Non-ω-overlapping TRSs are UN

Stefan Kahrs and Connor Smith

School of Computing
University of Kent
Canterbury, United Kingdom
{S.M.Kahrs,cls204}@kent.ac.uk

Abstract

This paper solves problem #79 of RTA's list of open problems [14] — in the positive. If the rules of a TRS do not overlap w.r.t. substitutions of infinite terms then the TRS has unique normal forms. We solve the problem by reducing the problem to one of consistency for "similar" constructor term rewriting systems. For this we introduce a new proof technique. We define a relation $\Downarrow$ that is consistent by construction, and which — if transitive — would coincide with the rewrite system's equivalence relation $=_R$.

We then prove the transitivity of $\Downarrow$ by coalgebraic reasoning. Any concrete proof for instances of this relation only refers to terms of some finite coalgebra, and we then construct an equivalence relation on that coalgebra which coincides with $\Downarrow$.

Keywords and phrases consistency, omega-substitutions, uniqueness of normal forms

Digital Object Identifier 10.4230/LIPIcs.FSCD.2016.12

1 Introduction

For over 40 years [13] it has been known that TRSs that are left-linear and non-overlapping are confluent, and for over 30 years [8] that non-overlapping on its own may not even give us unique normal forms:

▶ **Example 1.** By Huet [8]: $\{F(x,x) \to A, F(x,G(x)) \to B, C \to G(C)\}$. The term $F(C,C)$ possesses two distinct normal forms, A and B.

However, in a certain sense the first two rules overlap semantically: the infinite term $G(G(\cdots))$ provides such an overlap, and in the world of infinitary rewriting [9] the term C even rewrites to that term in the limit.

The notion of overlap is based on the notion of substitution. By changing the codomain of the substitutions of concern from the set of finite terms to the set of infinitary (finite or infinite) ones we arrive at the notion of ω-overlap.

This creates the question: *do non-ω-overlapping TRSs have unique normal forms*? This was first conjectured 27 years ago by Ogawa [11], with an incomplete proof, and the problem is still listed as open problem 79 in RTA's list of open problems.

When making the step from a rewrite relation $\to_R$ to its equivalence closure $=_R$ one is typically interested in its consistency [3, p32ff], i.e. are there terms t, u such that $\neg(t =_R u)$?

Both uniqueness of normal forms (UN) and consistency (CON) can be looked at as properties of open terms or ground terms. We stick in this paper to the versions on open terms, as these notions are unaffected by signature extensions; for the versions on ground terms, UN can be lost and CON gained when we extend the signature. Moreover, on open terms UN implies CON.

For non-ω-overlapping systems UN and CON are closely related, as we can extend non-UN systems in a seemingly harmless way to make them fail CON too:

1st International Conference on Formal Structures for Computation and Deduction (FSCD 2016).
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

▶ **Example 2.** Add to the system of Example 1 the rewrite rules $H(A, x, y) \to x$ and $H(B, x, y) \to y$. The system remains non-overlapping but it is now inconsistent.

Even if a TRS is non-ω-overlapping, the reduction relation $\to_R$ may still not be confluent (and so we need a different approach to show consistency); this follows from a well-known example by Klop [10]:

▶ **Example 3.** $\{A \to C(A), C(x) \to D(x, C(x)), D(x, x) \to E\}$.

In this system we have $A \to_R^* E$ and $A \to_R^* C(E)$, but $C(E)$ and E have no common reduct.

1.1 Translation of TRSs to Constructor TRSs

We are going to show how TRSs can be translated into Constructor TRSs, without affecting its equivalence in a substantial way, in particular: consistency is both preserved and reflected by the translation, as is strong normalisation.

The translation works by (i) doubling up the signature, so that for each function symbol F we have both a constructor version F_c and a destructor F_d; (ii) translate the rewrite rules to make them comply with the regime of Constructor TRSs; (iii) add further rules that make former patterns regain pattern status.

▶ **Example 4.** If we take the rewrite rules of Combinatory Logic, $A(A(K, x), y) \to x$ and $A(A(A(S, x), y), z) \to A(A(x, z), A(y, z))$ and apply the translation, we end up with the following system:

$$
\begin{array}{cc}
A_d(A_c(K_c, x), y) \to x & A_d(A_c(A_c(S_c, x), y), z) \to A_d(A_d(x, z), A_d(y, z)) \\
A_d(K_c, x) \to A_c(K_c, x) & A_d(S_c, x) \to A_c(S_c, x) \\
K_d \to K_c & A_d(A_c(S_c, x), y) \to A_c(A_c(S_c, x), y) \\
 & S_d \to S_c
\end{array}
$$

The top two rules are the translated versions of the original rules, the ones below are their respective pattern rules.

In Example 4, an orthogonal TRS was translated into an orthogonal Constructor TRS. In general, this will not quite be the case, and non-ω-overlapping TRSs will not remain non-ω-overlapping either. However, all overlaps created by the translation are benign.

1.2 Consistency of Constructor Rewriting

At the heart of our overall proof is showing (for our rewrite systems in question) that the equivalence closure $=_R$ of single rewrite steps is a subrelation of a consistent relation $\Downarrow$ and therefore itself consistent. This relation $\Downarrow$ is defined using slightly stronger closure principles than those that characterise the joinability relation $\downarrow$, however they remain weak enough to ensure (for arbitrary TRSs) that $\Downarrow$ is consistent. Because $\Downarrow$ is closed under the same operations as $=_R$, *except for transitivity*, proving consistency of $=_R$ can be reduced to proving that $\Downarrow$ is transitive.

Our proof idea is then based on the following fundamental observations: (i) (inductive, finitely-branching) proofs are finite objects, (ii) therefore each proof can only refer to finitely many terms. Instead of asking the question: "is $t \Downarrow u$ true?" we consider its provability relative to some finite set of terms A ($t \Downarrow_A u$); we need A to be closed under subterms which implies that it is a coalgebra of the signature. We show that — provided the TRS is "suitably well-behaved" — such finite coalgebras give rise to a single structure one might call a *universal proof* for A that proves $t \Downarrow_A u$ whenever it holds. This universal proof also exhibits the property that $\Downarrow_A$ is an equivalence relation. We have that $t \Downarrow u$ iff $t \Downarrow_A u$ for some finite A. Since these coalgebras are closed under union, and $A \subseteq B \wedge t \Downarrow_A u \Rightarrow t \Downarrow_B u$, we have that $\Downarrow$ itself is transitive.

2 Preliminaries

We assume familiarity with the standard notions of term rewriting and infinitary term rewriting [18, 2], but use this section to fix some notation.

A signature Σ is a pair $(\mathcal{F}, \#)$ comprising a set $\mathcal{F}$ of function symbols and a function $\# : \mathcal{F} \to \mathcal{N}$ assigning to each function symbol an arity. We write $\mathit{Ter}(\Sigma, X)$ for the set of finite terms over the variable set X, and $\mathit{Ter}^\omega(\Sigma, X)$ and $\mathit{Ter}^\infty(\Sigma, X)$ for the corresponding sets of rational and infinitary terms. Given a rewrite rule $l \to r$ we write $\xrightarrow{l \to r}$ for the substitutive closure of the rule, and $\xrightarrow{\epsilon}$ for the union of $\xrightarrow{l \to r}$ for all rewrite rules $l \to r$ of a TRS.

We say that a relation R on $\mathit{Ter}(\Sigma, X)$ is consistent if $\forall x, y \in X. x\ R\ y \Rightarrow x = y$. We say that a TRS is consistent (has the CON property) if the congruence closure of $\xrightarrow{\epsilon}$ is consistent on $\mathit{Ter}(\Sigma, Y)$, for an infinite set Y.

A substitution is a map $\sigma : V \to \mathit{Ter}(\Sigma, X)$ which we homomorphically extend to $\mathit{Ter}(\Sigma, V) \to \mathit{Ter}(\Sigma, X)$. Two terms $t \in \mathit{Ter}(\Sigma, V), u \in \mathit{Ter}(\Sigma, W)$ are said to be *unifiable* iff there is a pair of substitutions $\sigma : V \to \mathit{Ter}(\Sigma, X)$, $\theta : W \to \mathit{Ter}(\Sigma, X)$ such that $\sigma(t) = \theta(u)$. A pair of terms is said to be *ω-unifiable* if these conditions hold for substitutions with infinite terms in their codomain. Unifiability implies ω-unifiability, as all finite terms inhabit the infinite term universe as well.

As an aside, ω-unifiability of finite terms coincides with their unifiability w.r.t. substitutions with rational terms. This was first studied by Huet [7], and is these days usually implemented via union/find structures [16], which incidentally provide some inspiration for the notion of "proof graph" we consider later on.

2.1 Constructor Rewriting

A TRS is a *Constructor TRS* if the signature Σ is a *constructor signature*, i.e. it splits into two disjoint subsignatures Σ_c and Σ_d such that for any rewrite rule $F(p_1, \ldots, p_n) \to r$ we have $F \in \Sigma_d$ and $p_1, \ldots, p_n \in \mathit{Ter}(\Sigma_c, X)$.

The standard notion of non-overlapping TRSs is based on the notion of unifiability, and it can be simplified for Constructor TRSs. A Constructor TRS is non-overlapping iff the left-hand sides of any two different rules are not unifiable. Replacing 'unifiability' in that setting with 'ω-unifiability' provides the analogous (stronger) notion of non-ω-overlapping. A similar notion is that of *almost* non-ω-overlapping TRSs, which means that non-variable proper subterms of left-hand sides of rules are not ω-unifiable with left-hand sides of rules, and that $\xrightarrow{\epsilon}$ is deterministic.

2.2 Term-Coalgebras

In order to consider coalgebras of signatures Σ we would have to view signatures as functors on the category Set. However, we only need the following special instance of this concept later, which helps to keep the proofs short:

▶ **Definition 5.** Given a signature Σ, a *term-coalgebra* is a set $A \subseteq \mathit{Ter}^\infty(\Sigma, \emptyset)$ which is closed under subterms. It is called *finite* if it is a finite set, and *strongly finite* if in addition $A \subseteq \mathit{Ter}(\Sigma, \emptyset)$. We refer to the elements of a coalgebra as *nodes*.

More generally, Σ-coalgebras A would be characterized by a function $\upsilon : A \to \Sigma(A)$ which maps a node to a structure containing its root function symbol and the list of its subnodes. In that setting two nodes are bisimilar if their repeated unfolding via υ yield the same infinitary term. In a term-coalgebra this is unique, so bisimilarity coincides there with equality.

We also allow for variables in term-coalgebras by "freezing" them, i.e. using the canonical isomophism between $\mathit{Ter}^\infty(\Sigma_d + \Sigma_c, X)$ and $\mathit{Ter}^\infty(\Sigma_d + (\Sigma_c + X), \emptyset)$. Thus, when considered as a member of a term-coalgebra a variable is a nullary constructor. For heterogeneous relations between term-coalgebras we must therefore have that the variable set X is the same, so that they are coalgebras of the same functor. Relations between term-coalgebras can be consistent simply due to the lack of variables occurring in them as nodes, or indeed any other nodes: the empty set is a term-coalgebra that can only be consistently related to other term-coalgebras.

2.3 Relational Algebra

We use some standard constructions from relational algebra; in particular, we write $R \cdot S$ for relational composition in diagrammatical order, i.e. $a\ (R \cdot S)\ b \iff \exists c.\, a\ R\ c \wedge c\ S\ b$. As constants, we also use the empty relation $\emptyset$, and the identity relation *id*.

Binary relations on any set form a complete lattice, and so Tarski's fixpoint theorems [17] apply — any monotonic function f on these relation domains has a smallest fixpoint, $\mu(f)$, and a largest fixpoint $\nu(f)$. One usually writes $\mu x.f(x)$ for $\mu(\lambda x.f(x))$, etc. Most operations in relational algebra are monotonic (with the notable exception of complement, which we are not using here), as are the smallest/largest fixpoint constructions themselves [1, Proposition 1.2.18]. Thus any composition of these operations will result in a monotonic function on relations that therefore has both of these fixpoints. In the following, we will tacitly exploit that any function arrived by these means is monotonic.

▶ **Definition 6.** A predicate P on a complete lattice L is called *sup-continuous* iff for any function $f : I \to L$ such that $\forall x \in I.\, P(f(x))$ we also have $P(\bigsqcup_{i \in I} f(i))$.

Note that — as the definition also applies when the index set is empty — we would also necessarily have $P(\bot)$.

▶ **Proposition 7.** *Let P be a sup-continuous predicate on a complete lattice L, and f a monotonic function on L that preserves P, i.e. $\forall x \in L.\, P(x) \Rightarrow P(f(x))$. Then $P(\mu x.f(x))$.*

Proof. This follows from [1, Theorem 1.2.11]. That theorem defines for any ordinal β, $x_\beta = \bigsqcup\{f(x_\alpha) \mid \alpha < \beta\}$, and shows that for some β that is sufficiently large $x_\beta = \mu x.f(x)$. Hence the result follows by ordinal induction on the ordinal $\beta + 1$. ◀

3 Constructor Translation

We first demonstrate that a TRS can, in a sense, be viewed as a Constructor TRS, by translating it into a Constructor TRS with similar properties. This similarity is particularly strong for non-ω-overlapping TRSs.

To translate TRSs we use the concept of signature morphism — see [15] for a more general and modern version of the concept; we specialise it here for the standard signatures used in TRSs, as this concept rarely shows up in term rewriting literature.

▶ **Definition 8.** A signature morphism between signatures $\Sigma = (\mathcal{F}_\Sigma, \#_\Sigma)$ and $\Theta = (\mathcal{F}_\Theta, \#_\Theta)$ is a function $f : \mathcal{F}_\Sigma \to \mathcal{F}_\Theta$ such that $\#_\Theta(f(G)) = \#_\Sigma(G)$. Each signature morphism $f : \Sigma \to \Theta$ induces a map $T_f : \mathit{Ter}(\Sigma, X) \to \mathit{Ter}(\Theta, X)$ given as $T_f(F(t_1, \ldots, t_n)) = f(F)(T_f(t_1), \ldots, T_f(t_n))$ and $T_f(x) = x$ for $x \in X$.

Signatures and signature morphisms form a category, and this category clearly has coproducts, given by the disjoint union of signatures.

▶ **Definition 9.** Given a signature Σ we write $\Sigma 2$ for the coproduct $\Sigma + \Sigma$, which we view as a constructor signature; the images of Σ under the injections ι_1 and ι_2 give us Σ_c and Σ_d, respectively. We write F_c and F_d for the function symbols $\iota_1(F)$ and $\iota_2(F)$, respectively. We use the abbreviations $\lfloor t \rfloor$ for $T_{\iota_1}(t)$ and $\lceil t \rceil$ for $T_{\iota_2}(t)$.

So $\Sigma 2$ contains two copies of every function symbol, one as a constructor, and one as a destructor. The two embedding signature morphisms induce two different embeddings of terms, labelling all symbols as constructors or destructors, respectively.

▶ **Definition 10.** Let $\gamma : \Sigma 2 \to \Sigma$ be the signature morphism $[\mathit{id}, \mathit{id}]$, i.e. $\gamma(F_c) = F$, $\gamma(F_d) = F$. We write $|t|$ for $T_\gamma(t)$.

Thus, given a "labelled" term $t \in \mathit{Ter}(\Sigma 2, X)$, $|t| \in \mathit{Ter}(\Sigma, X)$ is the term we get when we erase the labels from t. Clearly, we have $|\lceil t \rceil| = t = |\lfloor t \rfloor|$, but no corresponding property when we go the other way, e.g. u and $\lceil |u| \rceil$ can differ.

▶ **Definition 11.** Given a TRS $T = (\Sigma, R)$, a *pattern* is a proper subterm of the left-hand side of a rule in R. We write $\mathrm{Pat}(T)$ for the set of all patterns of the TRS T.

Recall that Constructor TRSs are characterised by having all their patterns confined to $\mathit{Ter}(\Sigma_c, X)$. Therefore, patterns play a special role in the translation of TRSs into Constructor TRSs:

▶ **Definition 12.** Let T be a TRS with ruleset R and signature Σ. The *constructor translation of T* is a Constructor TRS $T' = (\Sigma 2, R')$ built as follows. $R' = R'_d \cup R'_c$, where $R'_d = \{F_d(\lfloor t_1 \rfloor, \ldots, \lfloor t_n \rfloor) \to \lceil r \rceil \mid F(t_1, \ldots, t_n) \to r \in R\}$ and $R'_c = \{F_d(\lfloor t_1 \rfloor, \ldots, \lfloor t_n \rfloor) \to F_c(\lfloor t_1 \rfloor, \ldots, \lfloor t_n \rfloor) \mid F(t_1, \ldots, t_n) \in \mathrm{Pat}(T)\}$.

Any rule of the original TRS becomes a rule in R'_d by turning its patterns into constructor patterns, and every non-variable pattern of T becomes a rule in R'_c. We have already seen the translation of Combinatory Logic (Example 4) as an example for this translation in the introduction. For simplicity, the constructor translation does not make a distinction which symbols already acted like constructors, such as the constants K and S.

We can relate a TRS to its constructor translation. First, when terms lose pattern status via the destructor translation then they can regain it through rewriting:

▶ **Lemma 13.** *Let T be a TRS and T' its constructor translation. For any $p \in \mathrm{Pat}(T)$ we have $\lceil p \rceil \to^*_{T'} \lfloor p \rfloor$.*

Proof. By induction on the term structure of p. If p is a variable then $\lceil p \rceil = \lfloor p \rfloor$. Otherwise, $p = F(t_1, \ldots, t_n)$ and $\lceil p \rceil = F_d(\lceil t_1 \rceil, \ldots, \lceil t_n \rceil)$. By induction hypothesis $\lceil t_i \rceil \to^*_{R'} \lfloor t_i \rfloor$, for all i. Therefore, $F_d(\lceil t_1 \rceil, \ldots, \lceil t_n \rceil) \to^*_{T'} F_d(\lfloor t_1 \rfloor, \ldots, \lfloor t_n \rfloor)$.

Moreover, $F_d(\lfloor t_1 \rfloor, \ldots, \lfloor t_n \rfloor) \to F_c(\lfloor t_1 \rfloor, \ldots, \lfloor t_n \rfloor)$ is a rule in R' and therefore overall $\lceil p \rceil = F_d(\lceil t_1 \rceil, \ldots, \lceil t_n \rceil) \to^*_{T'} F_d(\lfloor t_1 \rfloor, \ldots, \lfloor t_n \rfloor) \to_{T'} F_c(\lfloor t_1 \rfloor, \ldots, \lfloor t_n \rfloor) = \lfloor p \rfloor$. ◀

▶ **Lemma 14.** *Let T be a TRS and T' its constructor translation. If $t \to_T u$ then $\lceil t \rceil \to^+_{T'} \lceil u \rceil$. If $p \to_{T'} q$ then $\lfloor p \rfloor \to_T \lfloor q \rfloor \vee \lfloor p \rfloor = \lfloor q \rfloor$.*

Proof. If $t \to_T u$ then we must have that for some context C, substitution σ and rewrite rule $F(p_1, \ldots, p_n) \to r$ in T, $t = C[F(\sigma(p_1), \ldots, \sigma(p_n))]$ and $u = C[\sigma(r)]$. We clearly have $\lceil t \rceil = \lceil C \rceil[F_d(\lceil \sigma \rceil(\lceil p_1 \rceil), \ldots, \lceil \sigma \rceil(\lceil p_n \rceil))]$ and $\lceil u \rceil = \lceil C \rceil[\lceil \sigma \rceil(\lceil r \rceil)]$, where $\lceil C \rceil$ and $\lceil \sigma \rceil$ are straightforward extensions of the signature morphism to contexts and substitutions. By Lemma 13 we have $\lceil p_i \rceil \to^*_{T'} \lfloor p_i \rfloor$, hence by substitutivity of rewriting $\lceil \sigma \rceil(\lceil p_1 \rceil) \to^*_{T'} \lceil \sigma \rceil(\lfloor p_i \rfloor)$. Compatibility of rewriting gives us $\lceil t \rceil \to^*_{T'} \lceil C \rceil[F_d(\lceil \sigma \rceil(\lfloor p_1 \rfloor), \ldots, \lceil \sigma \rceil(\lfloor p_n \rfloor))]$. The latter term then rewrites in one step to $\lceil u \rceil$.

In the case of $p \to_{T'} q$ a rewrite step with a rule from R'_c gives us $\lfloor p \rfloor = \lfloor q \rfloor$, otherwise it is a translated rule from the old system and we have $\lfloor p \rfloor \to_T \lfloor q \rfloor$. ◀

▶ **Proposition 15.** *Let T be a TRS and T' be its constructor translation. For $t, u \in Ter(\Sigma, X)$, if $t =_T u$ then $\lceil t \rceil =_{T'} \lceil u \rceil$. For $p, q \in Ter(\Sigma 2, Y)$, if $p =_{T'} q$ then $\lfloor p \rfloor =_T \lfloor q \rfloor$.*

Proof. Either way we split the equational proof into individual rewrite steps, and then rebuild these using Lemma 14. ◀

Proposition 15 tells us that we can translate equations back and forth between a TRS and its constructor translation. This has a consequence on consistency.

▶ **Corollary 16.** *Let T be a TRS and T' its constructor translation. Then T is consistent iff T' is.*

Proof. If, say, T is inconsistent, then $x =_T y$, for distinct variables x and y. By Proposition 15 we have $\lceil x \rceil =_{T'} \lceil y \rceil$. But $\lceil x \rceil = x$ and $\lceil y \rceil = y$ and so T' is inconsistent too. The other direction is analogous. ◀

So, the constructor translation preserves and reflects consistency — in the following we really only need that it reflects that property. Aside: regarding termination and confluence, the constructor translation preserves and reflects the former, but only reflects the latter. In general, it does not even preserve weak confluence.

Notice that our construction can fail to produce an almost non-ω-overlapping TRS if our original TRS was merely almost-non-ω-overlapping.

▶ **Example 17.** Consider the following rules describing an if-and-only-if operator on the Booleans: $\{Iff(F, x) \to N(x), Iff(x, F) \to N(x), Iff(x, x) \to N(F), N(N(F)) \to F\}$.

The system is almost non-ω-overlapping, with trivial overlaps between any of the first three rules. However, the constructor translation makes some of the trivial overlaps non-trivial, because F becomes F_c on the left and F_d on the right.

4 Strongly Almost non-ω-overlapping Constructor TRSs

In the Introduction we were mentioning that overlaps created by the constructor translation are "benign". We will now characterise how benign they are more precisely.

▶ **Definition 18.** Two rewrite rules $l_1 \to r_1$, $l_1, r_1 \in Ter(\Sigma, X)$, and $l_2 \to r_2$, $l_2, r_2 \in Ter(\Sigma, Y)$, have a *common generalisation* $l_3 \to r_3$ iff there are substitutions $\sigma_1 : Z \to Ter(\Sigma, X)$, $\sigma_2 : Z \to Ter(\Sigma, Y)$ such that:
- $\sigma_1(l_3) = l_1$ and $\sigma_2(l_3) = l_2$, and $\sigma_1(r_3) = r_1$ and $\sigma_2(r_3) = r_2$,
- all variables in r_3 occur in l_3.

The idea goes back to Plotkin's concept of generalisation and anti-unifiers [12]. Indeed we can check whether two rules have a common generalisation by computing the anti-unifier of the terms $R(l_1, r_1)$ and $R(l_2, r_2)$, and then checking whether the result — which must have the form $R(l_3, r_3)$ — satisfies the final condition on variables.

▶ **Lemma 19.** *If two rewrite rules of a Constructor TRS have a common generalisation $l_3 \to r_3$ then this is either a legal rewrite rule for a Constructor TRS over the same signature, or l_3 is a variable.*

Proof. Proper subterms of l_3 must be constructor terms, otherwise $\sigma_1(l_3) = l_1$ must have non-constructor subterms, contradicting the premise. ◀

We do not need the concrete rewrite system a common generalisation would be part of; all we need is that the rule behaves like a rewrite rule from a Constructor TRS.

▶ **Definition 20.** A TRS is called *strongly almost non-ω-overlapping* iff (i) all ω-overlaps are in root position, (ii) whenever two left-hand sides are ω-unifiable then their rules have a common generalisation.

To justify the chosen terminology:

▶ **Proposition 21.** *Any non-ω-overlapping TRS is strongly almost non-ω-overlapping. Any strongly almost non-ω-overlapping TRS is almost non-ω-overlapping.*

Proof. A non-ω-overlapping TRS clearly satisfies the conditions of being strongly non-ω-overlapping, as its rules can only overlap with themselves, and that at the root.

For the second part, assume we have a strongly almost non-ω-overlapping TRS. Let $\langle\theta_1, \theta_2\rangle$ be a ω-unifier for the left-hand sides l_1, l_2. Then we have $\theta_1(l_1) = (\theta_1 \circ \sigma_1)(l_3)$, and similarly $\theta_2(l_2) = (\theta_2 \circ \sigma_2)(l_3)$. Thus, the composite substitutions $\theta_1 \circ \sigma_1$ and $\theta_2 \circ \sigma_2$ must agree on all variables occurring in l_3. The variable condition on r_3 then gives us $(\theta_1 \circ \sigma_1)(r_3) = (\theta_2 \circ \sigma_2)(r_3)$, and as $(\theta_2 \circ \sigma_2)(r_3) = \theta_2(r_2)$ and $(\theta_1 \circ \sigma_1)(r_3) = \theta_1(r_1)$ we have that $\langle\theta_1, \theta_2\rangle$ is also a ω-unifier for the right-hand sides r_1, r_2. ◀

▶ **Proposition 22.** *The constructor translation of an almost non-ω-overlapping TRS is strongly almost non-ω-overlapping.*

Proof. Because the constructor translation produces a Constructor TRS all ω-overlaps between left-hand sides of rules, if any, are at root position. Let $l_1 \to r_1$ and $l_2 \to r_2$ be two rules in the constructor translation R', such that l_1 and l_2 are ω-unifiable. That implies that $|l_1|$ and $|l_2|$ are ω-unifiable too.

If both rules are translated rules then we can only avoid a contradiction by $|l_1| = |l_2|$ and $|r_1| = |r_2|$ which implies $l_1 = l_2$ and $r_1 = r_2$, as the translation of rules is injective.

If the first rule is a translated rule and the second a pattern rule then $|l_2|$ is a non-variable subterm of a left-hand side in R, and is ω-unifiable with $|l_1|$ which contradicts our assumption about R.

If both rules are pattern rules then both rules clearly have the common generalisation $F_d(x_1, \ldots, x_n) \to F_c(x_1, \ldots, x_n)$, where F is the root symbol of the pattern. ◀

Note: it is not generally true that the constructor translation of a strongly almost non-ω-overlapping TRS is itself strongly almost non-ω-overlapping, because the constructor translation breaks the sharing of subterms of left-hand and right-hand sides, e.g. for the rules $F(C(x), y) \to G(C(x))$ and $F(y, B) \to G(y)$ — the common generalisation $F(y, z) \to G(y)$ of the two rules is not preserved by the constructor translation. One could fix this by providing a more sophisticated translation that maintains the sharing of common subexpressions between left-hand and right-hand sides.

5 Reasoning with Term-Coalgebras

The main purpose of this section is to establish some tools to reason about consistency. These tools are largely relation-algebraic, for relations operating on term-coalgebras, though they could be generalised to arbitrary Σ-coalgebras.

As an additional ingredient to define relations between or on term-coalgebras for a signature Σ we use the following notation: if $R \subset A \times B$, where A and B are term-coalgebras A and B then $\widetilde{R} \subseteq A \times B$ is defined as follows:

$$\forall t \in A.\, \forall u \in B.\, t\, \widetilde{R}\, u \iff \exists F \in \Sigma.\, \exists a_1, \ldots, a_n \in A.\, \exists b_1, \ldots, b_n \in B.$$
$$t = F(a_1, \ldots, a_n) \wedge u = F(b_1, \ldots, b_n) \wedge \forall i.\, a_i\, R\, b_i$$

This concept was first used in [5, 6]; we modified it slightly by removing the reflexivity case. For constructur signatures, we use the notations $\overline{R}$ and $\widehat{R}$ to mean $\widetilde{R}$ for the subsignatures Σ_d and Σ_c, respectively. In particular, $t\ \widehat{id}\ t$ iff the root symbol of t is a constructor, and so $\widehat{R} \cdot \overline{S} = \emptyset$. We still use $\widetilde{R}$ for constructor signatures, to refer to the combined signature; hence $\widetilde{R} = \overline{R} \cup \widehat{R}$.

One can generalise this to arbitrary Σ-coalgebras where the conditions $t = F(a_1, \ldots, a_n)$ and $u = F(b_1, \ldots, b_n)$ would be replaced by $\upsilon_A(t) = F(a_1, \ldots, a_n)$ and $\upsilon_B(u) = F(b_1, \ldots, b_n)$ where υ_A and υ_B are the unfolding maps of their respective coalgebras.

▶ **Proposition 23.** *Some general relation-algebraic properties of* $\widetilde{R}$*:*
1. $\widetilde{R \cap S} = \widetilde{R} \cap \widetilde{S}$*, which moreover implies that the function* $x \mapsto \widetilde{x}$ *is monotonic.*
2. $\widetilde{R^{-1}} = \widetilde{R}^{-1}$.
3. $\widetilde{R \cdot S} \supseteq \widetilde{R} \cdot \widetilde{S}$*. Therefore also:* $\widetilde{R^*} \supseteq \widetilde{R}^*$.

Proof. Trivial. ◀

We have $\widetilde{R \cup S} \supseteq \widetilde{R} \cup \widetilde{S}$ by monotonicity, and it is not an equation because a signature can contain function symbols of arity greater than 1. Also, the relation $\widehat{\emptyset}$ is generally not the empty relation — it will relate all bisimilar nodes that have no subnodes and are topped with a constructor, and therefore also variables.

For term-coalgebras we have $id = \widetilde{id}$, but arbitrary Σ-coalgebras would only give us $id \subseteq \widetilde{id}$, because a coalgebra can contain distinct bisimilar nodes with identical subnodes.

▶ **Definition 24.** A relation R between term-coalgebras is called Σ-closed iff $\widetilde{R} \subseteq R$.

Note: this is standard terminology taken from [2], except that we generalise it to coalgebras.

▶ **Definition 25.** Let $V = \wp(A \times B)$ be the set of relations between term-coalgebras A and B. Then the function $\mathrm{CT} : V \to V$ is defined by $\mathrm{CT}(R) \doteq \mu x.\, R \cup \widetilde{x}$. Thus $\mathrm{CT}(R)$ is the smallest Σ-closed relation containing R.

We can use the $\widetilde{R}$ notation to define $=_R$ in a relation-algebraic way:

▶ **Definition 26.** The inductive congruence closure $\mathrm{CGI}(R)$ of a relation R on a term-coalgebra is defined as: $\mathrm{CGI}(R) \doteq \mu x.\, R \cup x^{-1} \cup (x \cdot x) \cup id \cup \widetilde{x}$.

Thus $=_R$ is then $\mathrm{CGI}(\xrightarrow{\epsilon})$ on the coalgebra $\mathit{Ter}(\Sigma, X)$. Notice that for rewrite systems this is in general not the same as the equivalence closure of rewrite steps, because a coalgebra might lack the intermediate terms. To reason about pattern matching we will later need a stronger notion of consistency, that includes reasoning about constructors:

▶ **Definition 27.** A relation R between term-coalgebras is called *constructor-compatible* iff $\widehat{id} \cdot R \cdot \widehat{id} \subseteq \widehat{R}$.

▶ **Lemma 28.** *Every constructor-compatible relation R between any two term-coalgebras A and B is consistent.*

Proof. Let $x\ R\ y$ where x and y are variables. Since variables in term-coalgebras are viewed as constructors we have $x\ \widehat{id_A}\ x$ and $y\ \widehat{id_B}\ y$. Hence $x\ \widehat{id_A}\ x\ R\ y\ \widehat{id_B}\ y$. Constructor-compatibility of R then gives us $x\ \widehat{R}\ y$ which means $x = y$ since they have no subterms. ◀

▶ **Lemma 29.** *Constructor-compatible relations are closed under arbitrary union. Relational inverse also preserves constructor-compatibility.*

Proof. Let R_i with $i \in I$ be a family of constructor-compatible relations.

$$\widehat{id} \cdot (\bigcup_{i\in I} R_i) \cdot \widehat{id} = \bigcup_{i\in I} (\widehat{id} \cdot R_i \cdot \widehat{id}) \subseteq \bigcup_{i\in I} \widehat{R_i} \subseteq \widehat{\bigcup_{i\in I} R_i}$$

For inverse, for relations between term-coalgebras A and B:

$$\widehat{id_B} \cdot R^{-1} \cdot \widehat{id_A} = ((\widehat{id_B} \cdot R^{-1} \cdot \widehat{id_A})^{-1})^{-1} = (\widehat{id_A} \cdot R \cdot \widehat{id_B})^{-1} \subseteq \widehat{R}^{-1} = \widehat{R^{-1}}$$

◀

Lemma 29 means that constructor-compatibility is a sup-continuous predicate on the lattice of binary relations between two coalgebras. We also have that any relation between term-coalgebras has a constructor-compatible interior — the union of all its subrelations that have this property.

▶ **Definition 30.** Given a Constructor TRS over a signature Σ, a *consistency invariant* is a consistent and Σ-closed relation S on a term-coalgebra A such that for any constructor-compatible equivalence $=_S \subseteq S$ we have $\overset{\epsilon}{\leftarrow} \cdot \overline{=_S} \cdot \overset{\epsilon}{\rightarrow} \subseteq \mathrm{CT}(=_S)$.

Explanation: if we have $a_1 \overset{\epsilon}{\leftarrow} a_2 \overline{=_S}\ a_3 \overset{\epsilon}{\rightarrow} a_4$ then the pair $\langle a_1, a_4\rangle$ can be viewed as a form of "semantical critical pair", because it has been obtained by root-rewrite-steps from $\langle a_2, a_3\rangle$ which share their root symbols and are "semantically equal" below the root. Thus a consistency invariant is characterised by the property that semantical critical pairs stay within the invariant. That this is relative to a term-coalgebra A matters insofar as rewrite steps with contracta outside A are simply discarded.

The reason $=_S$ is locally quantified in the definition of consistency invariant is that although constructor-compatible relations are closed under union, equivalence relations are not, so we cannot simply compute a suitable interior relation. The reason the definition uses constructor-compatible equivalences is that we can turn them into functions that unify the nodes in their equivalence classes.

▶ **Definition 31.** Given a term-coalgebra A, a function $f : A \to Ter^\infty(\Sigma, \emptyset)$ is called *constructor-preserving* iff

$$\forall a_1, \ldots a_n \in A. \forall C \in \Sigma_c.\ f(C(a_1, \ldots, a_n)) = C(f(a_1), \ldots, f(a_n))$$

▶ **Proposition 32.** *Let $=_\varrho$ be a constructor-compatible equivalence relation on a term-coalgebra A. Given some well-ordering on $=_\varrho$-equivalence classes, there is a function $U(=_\varrho) : A \to Ter^\infty(\Sigma, \emptyset)$ such that: (i) $U(=_\varrho)$ is constructor-preserving; (ii) $\forall a, b \in A.\ a =_\varrho b \Rightarrow U(=_\varrho)(a) = U(=_\varrho)(b)$.*

Proof. Let $\min D$ denote the minimum element of any non-empty subset D of any equivalence class w.r.t. that well-order. Let $[a] \subseteq A$ be the $=_\varrho$-equivalence class of some node $a \in A$. Let $B = \{b \in [a] \mid b \mathrel{\widehat{id}} b\}$. Then we define $U(=_\varrho)(a)$ as follows:

$$U(=_\varrho)(a) = \begin{cases} C(U(=_\varrho)(b_1), \ldots, U(=_\varrho)(b_n)) & \text{if } B \neq \emptyset \wedge \min B = C(b_1, \ldots, b_n) \\ \min[a] & \text{if } B = \emptyset \end{cases}$$

Clearly, $U(=_\varrho)$ satisfies condition (ii) because its definition only depends on the equivalence class $[a]$, not on a directly. Let $c \in [a]$ be constructor-topped, i.e. $c \mathrel{\widehat{id}} c$. Thus we have $c \mathrel{\widehat{id}} c =_\varrho \min B \mathrel{\widehat{id}} \min B$: and constructor-compatibility of $=_\varrho$ gives us: $c \mathrel{\widehat{=_\varrho}} \min B$. Therefore $c = C(c_1, \ldots, c_n)$ and $c_i =_\varrho b_i$ and the result follows. ◀

Notice that even if the coalgebra A only contains finite terms the function $U(=_\varrho)$ may still have infinite terms in its range; e.g. this would be the case for the equivalence class $[K, C(K)]$ if C is a constructor.

6 Rewrite-Related Reasoning

We now study some properties of our relations in the presence of pattern matching and rewrite rules. The aim is to establish invariants that “survive” the parallel application of rewrite rules at the root of a term (node).

Besides giving us an ω-unifier (for equivalences), constructor-compatible relations give us an invariant in pattern matching:

▶ **Lemma 33.** *Let $t \in Ter(\Sigma_c, X)$, $s = \sigma(t)$, $u = \theta(t)$, and $s \mathrel{R} u$ where R is a constructor-compatible relation between two term-coalgebras A and B. Then for any $x \in X$ that occurs in t, $\sigma(x) \mathrel{R} \theta(x)$.*

Proof. Let $x \in X$ be any variable occurring in t, i.e. there is some position $p \in \mathrm{Pos}(t)$ such that $t|_p = x$. The proof goes by induction on the length of p.

If $p = \langle\,\rangle$ (the empty position) then $\sigma(t) = \sigma(x)$; similarly, $\theta(u) = \theta(x)$, and so the result follows.

Otherwise, $p = i \cdot p'$ and $t = C(t_1, \ldots, t_n)$, for some constructor $C \in \Sigma_c$. $s = \sigma(t)$ implies $s = C(\sigma(t_1), \ldots, \sigma(t_n))$. Similarly, $u = C(\theta(t_1), \ldots, \theta(t_n))$. Thus, s and u are both constructor-topped, therefore $s \mathrel{R} u$ implies $s \mathrel{\widehat{R}} u$ by constructor-compatibility of R. Hence $\sigma(t_i) \mathrel{R} \theta(t_i)$, and we can apply the induction hypothesis to t_i w.r.t. position p'. ◀

For applying a substitution after matching we have the following result:

▶ **Lemma 34.** *Let R be a Σ-closed relation between two term-coalgebras A and B. Let $t \in Ter(\Sigma, X)$, $s = \sigma(t) \in A$, $u = \theta(t) \in B$. If for all variables $x \in X$ that occur in t we have $\sigma(x) \mathrel{R} \theta(x)$ then $s \mathrel{R} u$.*

Proof. By induction on the term structure of t. If $t \in X$ (it is a variable) then $s = \sigma(t) \mathrel{R} \theta(t) = u$ and the result follows from the assumption.

Otherwise, $t = F(t_1, \ldots, t_n)$, $s = F(\sigma(t_1), \ldots, \sigma(t_n))$, $u = F(\theta(t_1), \ldots, \theta(t_n))$. By induction hypothesis we have $s_i \mathrel{R} u_i$ (for all i), thus $s \mathrel{\widetilde{R}} u$ which entails the result, because R is Σ-closed. ◀

As a direct consequence we can characterise “how safe” parallel rewrite steps with the same rule are in a constructor rewrite system:

▶ **Corollary 35.** *Let S be constructor-compatible, and let $l \to r$ be a rewrite rule of a Constructor TRS. If $t_1 \xleftarrow{l\to r} t_2\, \overline{S}\, t_3 \xrightarrow{l\to r} t_4$ then t_1, t_4 are related by* $\mathrm{CT}(S)$.

Proof. Because l is the left-hand side of a rule of a Constructor TRS, its direct subterms are constructor terms. Thus Lemma 33 applies: the matching substitutions are pointwise related by S. Hence they are also related by $\mathrm{CT}(S)$ and the result then follows from Lemma 34. ◀

Besides non-determinism of $\xrightarrow{\epsilon}$ what might also introduce constructor-inconsistency if between root-rewrite steps we rewrite on subterms in some way that might allow the adjacent root steps to create an inconsistency. However, the situations allowing us to rewrite in opposite directions are limited by the following observation:

▶ **Lemma 36.** *Let $a \to b, c \to d$ be two rewrite rules of a Constructor TRS. Let $=_\varrho$ be a constructor-compatible equivalence on some term-coalgebra A. If $t_1 \xleftarrow{a\to b} t_2 \,\overline{=_\varrho}\, t_3 \xrightarrow{c\to d} t_4$ then a and c are ω-unifiable.*

Proof. We know $t_2 = \sigma(a)$ and $t_3 = \theta(c)$ for some substitutions σ and θ. We know from Proposition 32 that $U(=_\varrho)$ maps the direct subnodes of t_2 and t_3 to the same result.

Because that function is constructor-preserving we have $U(=_\varrho)(t_2|_i) = U(=_\varrho)(\sigma(a|_i)) = (U(=_\varrho)\circ\sigma)(a|_i)$ and similarly $U(=_\varrho)(t_3|_i) = (U(=_\varrho)\circ\theta)(c|_i)$. Thus the pair of maps $\langle U(=_\varrho)\circ \sigma, U(=_\varrho)\circ\theta\rangle$ is an ω-unifier for a and c. ◀

Lemma 36 means that if the reasoning between root steps is done safely, i.e. via a constructor-compatible equivalence on subterms then the subsequent rewrite steps were done with rules with ω-overlapping patterns.

▶ **Theorem 37.** *For a strongly almost non-ω-overlapping Constructor TRS, any Σ-closed consistent relation R on a term-coalgebra A is a consistency invariant.*

Proof. By Lemma 36 parallel rule applications are with ω-unifiable left-hand sides. As the system is strongly almost non-ω-overlapping both are therefore instances of a common generalisation $l_3 \to r_3$, and we can apply Corollary 35. ◀

The standard equivalence relation $=_R$ associated with a Constructor TRS can be expressed as $\mathrm{CGI}(\xrightarrow{\epsilon})$. We want to show that this relation is consistent. Instead, we are going to prove the stronger property that it is constructor-compatible.

To define the right kind of invariant we need another auxiliary function on binary relations over a term-coalgebra which allows us to compose rewrite steps and reasoning on subterms “in a safe way” with another relation.

▶ **Definition 38.** The unary function Ind_A on binary relations over a term-coalgebra A is defined as follows: $\mathrm{Ind}_A(x) \doteq (\xrightarrow{\epsilon}_A \cup \overline{x}) \cdot x$.

▶ **Lemma 39.** $\mathrm{Ind}_A(R)$ *is constructor-compatible.*

Proof. Constructor-topped terms are not in the domain of either $\xrightarrow{\epsilon}_A$ or $\overline{R}$. Hence $\widehat{id_A} \cdot \mathrm{Ind}_A(R) \cdot \widehat{id_A}$ is the empty relation. ◀

Using Ind_A we can construct a suitable consistent relation:

▶ **Definition 40.** Given a TRS with signature Σ, and a term-coalgebra A, the relation $\Downarrow_A$ is a relation on A defined as follows:

$$\Downarrow_A \doteq \mu x.\, f_A(x)$$
$$f_A(x) \doteq x^{-1} \cup \mathrm{Ind}_A(x) \cup id_A \cup \widehat{x} \cup x$$

We omit the index if $A = \mathit{Ter}^\infty(\Sigma + X, \emptyset)$.

▶ **Lemma 41.** *Let A and B be term-coalgebras with $A \subseteq B$. Then $\Downarrow_A \subseteq \Downarrow_B$.*

Proof. We have $id_A \subseteq id_B$ and $\xrightarrow{\epsilon}_A \subseteq \xrightarrow{\epsilon}_B$ simply because $A \subseteq B$. Hence $\mathrm{Ind}_A(x) \subseteq \mathrm{Ind}_B(x)$ and $f_A(x) \subseteq f_B(x)$. The result follows by monotonicity of fixpoint constructions (Proposition 1.2.18 in [1]). ◀

▶ **Proposition 42.** *$\Downarrow_A$ is Σ-closed.*

Proof. $\overline{\Downarrow_A} = \overline{\Downarrow_A} \cdot id_A \subseteq \overline{\Downarrow_A} \cdot \Downarrow_A \subseteq \mathrm{Ind}_A(\Downarrow_A) \subseteq \Downarrow_A$, and $\widehat{\Downarrow_A} \subseteq \Downarrow_A$ is immediate. ◀

In contrast to joinability, the direction of rewrite steps only matters here at root position. Below root we do not rewrite, we just look for the invariant again. This change makes the relation $\Downarrow$ strictly more expressive than $\downarrow$ (in non-confluent systems):

▶ **Example 43.** Recall that in Example 3 we had $A \to_R^* E$ and $A \to_R^* C(E)$, without a common reduct for the two terms. However, we do have $E \Downarrow C(E)$, even $E \Downarrow_B C(B)$ in a strictly finite term-coalgebra B: $B = \{A, E, C(A), C(E), D(A, C(A)), D(C(A), C(A))\}$. Because $A \to_R^* E$ and $C(A) \to_R^* E$ we also have $A \Downarrow_B E$ (and $C(A) \Downarrow_B E$), by symmetry $E \Downarrow_B A$ and so $C(E)\ \overline{\Downarrow_B}\ C(A)$. Because $\Downarrow_B$ is closed under prefixing with $\overline{\Downarrow_B}$ we get $C(E) \Downarrow_B E$.

▶ **Proposition 44.** *For any term-coalgebra A and w.r.t. any Constructor TRS, the relation $\Downarrow_A$ is constructor-compatible.*

Proof. First note that the function f_A preserves constructor-compatibility: We have that the indivual parts of f_A preserve constructor-compatibility (Lemmas 29 and 39), hence $\widehat{id_A} \cdot f_A(x) \cdot \widehat{id_A} \subseteq \widehat{x^{-1}} \cup \widehat{\mathrm{Ind}_A(x)} \cup \widehat{id_A} \cup \widehat{x} \subseteq \widehat{f_A(x)}$.

From Lemma 29 we get that constructor-compatibility is sup-continuous, hence $\mu x. f_A(x)$ is constructor-compatible by Proposition 7. ◀

▶ **Corollary 45.** *Given a strongly almost non-ω-overlapping Constructor TRS, $\Downarrow_A$ is a consistency invariant on any term-coalgebra A.*

Proof. This follows directly from Theorem 37 and Propositions 44 and 42. ◀

7 Proof Graphs

We introduce the new concept of *proof graphs*. The immediate purpose of these structures is to permit us to reason about consistency proofs, and manipulate them, if necessary. The overall goal is to show that $\Downarrow_A$ is an equivalence.

We assume throughout a fixed Constructor TRS $(\Sigma, \mathcal{R})$, and a fixed strongly finite term-coalgebra A.

▶ **Definition 46.** A *proof graph* $\varrho = (\xrightarrow{\varrho}, =_\varrho)$ is given by a binary relation $\xrightarrow{\varrho}$ on A with the following properties:

1. $(\overset{\varrho}{\to} \cup \overset{\varrho}{\leftarrow})^* = =_\varrho \subseteq \Downarrow_A$;
2. $\overset{\varrho}{\to}$ is deterministic, i.e. $\overset{\varrho}{\leftarrow} \cdot \overset{\varrho}{\to} \subseteq id_A$;
3. $\overset{\varrho}{\to}$ is terminating;
4. $\overset{\varrho}{\to} \subseteq \overset{\epsilon}{\to}_A \cup \overline{\Downarrow_A} \cup \widehat{=_\varrho}$

Explanation: the first condition means that a proof graph represents an equivalence relation which is a subrelation of $\Downarrow_A$; the second and third condition means that this representation is a forest of trees (a union/find structure); the fourth condition means that these edges have "good properties" when we want to extend the proof graph and merge equivalence classes.

▶ **Lemma 47.** *Let $\varrho = (\overset{\varrho}{\to}, =_\varrho)$ be a proof graph. Then $=_\varrho$ is constructor-compatible.*

Proof. Let $t =_\varrho u$, where $t\ \widehat{id_A}\ t$ and $u\ \widehat{id_A}\ u$. The first condition of the definition gives us $t\ (\overset{\varrho}{\to} \cup \overset{\varrho}{\leftarrow})^*\ u$; by the second and third condition $\overset{\varrho}{\to}$ there must a common reduct $s \in A$ with $t \overset{\varrho}{\to}{}^* s$ and $s \overset{\varrho}{\leftarrow}{}^* u$. Because the only outgoing edges for constructor-topped nodes are of the relation $\widehat{=_\varrho}$ we have $t\ \widehat{=_\varrho}{}^*\ u$, but $\widehat{=_\varrho}{}^* \subseteq \widehat{(=_\varrho)^*} = \widehat{=_\varrho}$, and so $t\ \widehat{=_\varrho}\ u$. ◀

7.1 Extensions of a Proof Graph

▶ **Definition 48.** A node $t \in A$ is called a *normal form* of ϱ iff it is a normal form of the relation $\overset{\varrho}{\to}$. We write NF_ϱ for the set of normal forms of ϱ. We write $[\varrho](a)$ for the normal form of a node a.

The normal forms of a proof graph represent its equivalence classes. We want a way to merge equivalences classes of a proof graph. In its simplest form (without allowing for "rewiring") this means the following:

▶ **Definition 49.** Given two proof graphs α and β, β is an extension of α iff $\overset{\alpha}{\to} \subseteq \overset{\beta}{\to}$.

▶ **Lemma 50.** *Let β be an extension of α. Then for all $a, b \in A$ with $a \overset{\beta}{\to} b \wedge \neg(a \overset{\alpha}{\to} b)$ we must have: $\neg(a =_\alpha b)$ and $a \in \mathrm{NF}_\alpha$.*

Proof. By contradiction: If $a \notin \mathrm{NF}_\alpha$ then $a \overset{\alpha}{\to} b'$ for some b', hence $a \overset{\beta}{\to} b'$ because $\overset{\alpha}{\to} \subseteq \overset{\beta}{\to}$. But then $\overset{\beta}{\to}$ fails to be deterministic, so β could not be a proof graph. If $a =_\alpha b$ then $b \overset{\alpha}{\to}{}^* a$, because $\overset{\alpha}{\to}$ is deterministic and terminating. Therefore $b \overset{\beta}{\to}{}^* a \overset{\beta}{\to} b$. Thus $\overset{\beta}{\to}$ is not terminating, so β could not be a proof graph. ◀

In addition to that one needs that the new merged equivalence class in $=_\beta$ is still contained in $\Downarrow_A$. However, that turns out not to be an issue because of our restrictions on edges.

▶ **Definition 51.** Given a proof graph $\varrho = (\overset{\varrho}{\to}, =_\varrho)$ the *grey edge relation* on nodes is defined as $\rightsquigarrow_\varrho = \overset{\epsilon}{\to}_A \cup \overline{\Downarrow_A} \cup \widehat{=_\varrho}$.

So grey edges include those that are in that proof graph and those that "might be".

▶ **Lemma 52.** *Let ϱ be a proof graph. Let $\to_\beta \subseteq \rightsquigarrow_\varrho$ such that $\to_\beta$ is deterministic and terminating, and let $=_\beta$ be the equivalence closure of $\to_\beta$. Then $=_\beta \subseteq \Downarrow_A$.*

Proof. If $t =_\beta u$ we must have an $s \in A$ with $t \to_\beta^* s$ and $u \to_\beta^* s$, because $\to_\beta$ is deterministic and terminating. We prove this by induction on the number of $\to_\beta$ steps. Moreover, we strengthen the claim by requiring that if $t\ \widehat{id_A}\ t$ and $u\ \widehat{id_A}\ u$ then $t\ \widehat{=_\varrho}\ u$.

If $t = u$ then by reflexivity of $\Downarrow_A$ we have $t \Downarrow_A u$. If in addition $t\ \widehat{id_A}\ t$ then $t\ \widehat{=_\varrho}\ t$ by reflexivity of $=_\varrho$.

If $t\ (\xrightarrow{\epsilon}_A \cup \overline{\Downarrow_A})\ t' =_\beta u$ then $t' \Downarrow_A u$ by induction hypothesis and so $t\ \mathrm{Ind}_A(\Downarrow_A)\ u$ which implies $t \Downarrow_A u$.

If $t =_\beta t'(\xrightarrow{\epsilon}_A \cup \overline{\Downarrow_A})^{-1} u$ then $t \Downarrow_A t'$ by induction hypothesis, $t' \Downarrow_A t$ by symmetry of $\Downarrow_A$, $u \Downarrow_A t$ by the previous argument, and $t \Downarrow_A u$ by symmetry.

Otherwise, we must have $t\ \widehat{id_A}\ t$ and $u\ \widehat{id_A}\ u$ and $t \mathrel{\widehat{=_\varrho}} t' =_\beta u$. Thus t' is constructor-topped and $t' \mathrel{\widehat{=_\varrho}} u$ by induction hypothesis. This implies $t \mathrel{\widehat{=_\varrho \cdot =_\varrho}} u$ and so $t \mathrel{\widehat{=_\varrho}} u$ by transitivity of $=_\varrho$. Since $=_\varrho \subseteq \Downarrow_A$ we have $\widehat{=_\varrho} \subseteq \widehat{\Downarrow_A} \subseteq \Downarrow_A$ and so $t \Downarrow_A u$. ◀

▶ **Corollary 53.** *Let $\varrho = (\xrightarrow{\varrho}, =_\varrho)$ be a proof graph, let $a \rightsquigarrow_\varrho b$ where $a \in \mathrm{NF}_\varrho$ and $\neg(a =_\varrho b)$. Then $\beta = (\xrightarrow{\varrho} \cup \{(a,b)\}, =_\beta)$ is an extension of ϱ.*

Proof. The conditions on a and b mean that $\xrightarrow{\beta}$ remains deterministic and terminating. Therefore, and because of $\xrightarrow{\beta} \subseteq \rightsquigarrow_\varrho$ we can apply Lemma 52 and get $=_\beta \subseteq \Downarrow_A$. Finally, because $=_\varrho \subset =_\beta$ we also have $\widehat{=_\varrho} \subseteq \widehat{=_\beta}$ and so all constructor edges can be retained. ◀

▶ **Definition 54.** A proof graph is called *complete* if it has no extensions other than itself.

▶ **Proposition 55.** *Every proof graph has a complete extension.*

Proof. Trivial, as each proper extension merges equivalence classes, and A is finite. ◀

▶ **Lemma 56.** *For every complete proof graph ϱ the relation $=_\varrho$ is Σ_c-closed.*

Proof. By contradiction. Suppose ϱ were complete and we had $t \mathrel{\widehat{=_\varrho}} u$ and $\neg(t =_\varrho u)$ then we must also have $t \mathrel{\widehat{=_\varrho}} [\alpha](t)$. This implies $[\alpha](t) \mathrel{\widehat{=_\varrho}} u$. By Corollary 53 $\beta = (\xrightarrow{\varrho} \cup \{(a,b)\}, =_\beta)$ is an extension of ϱ, which contradicts completeness of ϱ. ◀

The corresponding property is generally not true for Σ-closure, because we might only be able to attach new edges from $\overline{\Downarrow_A}$ to nodes that are redexes. But if a proof graph does not contain "too many redexes" then such a conflict does not materialise.

We can characterise proof graphs for which this is possible by considering the relation $\overline{\Downarrow_A}^*$ — which is an equivalence since $\Downarrow_A$ (and therefore $\overline{\Downarrow_A}$) is symmetric.

▶ **Definition 57.** For any $t \in A$ we define $E_t \doteq \{u \in A \mid t\ \overline{\Downarrow_A}^*\ u\}$. We also define $D_A \doteq \{E_t \mid t \in A\}$. $R_t \doteq \{u \in E_t \mid \exists v \in A.\, u \xrightarrow{\epsilon}_A v\}$.

Thus E_t is the equivalence class of t for the relation $\overline{\Downarrow_A}^*$, R_t the subset of redexes of E_t and D_A the collection of equivalence classes. Note that if t is constructor-topped then E_t is a singleton and R_t is empty. We use these notions to build a proof a graph in which redexes and Σ-closure are not in conflict.

▶ **Definition 58.** A *target* is a function $\mathrm{targ} : D_A \to A$ such that the following properties hold: (i) $\forall t \in A.\, \mathrm{targ}(E_t) \in E_t$ and (ii) $\forall t \in A.\, R_t \neq \emptyset \Rightarrow \mathrm{targ}(E_t) \in R_t$.

Thus a target singles out a member of each equivalence class, which moreover must be a redex if the class contains any redexes. The motivation is to build a proof graph whose subgraph on E_t is a tree with root $\mathrm{targ}(E_t)$.

▶ **Definition 59.** A *targeted proof graph* is a pair $(\varrho, \mathrm{targ}_\varrho)$ such that ϱ is a proof graph, targ_ϱ is a target, and we have:

$$\forall t \in A.\, t\ (\overline{\Downarrow_A} \cap \xrightarrow{\varrho})^*\ \mathrm{targ}(E_t) \lor t \in \mathrm{NF}_\varrho$$

$$\forall t \in A, u \in A.\, \mathrm{targ}(E_t) \xrightarrow{\varrho} u \Rightarrow u \notin E_t$$

Thus, in a targeted proof graph a subset of E_t is already connected with root $\mathrm{targ}(E_t)$ whilst all other nodes in E_t are not linked to anything.

▶ **Definition 60.** A *targeted extension* of a proof graph α is a targeted proof graph $(\beta, \mathrm{targ}_\beta)$ such that β is an extension of α.

▶ **Lemma 61.** *Any targeted proof graph has a targeted extension which is complete.*

Proof. Suppose ϱ was targeted and there is a $t \in \mathrm{NF}_\varrho$ such that with $t \neq \mathrm{targ}(E_t)$. Then there are nodes $t = t_0, \ldots, t_n = \mathrm{targ}(E_t)$ such that $\forall i.\, t_i \overline{\Downarrow_A}\, t_{i+1}$ and there must be some $j < n$ such that $t_j \in \mathrm{NF}_\varrho$ and t_{j+1} $(\overline{\Downarrow_A} \cap \xrightarrow{\varrho})^*$ $\mathrm{targ}(E_t)$. Thus we can extend ϱ with the edge (t_j, t_{j+1}), keep the same target, and then complete the extension.

If there is no such node then any extension will remain targeted, and so we can apply Proposition 55. ◀

▶ **Lemma 62.** *If $(\alpha, \mathrm{targ}_\alpha)$ is a complete targeted proof graph then $=_\alpha$ is Σ-closed.*

Proof. By Lemma 56 we know that $=_\alpha$ is Σ_c-closed. Let $t \overline{\equiv_\alpha}\, u$. Because α is complete and targeted $t \overset{\alpha}{\to}{}^* \mathrm{targ}_\alpha(E_t)$ and $u \overset{\alpha}{\to}{}^* \mathrm{targ}_\alpha(E_u)$ Since $t \overline{\equiv_\alpha}\, u$ we have $t \overline{\Downarrow_A}\, u$, hence $E_t = E_u$. ◀

7.2 The Universal Proof Graph

The kind of proof graph we want to build is one whose equivalence is the full relation $\Downarrow_A$, because that would show that $\Downarrow_A$ is transitive.

▶ **Definition 63.** A proof graph ϱ is *universal* iff $=_\varrho = \Downarrow_A$.

▶ **Lemma 64.** *If $\Downarrow_A$ is a consistency invariant for our rewrite system then any targeted proof graph which is complete is universal.*

Proof. Let $(\varrho, \mathrm{targ}_\varrho)$ a target proof graph which is complete. This is universal iff $=_\varrho$ and $\mathrm{CG}(\xrightarrow{\epsilon}_A)$ coincide. We write $=_R$ for $\mathrm{CG}(\xrightarrow{\epsilon}_A)$.

We prove the implication $\forall t, u \in A.\, t \Downarrow_A u \Rightarrow t =_\varrho u$ by induction on the term structure. Because $=_\varrho$ is a Σ-closed equivalence this reduces to $\forall t, u \in A.\, t \xrightarrow{\epsilon}_A u \Rightarrow t =_\varrho u$.

If $t \xrightarrow{\epsilon}_A u$ then $R_t \neq \emptyset$ and so $\mathrm{targ}_\varrho(E_t) \in R_t$ and we have t $(\overline{\Downarrow_A} \cap \xrightarrow{\varrho})^*$ $\mathrm{targ}_\varrho(E_t)$ by completeness of ϱ. Since $\overline{\Downarrow_A}^* \subseteq \overline{\equiv_R}^* \subseteq \overline{\equiv_R}$ we get $t \overline{\equiv_R}\, \mathrm{targ}_\varrho(E_t)$ and so by induction hypothesis $t \overline{\equiv_\varrho}\, \mathrm{targ}_\varrho(E_t)$. Because $\mathrm{targ}_\varrho(E_t) \in R_t$ there is some $r \in A$ with $\mathrm{targ}_\varrho(E_t) \xrightarrow{\epsilon}_A r$ and because of completeness we have $\mathrm{targ}_\varrho(E_t) =_\varrho r$. The consistency invariant property then gives us $u\ \mathrm{CT}(=_\varrho)\ r$. Because $=_\varrho$ is Σ-closed (Lemma 62) we get $u =_\varrho r$ and $t =_\varrho \mathrm{targ}_\varrho(E_t)$. Overall $t =_\varrho \mathrm{targ}\varrho(E_t) =_\varrho r =_\varrho u$. ◀

▶ **Lemma 65.** *If $\Downarrow_A$ is a consistency invariant for our rewrite system then there is a universal proof graph.*

Proof. Let $\mathrm{targ} : D_A \to A$ be any target function. Then $((\emptyset, id), \mathrm{targ})$ is a targeted proof graph. We can then apply Lemma 61 and Lemma 65. ◀

▶ **Theorem 66.** *Let (Σ, R) be a Constructor TRS such that $\Downarrow_A$ is a consistency invariant for any strongly finite term-coalgebra A. Then $=_R$ coincides with $\Downarrow$ on $Ter(\Sigma, \emptyset)$ (and is therefore constructor-compatible).*

Proof. Moreover, we even have that $t =_R u$ implies $t \Downarrow_B u$ for some strongly finite B.

Since $t =_R u$, we must have a sequence of distinct terms $s_1, \ldots, s_n$ with $t = s_1$ and $u = s_n$ such that for all $i < n$ either $s_i \to_R s_{i+1}$ or $s_{i+1} \to_R s_i$. Closing the set $\{s_1, \ldots, s_n\}$ under subterms then gives us the term-coalgebra B. By assumption $\Downarrow_B$ is a consistency invariant for B, therefore there is a universal proof graph for B by Lemma 65 and thus $t \Downarrow_B u$. By the coalgebra-inclusion argument (Lemma 41) we have $t \Downarrow u$. ◀

▶ **Corollary 67.** *Strongly almost non-ω-overlapping Constructor TRSs have a consistent equational theory.*

Proof. Follows from Theorem 66 and Corollary 45. ◀

8 Consequences

Now we can put these results together to deliver the main theorems.

▶ **Theorem 68.** *Non-ω-overlapping TRSs have a consistent equational theory.*

Proof. By Corollary 16 a TRS is consistent iff its constructor translation is. By Proposition 22 the constructor translation of a non-ω-overlapping TRS is strongly almost non-ω-overlapping, which — by Corollary 67 — means that it is consistent. ◀

From this, we also easily get uniqueness of normal forms:

▶ **Theorem 69.** *Non-ω-overlapping TRSs have unique normal forms.*

Proof. By contradiction. Suppose $T = (\Sigma, R)$ were a non-ω-overlapping TRS, $t, u \in Ter(\Sigma, X)$ were normal forms with $t =_R u$ and $t \neq u$. Then they remain normal forms in the TRS $U = (\Sigma + X + \{F\}, R \cup F(t, x, y) \to x, F(u, x, y) \to y)$. The system U is also non-ω-overlapping, as the new rules do not ω-overlap with each other or any old rule. But $x =_U F(t, x, y) =_U F(u, x, y) =_U y$, i.e. U is inconsistent which contradicts Theorem 68. ◀

9 Future Work

We would like to extend the result to wider ranges of TRSs. In particular, it would be nice to be able to extend it to almost non-ω-overlapping Constructor TRSs, as these are the kind of TRSs that are of concern in the Glasgow Haskell compiler which originated our interest [4]. This is almost certainly doable with just slight extensions of the techniques displayed here, though extending this further to arbitrary non-ω-overlapping TRSs might not be as straightforward.

Also of special interest are semi-equational Conditional TRSs as the can be used to turn TRSs into equivalent left-linear CTRSs, and non-duplicating TRSs into linear CTRSs, by transforming variable sharing into equational constraints.

10 Conclusion

We have proved that non-ω-overlapping TRS have a consistent equational theory, as well as unique normal forms. More important than the result itself is the novel proof technique that makes use of finite Σ-coalgebras, in order to show that certain relations are invariants across equational reasoning. The technique is related to the notion of "equivalent reductions" of orthogonal term rewriting, but in contrast does not require "the creation of" additional terms — the terms participating in the consistency proof are the same as the ones of the original equational proof.

References

1. André Arnold and Damian Niwiński. *Rudiments of μ-calculus*. North-Holland, 2001.
2. Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
3. Hendrik P. Barendregt. *The Lambda-Calculus, its Syntax and Semantics*. North-Holland, Amsterdam, 1984.
4. Richard Eisenberg, Dimitrios Vytiniotis, Simon Peyton Jones, and Stephanie Weirich. Closed type families with overlapping equations. In *Principles of Programming Languages*, pages 671–684. ACM, 2014.
5. Jörg Endrullis, Helle Hvid Hansen, Dimitri Hendriks, Andrew Polonsky, and Alexandra Silva. A coinductive treatment of infinitary rewriting. In *Workshop on Infinitary Rewriting*, page 8, 2013.
6. Jörg Endrullis, Dimitri Hendriks, Helle Hvid Hansen, Andrew Polonsky, and Alexandra Silva. A coinductive framework for infinitary rewriting and equational reasoning. In *Rewriting Techniques and Applications*, 2015.
7. Gérard Huet. *Résolution d'Equations dans les Langages d'Ordre 1,2,...,ω*. PhD thesis, Université de Paris VII, 1976.
8. Gérard Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27:797–821, 1980.
9. Richard Kennaway and Fer-Jan de Vries. *Term Rewriting Systems*, chapter Infinitary Rewriting, pages 668–711. Cambridge University Press, 2003.
10. Jan Willem Klop. *Combinatory Reduction Systems*. PhD thesis, Centrum voor Wiskunde en Informatica, 1980.
11. Mizuhito Ogawa and Satoshi Ono. On the uniquely converging property of nonlinear term rewriting systems. Technical Report IEICE COMP89-7, NTT Software Laboratories, 1989.
12. Gordon D. Plotkin. A note on inductive generalisation. *Machine Intelligence*, 5:153–163, 1970.
13. Barry K. Rosen. Tree-manipulating systems and Church-Rosser theorems. *Journal of the ACM*, 20:160–187, 1973.
14. RTA. The RTA list of open problems.
`http://www.cs.tau.ac.il/~nachum/rtaloop`
15. Donald Sannella and Andrzej Tarlecki. *Foundations of Algebraic Specification and Formal Software Development*. Springer, 2012.
16. Robert Endre Tarjan. Efficiency of a good but not linear set union algorithm. *J. ACM*, 22(2):215–225, April 1975.
17. Alfred Tarski. A lattice-theoretic fixed point theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.
18. Terese, editor. *Term Rewriting Systems*. Cambridge University Press, 2003.