On Disproving Termination of Constrained Term Rewriting Systems

Naoki Nishida
Nagoya University
Nagoya, Japan

Masahiko Sakai
Nagoya University
Nagoya, Japan

Tatsuya Hattori
Nagoya University
Nagoya, Japan

Abstract

This paper shows a sufficient condition for non-termination of constrained term rewriting systems. Constrained rewrite rules such that the right-hand sides are encompassments of the left-hand sides do not always cause non-termination, while they cause non-termination if they are unconstrained. For such constrained rewrite rules, we characterize additional constraints that ensure non-termination caused by the constrained rewrite rules. We also show that such additional constraints sometimes can be obtained from the constraints of the constrained rewrite rules, by removing some closures from the disjunctive normal forms of the constraints.

1 Introduction

Constrained (un)conditional term rewriting systems are sets of constrained (un)conditional rewrite rules [21, 15, 18, 7, 9, 5, 12, 20]. For such systems, the constraint parts of rules are evaluated by some built-in semantics given by the membership relation, linear integer arithmetics (so-called Presburger arithmetics), equational theories independent on the rewrite rules, and so on. The condition parts of rules are evaluated by the rewrite rules recursively. Constrained systems have been enriched such as constrained equational systems (CESs, for short) [10, 11], that are rewrite systems with built-in numbers and semantic data structures. This paper deals with constrained unconditional term rewriting systems (constrained TRSs, for short) defined in [5, 12, 20].

Theorem proving methods for constrained (un)conditional TRSs are investigated in several ways [6, 14, 9, 12, 5, 20]. In these methods, termination of given constrained systems has to be guaranteed (or proved in advance). Moreover, before starting the process of theorem proving, users have to specify reduction orders to be consistent with the reduction of the given systems. The methods proceeds, orienting equations by means of the reduction orders at each step of the expansion operation, and carrying them as hypotheses. All the combined systems of the given system and intermediate hypotheses must be consistent with the reduction orders in order to guarantee correctness of proofs obtained by the methods.

It is very difficult to know in advance which reduction orders are adequate for given constrained systems and equation sets to be proved, similarly to completion procedures [3]. To avoid this difficulty, termination provers are sometimes powerful on behalf of reduction orders, similarly to completion procedures [22]. At each step of the orientation, the methods orients almost all equations temporarily in a direction, combines each of them with the given system and the adopted hypotheses, and adds one of the temporarily oriented equations such that the termination provers succeeds in proving termination of the corresponding combined systems, into the adopted hypothesis set. This operation guarantees that, if the process finishes successfully, then the reduction in itself by the combined system consisting of the given system and all the adopted hypotheses is capable to be an alternative to the reduction orders that the users should have initially given to the theorem proving methods. Especially, in the case of constrained TRSs (as shown later or in [12, 20, 11]) obtained from imperative programs with while-loops, the approach of employing termination provers is often helpful because any path-based order such as lexicographic path order is hardly helpful in proving termination of such constrained TRSs.

When termination provers are employed in theorem proving methods, at each step of the expansion, the termination provers are invoked great many times and they are sometimes applied to non-terminating systems because the methods orients almost all equations temporarily in both directions until the methods

*This work is partly supported by MEXT. KAKENHI #20300010, #20500008, and #21700011.
find an appropriate oriented equation. In each execution, the termination provers tend to spend much
time on trying to prove termination of systems even if the systems are not terminating. Moreover, if
the termination provers fail to disprove termination, then the termination provers must wait for either the
ending of all the processes or the timeout. For this reason, improvements of disproving power of the
termination provers is promising for making the theorem proving methods efficient.

The DP framework [2, 14, 13], one of the termination proof techniques for unconstrained term rewrit-
ing systems (TRSs, for short), has been extended for the class of CESSs [10, 11]. To prove termination
of constrained TRSs, we can use the results in [10, 11], by adapting constrained TRSs to CESSs. On the
other hand, no sufficient condition for disproving termination of constrained systems is known, except
for the detection of unconstrained rewrite rules of the form \( t \rightarrow C[t\theta] \). It would be possible to extend this
method for constrained rules \( t \rightarrow C[t\theta] \) \([\phi]\), adding the side condition that \( \phi \) is valid. However, such an
extension is not very powerful because the constraints of rules are rarely valid.

This paper relaxes the side condition that \( \phi \Rightarrow \phi \theta \) is valid. The essential idea is to find an additional
satisfiable constraint \( \eta \) such that \( \eta \Rightarrow \phi \) and \( \eta \Rightarrow \eta \phi \) are valid.

One may think that the validity of \( \phi \Rightarrow \phi \theta \) is sufficient. The constrained rule \( t \rightarrow C[t\theta] \) \([\phi]\) causes non-termination if there is a substitution \( \sigma \) such that \( \sigma \rightarrow C[\sigma t\theta \sigma] \) and the reductions \( t \theta \sigma \rightarrow C \theta \theta \sigma t[\theta \theta \sigma \theta \sigma] \), \( t \theta \sigma \rightarrow C \theta \theta \sigma t[\theta \theta \sigma \theta \sigma] \), \( \cdot \cdot \cdot \) holds. These reductions are provided by the validity of \( \phi \Rightarrow \phi \theta \). It is regret-
ted, however, that this is not practical yet. For example, consider the following constrained TRS \( R_{sum}
\) obtained by transforming the following imperative program [12] (or [11], adapting CESSs to constrained
TRSs and adding the rules for the entry function \( \text{sum} \) and the \( \text{return-statement} \):

\[
\text{int } \text{sum} = \begin{cases}
\text{int } x; \\
\text{int } i = 0, z = 0; \\
\text{while}( x != i ) 
\quad \text{\{ } \\
\quad \quad \text{z }+\times i+1; \\
\quad \quad i ++; \\
\quad \text{\} } \\
\text{return } z; 
\end{cases}
\]

Here, the function symbols \( 0, s, p, \) and \( \text{plu} \) represent linear arithmetic expressions over integers as usual, and the predicate symbol \( \triangleright \) is interpreted by \( > \) over integers. For a non-negative integer \( n \), \( \text{sum}(s^n(0)) \) computes the summation from \( 0 \) to \( n \), e.g., \( \text{sum}(s^{10}(0)) \) is reduced by \( R_{sum} \) to \( s^{55}(0) \). Terms obtained by applying the entry function \( \text{sum} \) to terms over \( \{0, s(), \text{plu}(), \} \) are terminating but \( R_{sum} \)
is not terminating, e.g., \( \text{eval}(0, s(0), 0) \) is not terminating. On the other hand, the constrained TRS obtained
from \( R_{sum} \) by replacing \( x \triangleright i \triangleright i \triangleright x \) by \( x \triangleright i \) is terminating. Thus, it is not true that any
constrained rewrite rules of the form \( t \rightarrow C[t\theta] \) \([\phi]\) cause non-termination. Next, consider the second
rule \( \text{eval}(x, i, z) \rightarrow \text{eval}(x, s(i), \text{plu}(z, s(i))) ) [x \triangleright i \triangleright i \triangleright x] \), a substitution \( \theta = \{ i \rightarrow s(i), z \rightarrow \text{plu}(z, s(i)) \} \),
and a substitution \( \sigma = \{ x \rightarrow s(0), i \rightarrow 0, z \rightarrow 0 \} \). The reduction \( \text{eval}(x, i, z) \sigma = \text{eval}(s(0), 0, 0) \rightarrow R_{sum}
\text{eval}(s(0), s(0), \text{plu}(0, s(0))) \) is \( \text{eval}(x, i, z) \theta \sigma \) holds but \( \text{eval}(x, i, z) \theta \sigma \rightarrow R_{sum} \text{eval}(x, i, z) \theta \sigma \) does not
hold. In fact, \( x \triangleright i \triangleright i \triangleright x \) \( \Rightarrow \) \( x \triangleright i \triangleright i \triangleright x \) is not valid.

Recall what is sufficient for non-termination, i.e., the validity of \( \phi \sigma, \phi \theta \sigma, \phi \theta \theta \sigma, \cdot \cdot \cdot \). This is not
true for all substitutions \( \sigma \) satisfying \( \phi \) but true for some of them. For example, given \( \sigma = \{ x \rightarrow 0, i \rightarrow s(0) \} \), all of the constraints \( (x \triangleright i \triangleright i \triangleright x) \sigma, (x \triangleright i \triangleright i \triangleright x) \theta \sigma, (x \triangleright i \triangleright i \triangleright x) \theta \sigma \sigma, \cdot \cdot \cdot \) are valid, i.e.,
\( \text{eval}(x, i, z) \sigma \) is not terminating. Actually, this holds for any substitutions \( \sigma \) such that \( (i > x) \sigma, (i > x) \theta \sigma, (i > x) \theta \sigma \sigma, \cdot \cdot \cdot \) are valid. Thus, for such substitutions \( \sigma \), all of the constraints \( (x \triangleright i \triangleright i \triangleright x) \sigma, \cdot \cdot \cdot \) are valid because the validity of \( (i > x) \theta \sigma \sigma \) implies the
validity of \( (x \triangleright i \triangleright i \triangleright x) \theta \sigma \sigma \cdot \cdot \cdot \). To conclude, a constrained rule \( t \rightarrow C[t\theta] \) \([\phi]\) causes non-termination
if there exist a substitution \( \sigma \) and a constraint \( \eta \) such that \( \eta \sigma, \eta \theta \sigma, \eta \theta \theta \sigma, \cdot \cdot \cdot \) are valid.

By replacing the existence of \( \sigma \) by the satisfiability of \( \eta \), the sufficient condition is translated into the
one that there exists a satisfiable constraint \( \eta \) such that \( \eta \Rightarrow \phi \) and \( \eta \Rightarrow \eta \theta \) are valid: for a substitution \( \sigma \)
such that \( \eta \sigma' \) is valid, the validity of \( \eta \Rightarrow \eta \theta \) implies the validity of \( \eta \theta \sigma', \eta \theta \theta \sigma', \cdots \) and the validity of \( \eta \Rightarrow \phi \) implies the validity of \( \phi \theta \sigma', \phi \theta \theta \sigma', \cdots \).

This paper shows a sufficient condition for non-termination of constrained TRSs, following the above observation. We also show a heuristics to find additional constraints (such as \( \eta \) above) for constrained rules of the form \( t \rightarrow C[t \theta] [\phi] \) if the rules cause non-termination. For the sake of simplicity, we discuss on the simpler class than the class of CESs. Though, we believe that the results in this paper holds for more complicated classes, such as CESs.

We assume familiarity with the basic concepts and notations of term rewriting systems (TRSs, for short) \([5, 19]\), and first-order predicate logic \([16]\).

## 2 Constrained Term Rewriting Systems

In this section, we recall the definition of constrained term rewriting systems \([18, 4, 5, 12, 20]\).

Let \( \mathcal{G} \) be a signature such that \( \mathcal{G} \) has at least a constant (i.e., \( \mathcal{F}(\mathcal{G}) \neq \emptyset \)), and \( \mathcal{P} \) be a finite set of predicate symbols. We sometimes denote \( \neg \phi_1 \lor \phi_2 \) by \( \phi_1 \Rightarrow \phi_2 \) as usual. Let \( \mathcal{M} \) be a structure for formulas over \((\mathcal{G}, \mathcal{P}, \mathcal{Y})\), where the interpretation of closed formulas \( \phi \) by \( \mathcal{M} \), denoted by \( \mathcal{M} \models \phi \), is defined as usual. We suppose that for every element \( a \) in the universe of \( \mathcal{M} \), there exists a term \( t \in \mathcal{F}(\mathcal{G}) \) such that \( (t)^\mathcal{M} = a \). The validity and satisfiability of formulas are defined as usual.

When \( \mathcal{G}, \mathcal{P}, \) and \( \mathcal{M} \) are fixed in context, we may call formulas over \((\mathcal{G}, \mathcal{P}, \mathcal{Y})\) constraints (w.r.t. \( \mathcal{M} \)). We assume that for each structure we use, there exist decision algorithms for the validity (and satisfiability) w.r.t. the structure. For example, an algorithm for the truth value of closed (quantified) formulas for linear integer arithmetics is well-known \([8]\), that can be used for deciding the validity and satisfiability. This assumption is usual in recent frameworks of constrained systems \([1, 10]\).

Let \( \mathcal{G} \) and \( \mathcal{F} \) be signatures such that \( \mathcal{F} \cap \mathcal{G} = \emptyset \) and \( \mathcal{G} \) contains at least a constant, \( \mathcal{P} \) be a finite set of predicate symbols, and \( \mathcal{M} \) be a structure for \((\mathcal{G}, \mathcal{P})\). A constrained rewrite rule over \((\mathcal{G}, \mathcal{P}, \mathcal{Y}, \mathcal{M})\) is a triple \((l, r, \phi)\), written as \( l \rightarrow r [\phi] \), such that \( l \) and \( r \) are terms in \( \mathcal{F}(\mathcal{G} \cup \mathcal{P}, \mathcal{Y}) \), \( l \) is not a variable, \( \phi \) is a quantifier-free constraint w.r.t. \( \mathcal{M} \) (i.e., a formula over \((\mathcal{G}, \mathcal{P}, \mathcal{Y})\)), and \( \text{Var}(l) \supseteq \text{Var}(r) \cup \text{fv}(\phi) \), and \( \phi \) is satisfiable w.r.t. \( \mathcal{M} \). We may write \( l \rightarrow r \) instead of \( l \rightarrow r [\top] \). A constrained term rewriting system (constrained TRS, for short) over \((\mathcal{G}, \mathcal{P}, \mathcal{Y}, \mathcal{M})\) is a finite set of constrained rewrite rules over \((\mathcal{G}, \mathcal{P}, \mathcal{Y}, \mathcal{M})\). The rewrite relation \( \rightarrow_R \) is defined by \( \{(C[\sigma]_p, C[r\sigma]_p) \mid l \rightarrow r [\phi] \in R, \text{Ran}(\sigma)|_{\text{fv}(\phi)} \subseteq \mathcal{F}(\mathcal{G}, \mathcal{Y}), \phi \sigma \text{ is valid w.r.t. } \mathcal{M}\} \).

Example 1. Let a signature \( \mathcal{G}_{PA} = \{0, s(\cdot), p(\cdot), \text{plus}(\cdot)\} \), a predicate set \( \mathcal{P}_{PA} = \{\geq (\cdot, \cdot)\} \), and a structure for \((\mathcal{G}_{PA}, \mathcal{P}_{PA})\) such that the universe of \( \mathcal{M}_{PA} \) is the set \( \mathbb{Z} \) of integers, \( 0^\mathcal{M}_{PA} = 0, s^\mathcal{M}_{PA}(x) = x + 1, p^\mathcal{M}_{PA}(x) = x \leq 1 \), \( \text{plus}^\mathcal{M}_{PA}(x_1, x_2) = x_1 + x_2 \), and \( \geq^\mathcal{M}_{PA} = \{(m, n) \mid m, n \in \mathbb{Z}, m > n\} \). The constrained TRS \( R_{\text{sum}} \) in Section 1 is over \( \{(\text{sum}(\cdot), \text{eval}(\cdot), \cdot)\}, \mathcal{G}_{PA}, \mathcal{P}_{PA}, \mathcal{Y}, \mathcal{M}_{PA}\).

In contrast to conditional TRSs \([19]\), no constrained rewrite rule is used for evaluating the truth value of instanitiated constraints in the rewrite relation. For this reason, the termination is different from operational termination of conditional TRSs \([17]\), that is finiteness of derivation trees.

## 3 Sufficient Condition for Non-Termination of Constrained TRSs

Throughout this section, we assume that \( R \) is a constrained TRS over \((\mathcal{G}, \mathcal{P}, \mathcal{Y}, \mathcal{M})\). Before showing the main result based on the observation in Section 1 we start with adapting a well-known sufficient condition for non-termination of unconstrained TRSs to constrained systems. The well-known sufficient condition is the detection of the reduction \( t \rightarrow^+ C[t \theta] \). If the range of \( \theta \) is over terms in \( \mathcal{F}(\mathcal{G}, \mathcal{Y}) \), then this is also a sufficient condition for non-termination of constrained TRSs because the reduction is closed under contexts and substitutions whose ranges are over terms in \( \mathcal{F}(\mathcal{G}, \mathcal{Y}) \): if there exists a reduction \( t \rightarrow_R C[t \theta] \) such that \( \text{Ran}(\theta) \subseteq \mathcal{F}(\mathcal{G}, \mathcal{Y}) \), then \( R \) is not terminating.

A simple way to find the reduction \( t \rightarrow^+ C[t \theta] \) is to find a rewrite rule of the form \( t \rightarrow C[t \theta] \). In contrast, as shown in Section 1, this approach is not powerful for the case of constrained TRSs and thus
we have to find additional constraints $\eta$ (as shown in Section 1) for constrained rules $t \rightarrow C[t\theta] [\phi]$ that cause non-termination. The role of $\eta$ is to narrow down the original constraint $\phi$.

**Theorem 1.** $R$ is not terminating if there exist a constrained rewrite rule $t \rightarrow C[t\theta] [\phi] \in R$ and a constraint $\eta$ such that $\mathcal{Ran}(\theta|_{\text{Ev}(\theta)}) \subseteq \mathcal{T}(\mathcal{G}, \mathcal{V})$, $\eta$ is satisfiable w.r.t. $\mathcal{M}$, and $\eta \Rightarrow \eta \theta$ and $\eta \Rightarrow \phi$ are valid w.r.t. $\mathcal{M}$.

As the well-known sufficient condition is based on the detection of a reduction $t \rightarrow^+ C[t\theta]$, we would like to extend Theorem 1 so as to be based on the detection of reductions that cause non-termination. However, this seems impossible because constraints such as $\phi$ and $\eta$ cannot be formalized for the reduction sequences. Since $\eta$ in Theorem 1 can be considered as an environment where $t \rightarrow C[t\theta] [\phi]$ is always applicable to any instance of $t$, we extend Theorem 1 for the reduction under constraints.

To this end, we review the reduction under constraints. Let $\psi$ be a constraint that is satisfiable w.r.t. $\mathcal{M}$. Then, the constrained reduction $\rightarrow_{\psi, \mathcal{R}}$ under $\psi$ is defined as follows: $\rightarrow_{\psi, \mathcal{R}} = \{(C[l\sigma], C[r\sigma]) \mid l \rightarrow r [\phi] \in R, \psi \Rightarrow \phi \sigma \text{ is valid w.r.t. } \mathcal{M}\}$. Note that $\rightarrow_{\psi, \mathcal{R}}$ and $\rightarrow_{\theta, \mathcal{R}}$ are not identical. It is clear that $s \rightarrow_{\psi, \mathcal{R}} t$ implies $s\theta \rightarrow_{\theta, \mathcal{R}} t\theta$ for any substitution $\theta$ such that $\psi\theta$ is valid w.r.t. $\mathcal{M}$. Thus, to show non-termination of $\rightarrow_{\mathcal{R}}$, it suffices to show non-termination of $\rightarrow_{\psi, \mathcal{R}}$ for some $\psi$. Theorem 1 is extended to constrained reductions as follows.

**Theorem 2.** Let $\eta$ be a constraint. Then, there exists an infinite derivation of $\rightarrow_{\eta, \mathcal{R}}$ (i.e., $R$ is not terminating) if there exist a term $t$, a context $C$, and a substitution $\theta$ such that $\mathcal{Ran}(\theta|_{\text{Ev}(\theta)}) \subseteq \mathcal{T}(\mathcal{G}, \mathcal{V})$, $t \rightarrow_{\eta, \mathcal{R}}^+ C[t\theta]$, and $\eta \Rightarrow \eta \theta$ is valid w.r.t. $\mathcal{M}$.

A difficulty of automating the method based on Theorem 1 is how to find additional constraints $\eta$. However, the constraints of rules sometimes provide hints to find the constraints $\eta$. For the example $\text{eval}(x, i, z) \rightarrow \text{eval}(x, s(i), \text{plus}(z, s(i))) [x \Rightarrow i \lor i \Rightarrow x] \in R_{\text{sum}}$ in Section 1, we used the constraint $i \Rightarrow x$ as $\eta$. This constraint $i \Rightarrow x$ is a closure of the original constraint $x \Rightarrow i \lor i \Rightarrow x$ of the rule. The following theorem incorporates a heuristics to find the additional constraints $\eta$.

**Theorem 3.** Let $\phi$ be a constraint and $\theta$ be a substitution with $\mathcal{Ran}(\theta) \subseteq \mathcal{T}(\mathcal{G}, \mathcal{V})$, $\eta_1 \lor \cdots \lor \eta_n$ be a disjunctive normal form (DNF, for short) of $\phi$, and $(V, E)$ be a directed graph such that $V = \{\eta_i \mid \eta_i \text{ is satisfiable w.r.t. } \mathcal{M}\}$ and $E = \{\eta_i, \eta_j \mid \eta_i, \eta_j \in V, \eta_i \Rightarrow \eta_j \theta \text{ is valid w.r.t. } \mathcal{M}\}$, and a path $\eta_1', \cdots, \eta_k'$ of nodes ($k > 0$) be a cycle in $(V, E)$. Then, $\eta_1' \lor \cdots \lor \eta_k'$ is satisfiable w.r.t. $\mathcal{M}$, and $(\eta_1' \lor \cdots \lor \eta_k') \Rightarrow \phi$ and $(\eta_1' \lor \cdots \lor \eta_k') \Rightarrow (\eta_1' \lor \cdots \lor \eta_k') \theta$ are valid w.r.t. $\mathcal{M}$.

Since $\eta_1', \cdots, \eta_k'$ are closures of the DNF of $\phi$, the range of valuations satisfying the constraint $\eta_1' \lor \cdots \lor \eta_k'$ obtained from the range of $\phi$, and $\eta_1' \lor \cdots \lor \eta_k'$ plays a role of $\eta$ in Theorem 1. This is the reason why DNFs are used.

**Example 2.** Consider the constraint $x \Rightarrow i \lor i \Rightarrow x$. Let $\phi$ be $x \Rightarrow i \lor i \Rightarrow x$ and $\theta$ be a substitution $\{i \Rightarrow s(i), z \Rightarrow \text{plus}(z, s(i))\}$. Then, neither $x \Rightarrow i \Rightarrow x \Rightarrow s(i), x \Rightarrow i \Rightarrow s(i) \Rightarrow x$, nor $i \Rightarrow x \Rightarrow i \Rightarrow x$ is valid but $i \Rightarrow x \Rightarrow s(i) \Rightarrow x$ is valid. Thus, we have the graph $\{(i \Rightarrow x), (i \Rightarrow x, i \Rightarrow x)\}$, that has a cycle.

Finally, we obtain the following sufficient condition for non-termination.

**Theorem 4.** $R$ is not terminating if there exist a rule $l \rightarrow C[l\theta] [\phi] \in R$ such that $\mathcal{Ran}(\theta|_{\text{Ev}(\theta)}) \subseteq \mathcal{T}(\mathcal{G}, \mathcal{V})$, and for a DNF $\eta_1 \lor \cdots \lor \eta_n$ of $\phi$, the directed graph $\{\{\eta_i \mid \eta_i \text{ is satisfiable w.r.t. } \mathcal{M}\}, \{\eta_i, \eta_j \mid \eta_i, \eta_j \in V, \eta_i \Rightarrow \eta_j \theta \text{ is valid w.r.t. } \mathcal{M}\}\}$ has a cycle.

Note that the sufficient condition in Theorem 4 is decidable if the validity of constraints over $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is decidable. When $n = 1$ in Theorem 4 it suffices to check the validity of $\eta_1 \Rightarrow \eta_1 \theta$.

**Example 3.** The directed graph constructed from the rule $\text{eval}(x, i, z) \rightarrow \text{eval}(x, s(i), \text{plus}(z, s(i))) [x \Rightarrow i \lor i \Rightarrow x] \in R_{\text{sum}}$ in Section 1 has a cycle. Therefore, by Theorem 4, $R_{\text{sum}}$ is not terminating.
The method in this paper to disprove termination analyzes single constrained rules of the form \( l \rightarrow C[l\theta] \) \([w]\). For this reason, the method is not very powerful while the sufficient condition in Theorem 2 is more general. To make the method more practical, we need to improve it. Moreover, we have to experiment for many non-terminating constrained TRSs in order to verify the usefulness of the method.

Acknowledgments We thank Stephan Falke for his useful comments to improve this paper.

References