The Relative Chain Framework for Modular Termination

Tom.Divers@bristol.ac.uk, Eddie.Jones@bristol.ac.uk

1   Introduction

We present a new framework of **relative chains** for proving the termination of the union of two terminating rewrite systems. Our approach extends the dependency pair framework, introduced by Arts and Giesl in [1]. We demonstrate how our framework can be used to prove the termination of functional programs augmented with equational hypotheses, and **...**

2   Preliminaries

In this section, we discuss some mathematical preliminaries concerning term rewriting and reduction relations. The foundations of this field are explored in depth in [2].

2.1   Term Rewrite Systems

We consider the setting of term rewriting systems over a **signature** $T(\Sigma, V)$. A signature consists of a finite set of **function symbols** Σ and an infinite set of **variables** V. Σ is presumed to be partitioned into a set of **defined functions** Σ_{def} and **constructors** Σ_{con}. Each $f \in \Sigma$ has an associated natural number called its **arity**, and $\Sigma^{(i)}$ denotes all the function symbols in Σ with arity i.

$T(\Sigma, V)$ is defined inductively as follows:

1. $\Sigma^{(0)} \subseteq T(\Sigma, V)$ and $V \subseteq T(\Sigma, V)$.

2. If $f \in \Sigma^{(n)}$, and $M_i \in T(\Sigma, V)$ for each $i \in [n]$, then $f(M_1, \cdots, M_n) \in T(\Sigma, V)$.

The recursive structure of $T(\Sigma, V)$ prompts us to think of terms in a TRS as **syntax trees**. We define $Pos(M) \subseteq \mathbb{N}^*$ for each $M \in T(\Sigma, V)$ to be the set of positions in M's syntax tree. The root of the tree has position ε, and the ith child of a node at position p has position pi. $M|_p$ is the subterm of M rooted at position p, and $M[N]_p$ denotes M with its term at position p replaced by $N \in T(\Sigma, V)$. Additionally, we define $Var(M) \subseteq V$ to be the set of variables appearing in a term. M is **closed** iff $Var(M) = \emptyset$.

A **(one-hole) context** $C[\cdot] : T(\Sigma, V) \to T(\Sigma, V)$ is a term with a hole $\square$ at some position. $C[M]$ denotes the context $C[\cdot]$ with its hole replaced by the term M.

A **substitution** $\theta : X \to T(\Sigma, V)$ is a function mapping variables to terms, for which the set $\text{dom}(\theta) := \{x \in V \mid \theta(x) \neq x\}$ is finite. Hence, we may write $\theta = [x_1 \mapsto M_1, x_2 \mapsto M_2, \cdots]$ for finitely many $x_i \in V$ and $M_i \in T(\Sigma, V)$. $M\theta$ denotes the term M with each variable replaced by its θ-image, and we call $M\theta$ an **instance** of M. For two substitutions θ, σ, we say that σ is **less general** than θ (and write $\sigma \leq \theta$) iff there exists some other substitution θ' such that $\sigma = \theta' \circ \theta$.

A reduction system $(X, \to)$ consists of a set X equipped with a binary relation $\to \subseteq X \times X$. A reduction is **terminating** iff there exist no infinite sequences $x_1 x_2 \cdots \in X^\omega$ with $x_1 \to x_2 \to \cdots$. $x \in X$ is a **redux** of $\to$ iff $x \to y$ for some $y \in X$. If x is not a redux, we say that it is in $\to$-**normal form**. A reduction system over $T(\Sigma, V)$ is a **term rewrite system (TRS)** iff it is closed under contexts and substitutions (i.e. if $M \to N$, then $C[M\theta] \to C[N\theta]$ for all contexts $C[\cdot]$ and substitutions θ).

Here, we will consider TRSs defined by a set of equations $R \subseteq T(\Sigma, V) \times T(\Sigma, V)$. Note that our equations are presumed to be **oriented**, meaning that $M \approx N \in R$ does not imply that $N \approx M \in R$. We also assume that $Var(M) \supseteq Var(N)$ for each $M \approx N \in R$. We define $\to_R$ such that, for each $M \approx N \in R$, and for all contexts $C[\cdot]$ and substitutions θ, $C[M\theta] \to_R C[N\theta]$.

An equation $M \approx N$ is **stable** iff M is headed by a defined function symbol. A TRS is stable iff all of its equations are stable. A TRS is a **functional program** iff it is stable, and for each rule $f(x_1, \cdots, x_n) \approx N$, each x_i contains no defined function symbols. We say that an equation $M \approx N$ is Q-**normal** (where Q is some TRS) iff N is in Q-normal form. A TRS is Q-normal iff all of its rules are Q-normal.

A strict **reduction order** $\succ \subseteq T(\Sigma, V)^2$ is an order on terms that is closed under substitutions. $\succsim$ is defined to be the reflexive closure of $\succ$. A reduction order is also **monotonic** iff it is closed under contexts. $\succ$ is **well-founded** iff every set $K \subseteq T(\Sigma, V)$ has a minimum under $\succ$ (i.e. $\forall K \subseteq T(\Sigma, V), \exists x \in K, \forall y \in K, x \precsim y$).

One particular ordering that will be of use is the subterm ordering $\sqsubset$, which is defined as follows: $M \sqsubset N$ iff $N|_p = M$ for some $p \in Pos(N)$.

2.2 The Dependency Pair Framework

The dependency pair framework [1] allows us to analyse the interdependencies between functions more precisely than by simply analysing the rules of a TRS. A dependency pair of a TRS R is written $\langle M, N \rangle_R \in T(\Sigma \cup \Sigma^\sharp, V)^2$. We define $\Sigma^\sharp$ such that, for each $f \in \Sigma$, we have $F \in \Sigma^\sharp$. These pairs are constructed as follows:

Definition 1 (Dependency pairs). Consider some TRS R, and assume that $f(s_1, \cdots, s_n) \approx C[g(t_1, \cdots, t_m)] \in R$ for some context $C[\cdot]$ and function symbols $f, g \in \Sigma_{def}$. This induces the **dependency pair** $\langle F(s_1, \cdots, s_n), G(t_1, \cdots, t_m) \rangle_R$, where $F, G \in \Sigma^\sharp$. $DP(R)$ is defined to be the set of all dependency pairs induced by R.

The fact that each dependency pair is induced by exactly one equational constraint leads trivially to the following result:

Lemma 2. $DP(R_1 \cup R_2) = DP(R_1) \cup DP(R_2)$

In order to reason about the termination of a TRS through the analysis of its dependency pairs, we need to introduce the following definition, which allows us to connect reduction sequences of our TRSs to sequences of dependency pairs. Theorem 4 then asserts that termination can be equivalently characterised by the non-existence of infinite chains.

Definition 3 (Chains). Consider some fixed TRS R. An R-**chain** is a sequence of dependency pairs $\langle s_1, t_1 \rangle_R \langle s_2, t_2 \rangle_R \cdots$, along with a substitution θ, such that $t_i \theta \rightarrow_R^* s_{i+1} \theta$ for all i.

Theorem 4 (Arts and Giesl [1]). *A TRS is terminating iff it does not induce an infinite chain.*

3 Relative Chains

In order to analyse the termination of the union of two TRSs, we consider the union of their dependency pairs. The following definition generalises the notion of a chain, separating rewrite system used in the intermediate reduction steps from the rewrite system that induces the dependency pairs in the chain.

Definition 5 (Relative chains). Consider two TRSs R and S. A sequence of R's dependency pairs $\langle s_1, t_1 \rangle_R \langle s_2, t_2 \rangle_R \cdots$ forms an R/S-**chain** iff there exists some substitution θ satisfying $t_i \theta \rightarrow_S^* s_{i+1} \theta$ for each i.

We also define what it means for a chain over the union of two rewrite systems to alternate between applying rules in each of the systems:

Definition 6 (Spanning chains). Consider some TRS $S := R_1 \cup R_2$, which is the union of two other TRSs. An S-chain is **spanning** iff it is of the form $\cdots p_1 p_2 \cdots$ where $p_1 \in DP(R_1)$ and $p_2 \in DP(R_2)$, or vice versa. In this case, we say that this chain spans **from R_1 to R_2**.

The following theorem characterises modular termination in terms of sub-relative termination and the nonexistence of two-way spanning chains:

Theorem 7. *Consider two TRSs R_1 and R_2. $S := R_1 \cup R_2$ is terminating if both of the following hold:*

1. *There are no infinite R_1/S- or R_2/S-chains.*

2. *Every spanning S-chain only spans from R_1 to R_2 (or vice versa).*

Proof. We proceed by contrapositive, showing that the nontermination of S implies the failure of one of our conditions. Assume that S is nonterminating; hence, by Theorem 4, S induces an infinite chain. We now demonstrate that any such chain violates one of our conditions. Observe that, since $DP(S)$ is finite, an infinite chain must repeat some finite cycle $s \in DP(S)^*$ infinitely many times; additionally, assume that s is the longest such cycle. Also note that, via Lemma 2, $DP(S) = DP(R_1) \cup DP(R_2)$. Hence, this cycle must occur either entirely within $DP(R_1)$ or $DP(R_2)$, or must alternate between $DP(R_1)$ and $DP(R_2)$. More formally, we have two cases:

1. $s \in DP(R_1)^*$ or $s \in DP(R_2)^*$. Hence, R_1 or R_2 induces an infinite R/S-chain, which violates condition (1).

2. $s = \cdots x_1 x_2 \cdots y_2 y_1 \cdots$, where $x_i, y_i \in DP(R_i), i \in \{1, 2\}$ (i.e. s spans from R_1 to R_2 and from R_2 to R_1), which violates condition (2). $\qquad\square$

4 Flat Termination

The following definition is based on the estimation dependency pair graphs introduced in [1]. Our contribution focuses primarily on studying a termination criterion dependent on the non-existence of certain kinds of cycles in the estimated dependency graph.

Definition 8 (Flat termination). Consider some TRS P over $T(\Sigma, V)$, which induces the dependency pairs $\langle \cdot, \cdot \rangle_P$ over $T(\Sigma \cup \Sigma^\sharp, V)$. Also assume that we have access to an infinite set of unused **flat variables** $V^\flat \subseteq V$.

We define a set of **flat dependency pairs** $DP^\flat(P) \subseteq T(\Sigma_{con} \cup \Sigma^\sharp, V^\flat)^2$, where each rule is obtained by **flattening** a dependency pair. For a dependency pair $\langle s, G(t_1, \cdots, t_n) \rangle_P$, its flattening is written $\langle s, G(t_1^\flat, \cdots, t_n^\flat) \rangle_P^\flat$. A term t's flattening $t^\flat$ is computed recursively as follows (where $x^\flat \in V^\flat$ is an arbitrary unused flat variable):

- If $t \in \Sigma_{con}$, then $t^\flat := t$

- If $t \in V$, or $t = f(b_1, \cdots, b_k)$ where $f \in \Sigma_{def}$, then $t^\flat := x^\flat$.

- If $t = f(b_1, \cdots, b_k)$ where $f \in \Sigma_{con}$, then $t^\flat := f(b_1^\flat, \cdots, b_k^\flat)$.

$P^\flat$- and $P^\flat/Q$-chains are defined as before, but over these flattened dependency pairs. A TRS P is **flat-terminating** iff it induces no infinite $P^\flat$-chains.

Lemma 9. *Every P/Q-chain corresponds to a $P^\flat/Q$-chain. Therefore, if P is flat-terminating, then it is also terminating.*

Lemma 10. *Consider some TRS P, and some other stable TRS S. For any $P^\flat/S$-chain $\langle s_1, t_1 \rangle_P^\flat \langle s_2, t_2 \rangle_P^\flat \cdots$ unified by θ, there exists some $\sigma \leq \theta$ such that $t_i \sigma = s_{i+1} \sigma$ for each i.*

Proof. We construct the substitutions σ_i by considering each reduction step in the chain. For each $t_i \to_S^* s_{i+1}$, we define the substitution $\sigma \leq \theta$. In the case where $t_i \theta = s_{i+1} \theta$, we simply define $\sigma_i := \theta$.

We now consider the case where $t_i \theta \to_S^+ s_{i+1} \theta$. Given that t_i is flat, we can deduce that it does not contain any symbols from Σ_{def}. Also note that S is stable; hence, t_i is not a redux of S. It follows that $\theta(x^\flat)$ must be a redux of S for some flat variable $x^\flat = t_i|_p$. Define $X_i^\flat \subseteq Var(t_i)$ to be the set of all such variables. Observe that, in the process of rewriting $t_i \theta$ to $s_i \theta$, we must reduce one such $\theta(x^\flat)$, transforming it into some other term $(s_{i+1} \theta)|_p \leftarrow_S^+ \theta(x^\flat)$. We choose a σ_i that simultaneously applies all of these sub-reductions for each such member of $X_i^\flat$, and defaults to θ otherwise:

$$\sigma_i(y) := \begin{cases} (s_{i+1}\theta)|_p & \text{if } y \in X_i^\flat \text{ and } \theta(y) \to_S^+ (s_{i+1}\theta)|_p \\ \theta(y) & \text{otherwise} \end{cases}$$

Note that our construction ensures $t_i \sigma_i = s_{i+1} \sigma_i$ for all i. Our global substitution $\sigma := \sigma_1 \circ \sigma_2 \circ \cdots$ is obtained by simply taking the composition of each σ_i.

It remains to show that σ is a valid substitution that rewrites only finitely many variables, even when our chain is infinite. Observe that some suffix of any infinite chain must consist of the same sequence of flattened pairs repeated infinitely many times. Hence, we only have finitely many distinct σ_i; given that each of these has finite domain, their composition must also have finite domain. $\square$

The following trivial corollaries demonstrate the power of this result. The first of these tells us that whether a given TRS is flat-terminating is decidable [1]. The second of these emphasises the fact that a flat-terminating TRS has no infinite chains relative to any other TRS, which is enormously helpful for demonstrating modular termination.

Corollary 11. *A TRS P is flat-terminating iff it induces no infinite $P^\flat/=$-chains.*

Corollary 12. *A flat-terminating TRS P and a stable TRS Q induce no infinite $P^\flat/Q$-chains.*

We now prove a result motivated by applying equational reasoning techniques to program-hypotheses pairs. We presume that our program P is some stable, flat-terminating TRS, and we also assert that a set of hypotheses H is some other stable, P-normal TRS over a collection of universally quantified variables. These variables are represented as a subset of constructors $\Sigma_\forall \subseteq \Sigma_{con}$. The idea here is that we can substitute any variable with a universally-quantified one, but not vice-versa. Hence, a valid hypothesis is closed. This leads us to the following result concerning the union of programs and hypotheses:

Theorem 13. *If P and H are both stable, P is flat-terminating, and H is terminating, P-normal and closed, then $S := P \cup H$ is terminating.*

Proof. By Theorem 7, our result follows from the conjunction of the following:

1. There are no infinite H/S-chains.

2. All of S's spanning chains span from P to H.

3. There are no infinite P/S-chains.

From the fact that H is P-normal and closed, we can deduce that, for each $l \approx r \in H$ and for any substitution θ, $r\theta$ is in P-normal form. This, combined with H's termination, leads to (1). We can also deduce that each $\langle s_H, r_H \rangle_H \langle s_P, r_P \rangle_P \in DP(H) \times DP(P)$ cannot form a chain under S, which leads to (2). By Corollary 12, and from the fact that S is the union of two stable TRSs (and is therefore also stable), we have that there are no infinite $P^\flat/S$-chains. By Lemma 9, we deduce that there are no infinite P/S-chains, and have thus shown (3). $\qquad\square$

5 Flat Termination as Size-Change Termination

We begin by adapting a result from [1] concerning the existence of program orderings.

Theorem 14. *A TRS P is flat-terminating iff there exists a well-founded monotonic reduction order $\succ$ that satisfies*

- *$l \succsim r$ for each $l \approx r \in P$*

- *$s \succ t$ for each $\langle s, t \rangle_P^\flat \in DP^\flat(P)$*

This theorem will be of use in our demonstration that flat termination is equivalent to **size-change termination**.

Definition 15 (Size-change termination [3, 4])**.** Consider some functional program $P \subseteq T(\Sigma_{con} \cup \Sigma_{def}, V)^2$. For each rule $f(s_1, \cdots, s_n) \approx N$, and for each function symbol g such that $N = C[g(t_1, \cdots, t_m)]$ for some context C, we define a directed bipartite **size-change graph** $G_f^g := (L, R, E)$ with respect to some ordering on closed constructor terms $\succ \subseteq T(\Sigma_{con}, \emptyset)^2$.

G_f^g contains a vertex for each argument of f and g (i.e. $L = \{1_f, \cdots, n_f\}$ and $R = \{1_g, \cdots, m_g\}$). Each of G_f^g's directed edges $e \in L \times R \times \{\succ, \succsim\}$ is labelled either with $\succ$ or $\succsim$. For each pair of arguments s_i and t_j, if $s_i \succ t_j$, we have $(i_f, j_g, \succ) \in E$. Otherwise, if $s_i \succsim t_j$, we have $(i_f, j_g, \succsim) \in E$.

We define the concatenation of two size-change graphs $G_f^g \circ G_g^h$ to be the graph obtained by .

P is defined to be **size-change terminating** iff some infinite path of every concatenation of an infinite sequence of size-change graphs contains infinitely many edges labelled $\succ$.

Observe that, like flat termination, this formalism of size-change termination is also stronger than termination. If a program is size-change terminating, then any possible infinite reduction sequence infinitely decrements some closed constructor term with respect to $\succ$, which violates $\succ$'s well-foundedness.

Lemma 16. *A program P is size-change terminating iff any maximal multigraph contains an edge labelled $\succ$.*

Theorem 17. *A functional program P is size-change terminating (with respect to the closed constructor subterm ordering $\sqsubset$) iff it is flat terminating.*

Proof. ($\Rightarrow$) Consider some size-change terminating program P. By Lemma 16, any maximal size-change multigraph $G = G_1^2; G_2^3; \cdots; G_{n-1}^1$ of P must contain an edge labelled $\succ$.

Firstly, we observe that G corresponds to a cycle of dependency pairs. We now observe that

finish this direction (but it's fairly trivial) – infinite edge traversal violates well-foundedness of $\sqsubset$

($\Leftarrow$) Consider some flat terminating program P, and presume that it is not size-change terminating with respect to $\sqsubset$. Consider any finite flat chain $C := \langle F(s_1, \cdots, s_n), g_1 \rangle_P^\flat \cdots \langle g_l, F(t_1, \cdots, t_n) \rangle_P^\flat$. From P's flat termination, we can deduce that C^ω cannot also be a flat chain. Hence, for any substitution σ, we have $t_i\sigma \neq s_i\sigma$ for each $i \in [n]$.

From the fact that P is not size-change terminating, we also know that $t_i\sigma \not\sqsubset s_i\sigma$ for each $i \in [n]$. We also know that each s_i and t_i must be flat; hence, $t_i\sigma \sqsupset s_i\sigma$. This violates P's termination (implied by its flat termination), which is contradictory. $\qquad\square$

References

[1] Thomas Arts and Jürgen Giesl. "Termination of term rewriting using dependency pairs". In: *Theoretical Computer Science* 236.1 (2000), pp. 133–178. DOI: 10.1016/S0304-3975(99)00207-8.

[2] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998. DOI: 10.1017/CBO9781139172752.

[3] Chin Soon Lee, Neil D. Jones, and Amir M. Ben-Amram. "The size-change principle for program termination". In: *SIGPLAN Not.* 36.3 (2001), pp. 81–92. DOI: 10.1145/373243.360210.

[4] René Thiemann and Jürgen Giesl. "The size-change principle and dependency pairs for termination of term rewriting". In: *AAECC* 16 (2005), pp. 229–270. DOI: 10.1007/s00200-005-0179-7.