

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. 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 [1].

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 $i \in [n]$, $n \geq 0$, then $f M_1 \dots M_n \in T(\Sigma, V)$.

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 i th 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** $C[\cdot] : T(\Sigma, V) \rightarrow T(\Sigma, V)$ 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** $\theta : X \rightarrow T(\Sigma, V)$ 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 $\theta = [x_1 \mapsto M_1, x_2 \mapsto M_2, \dots]$ 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, \rightarrow) consists of a set X equipped with a binary relation $\rightarrow \subseteq X \times X$. A reduction is **terminating** iff there exist no infinite sequences $x_1 x_2 \dots \in X^\omega$ with $x_1 \rightarrow x_2 \rightarrow \dots$. $x \in X$ is a **redux** of \rightarrow iff $x \rightarrow y$ for some $y \in X$. If x is not a redux, we say that it is in **\rightarrow -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 $M \rightarrow N$, then $C[M\theta] \rightarrow C[N\theta]$ 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 \rightarrow_R such that, for each $M \approx N \in R$, and for all contexts $C[\cdot]$ and substitutions θ , $C[M\theta] \rightarrow_R C[N\theta]$.

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 $f(x_1, \dots, x_n) \approx N$, 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 TRS \rightarrow also defines **innermost** and **outermost reduction systems** \xrightarrow{i} and \xrightarrow{o} . Consider some term M that is a redux of \rightarrow . In an innermost reduction sequence, we ...

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. $\forall K \subseteq T(\Sigma, V), \exists x \in K, \forall y \in K, x \preceq y$).

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 $p \in \text{Pos}(N)$.

3 Flat Termination

We introduce a definition of program termination inspired by the dependency pair framework which we prove to be a generalisation of **size-change termination** [2, 3].

Definition 1 (Flat termination). Consider some term $x \in T(\Sigma_{\text{def}} \cup \Sigma_{\text{con}}, V)$, and assume the existence of an infinite list of free variables V^b . We define x 's **flattening** x^b as follows:

- If $x \in \Sigma_{\text{con}}$, then $x^b := x$.
- If $x = f \ x_1 \ \cdots \ x_n$ for some $f \in \Sigma_{\text{con}}$, then $x^b := f \ x_1^b \ \cdots \ x_n^b$.
- If $x = f \ x_1 \ \cdots \ x_n$ for some $f \in \Sigma_{\text{def}}$, or if $x \in V$, then $x^b := v^b$, where v^b is the next unused variable in V^b .

Now consider some TRS $P \in T(\Sigma_{\text{def}} \cup \Sigma_{\text{con}}, V)^2$. We define its **flattening** P^b as follows:

$$P^b := \{f \ s_1 \ \cdots \ s_n \approx g \ t_1^b \ \cdots \ t_n^b \mid f \ s_1 \ \cdots \ s_n \approx C[g \ t_1 \ \cdots \ t_n] \in P\}$$

Lemma 2. *For any functional program P , if P^b is terminating, then P is terminating.*

Proof. Consider some infinite reduction sequence $t_1 \rightarrow_P t_2 \rightarrow_P \cdots$. Each term in this sequence must be some instance of the LHS of a rule in P . Hence, our sequence is of the form $C[f_1 \ a_1] \rightarrow_P C[f_2 \ a_2] \rightarrow_P \cdots$. \square

Definition 3 (Size-change termination [2, 3]). Consider some functional program $P \subseteq T(\Sigma_{\text{con}} \cup \Sigma_{\text{def}}, V)^2$. 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 constructor terms $\succ \subseteq T(\Sigma_{\text{con}}, V)^2$.

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

We define the concatenation of two size-change graphs $G_f^g \circ G_h^g$ to be the graph obtained by unifying all of G_f^g 's input vertices with all of G_h^g 's output vertices.

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.

Also observe that this formalism ensures that any two functional programs P and P^b must have the same set of size-change graphs. Hence, P is size-change terminating iff P^b is also size-change terminating.

Theorem 4. *Consider some functional program P . If P is size-change terminating with respect to \succ , P^b is also size-change terminating with respect to \succ .*

Proof. Assume that P is size-change terminating. Then in any call sequence of the form $G := (G_1^2 \circ G_2^3 \circ \cdots \circ G_{l-1}^1)^*$ contains an edge labelled \succ . Observe that this call sequences corresponds to some sequence of rules in P^b . \square

References

- [1] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998. DOI: 10.1017/CB09781139172752.
- [2] 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.
- [3] 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.