breazu-Tannen. Combining algebra and higher-order types. In *Proceedings of LICS'88*, pages 82–90. IEEE Computer Society Press, 1988.
12. V. Breazu-Tannen and J. Gallier. Polymorphic rewriting conserves algebraic strong normalisation. *Theoretical Computer Science*, 83:3–28, 1990.
13. T. Coquand and G. Huet. The Calculus of Constructions. *Information and Computation*, 76(2/3):95–120, February/March 1988.
14. R. Di Cosmo. A brief history of rewriting with extensionality. In F. Kamareddine, editor, *International Summer School on Type Theory and Term Rewriting, Glasgow, September 1996*. Kluwer, 199x. To appear.
15. R. Di Cosmo and D. Kesner. Combining algebraic rewriting, extensional lambda calculi, and fixpoints. *Theoretical Computer Science*, 169(2):201–220, 5 December 1996.
16. D. Dougherty. Adding algebraic rewriting to the untyped lambda calculus. *Information and Computation*, 101:251–267, 1992.
17. M. Fernandez. *Modèles de calcul multiparadigmes fondés sur la réécriture*. PhD thesis, Université Paris-Sud Orsay, 1993.
18. H. Geuvers and M.-J. Nederhof. A modular proof of strong normalisation for the Calculus of Constructions. *Journal of Functional Programming*, 1:155–189, 1991.
19. 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.
20. J.-P. Jouannaud and M. Okada. Abstract data type systems. *Theoretical Computer Science*, 173(2):349–391, 1997.
21. J.W. Klop. *Combinatory reduction systems*. Number 127 in Mathematical Centre Tracts. CWI, 1980.
22. J.W. Klop. Term-rewriting systems. In S. Abramsky, D. M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, pages 1–116. Oxford Science Publications, 1992. Volume 2.
23. K. Meinke and J.V. Tucker, editors. *Many sorted logic and its applications*. John Wiley and Sons, 1993.
24. J. van de Pol. *Termination of higher-order rewrite systems*. PhD thesis, University of Utrecht, 1996.
25. H. Zantema. Termination of term rewriting: Interpretation and type elimination. *Journal of Symbolic Computation*, 17(1):23–50, January 1994.