A Fast Decision Procedure For Uniqueness of Normal Forms w.r.t. Conversion of Shallow Term Rewriting Systems

Masaomi Yamaguchi
Tohoku University, Sendai, Japan
masaomi.yamaguchi.t4@dc.tohoku.ac.jp

Takahito Aoto
Niigata University, Niigata, Japan
aoto@ie.niigata-u.ac.jp

Abstract

Uniqueness of normal forms w.r.t. conversion (UNC) of term rewriting systems (TRSs) guarantees that there are no distinct convertible normal forms. It was recently shown that the UNC property of TRSs is decidable for shallow TRSs (Radcliffe et al., 2010). The existing procedure mainly consists of testing whether there exists a counterexample in a finite set of candidates; however, the procedure suffers a bottleneck of having a sheer number of such candidates. In this paper, we propose a new procedure which consists of checking a smaller number of such candidates and enumerating such candidates more efficiently. Correctness of the proposed procedure is proved and its complexity is analyzed. Furthermore, these two procedures have been implemented and it is experimentally confirmed that the proposed procedure runs much faster than the existing procedure.

2012 ACM Subject Classification Theory of computation → Equational logic and rewriting

Keywords and phrases uniqueness of normal forms w.r.t. conversion, shallow term rewriting systems, decision procedure

Digital Object Identifier 10.4230/LIPIcs.FSCD.2020.11

Supplementary Material Our implementations as well as the details of the experiments can be found in the webpage `http://www.nue.ie.niigata-u.ac.jp/experiments/fscd20/`.

Funding This work is partially supported by JSPS KAKENHI No. 18K11158.

Acknowledgements Thanks are due to the anonymous reviewers for many valuable comments.

1 Introduction

A term rewriting systems (TRS for short) is a well-known model of computation, which plays many roles in equational deduction and formal verification. A key property of the computation of TRSs that it is non-deterministic, which enables flexible computations in TRSs as well as flexible transformations between TRSs and equational axioms. Due to the non-determinism in the computations, however, it is not always guaranteed that results of computations are unique. Thus, properties ensuring unique results of computations are important topics in the study of TRSs. The most well-known such a property is *confluence* (CR), meaning that two convertible terms are joinable. Less known such properties include *uniqueness of normal forms w.r.t. conversion* (UNC) meaning that there are no distinct convertible normal forms. The UNC property[1] of TRSs has been studied in e.g. [3, 4, 11, 12, 13, 19, 10, 20]. Furthermore, interests in automation of proving these properties initiated to start *Confluence Competition* [1, 2, 14] among software tools for proving such properties; there the category of the UNC property has been started from the 2016 edition of the Competition.

[1] The UNC property have been also studied under the name of UN or $\mathrm{UN}^{=}$. We use UNC, following the convention employed in the Confluence Competition.

5th International Conference on Formal Structures for Computation and Deduction (FSCD 2020).
Editor: Zena M. Ariola; Article No. 11; pp. 11:1–11:23
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

One of the important topics concerning these properties is (un)decidability. It is known that the first-order theory of rewriting is decidable for left-linear right-ground TRSs [9]. Indeed, an implementation of the decision procedure of such theory have been reported in [18], and has been applied for (dis)proving these properties of left-linear and right-ground TRSs. An implementation of more efficient decision procedures of these properties for ground TRSs (a subclass of left-linear and right-ground TRSs) have been also reported in [11]. Another line of criteria for such (un)decidability is shallowness and flatness. Shallowness or flatness restricts the depth of (variable) occurrences in the rewrite rules. It is known that confluence is undecidable for flat (and hence shallow) TRSs [15]. In contrast, a polynomial algorithm for deciding the UNC property of linear shallow TRSs have been shown in [20], and it was recently shown that the UNC property of shallow TRSs is decidable [16, 17].

The existing procedure of [17] mainly consists of testing whether there exists a counter-example in a finite set of candidates. However, the procedure suffers a bottleneck of having a sheer number of such candidates even for small examples. In this paper, we propose a new procedure which reduces the number of such candidates to be checked and also enumerates such candidates more efficiently. The proposed procedure has the same structure and is based on the same ideas as the one of [17]; the difference is in the ways of checking the two main cases (whether or not there exists a counterexample to UNC in which the convertible normal forms are convertible to a constant in $\widehat{E_{\mathcal{R}}}$, a complete equations set for TRS $\mathcal{R}$ [8]).

The idea of the proposed method is to construct normal forms which can be reached by minimal constant expansion steps of $\widehat{E_{\mathcal{R}}}$. Based on this idea, we introduce constant propagation algorithm that incrementally constructs normal forms of each constant. Using this algorithm, we can determine whether there exists any minimal counterexample that is equivalent to a constant efficiently. If there exists no such a counterexample, we can check the UNC property efficiently by using the normal forms obtained by the algorithm.

We prove correctness of the proposed decision procedure and analyze its complexity. Furthermore, we implement two UNC decision procedures those based on existing method [17] and those based on proposed method, and experimentally confirm that proposed method runs much faster than existing one.

The rest of the paper is organized as follows: In Section 2, we present basic notions and notations used in this paper, and recall some preliminary backgrounds on our decision procedure. In addition, we overview the existing procedure [17]. In Section 3, we present our new decision procedure, together with its main ingredients – construction of two key sets CP_{NF} and CW – illustrating them through concrete examples. In Section 4, a notion of constant propagation class is introduced; it is used to show the correctness of checking the existence of a minimal witness that is equivalent to a constant by CP_{NF} in Section 5. Section 6 is devoted to show the correctness of checking the existence of a minimal witness that is *not* equivalent to a constant by CW. The correctness theorem and complexity analysis of our decision procedure are given in Section 7. In Section 8, we report our implementation and experiments. In Section 9, we conclude.

2 Preliminaries

In this section, we fix notations that will be used in this paper. Familiarity with term rewriting systems are assumed (see e.g. [6]).

2.1 Term rewriting systems

We denote by $\mathcal{V}$ a countably infinite set of *variables*, and by $\mathcal{F}$ the finite set of (arity-fixed) *function symbols*, which includes the set $\mathcal{C}$ of constants; variables are denoted by $x, y, z, \ldots$, function symbols by $f, g, h, \ldots$, and constants by $a, b, c, \ldots$. The set of terms is denoted by $\mathrm{T}(\mathcal{F},\mathcal{V})$ and the set of non-constant non-variable terms by $\mathrm{T}_f(\mathcal{F},\mathcal{V})$; they may be abbreviated to T and T_f, respectively. We define $\mathrm{height}(t) = 0$ for $t \in \mathcal{C} \cup \mathcal{V}$, and $\mathrm{height}(f(t_1,\ldots,t_n)) = 1 + \max\{\mathrm{height}(t_1),\ldots,\mathrm{height}(t_n)\}$ $(n \geq 1)$. The *size* of a term t, denoted by $|t|$, is 1 if $t \in \mathcal{V}$, and is $1 + \sum_{i=1}^{n} |t_i|$ if $t = f(t_1,\ldots,t_n)$. The set of variables in a term t is denoted by $\mathcal{V}(t)$. The *root symbol* of a term t is denoted by $\mathrm{root}(t)$.

A *substitution* σ is a mapping $\sigma\colon \mathcal{V} \to \mathrm{T}(\mathcal{F},\mathcal{V})$ such that the set $\mathrm{dom}(\sigma) = \{x \mid \sigma(x) \neq x\}$ is finite. When $\mathrm{dom}(\sigma) \subseteq \{x_1,\ldots,x_n\}$, we also write it as $\{x_1 := \sigma(x_1),\ldots,x_n := \sigma(x_n)\}$. A substitution is identified with its homomorphic extension; we write $t\sigma$ for $\sigma(t)$. We write $s \leqslant t$ if $\sigma(s) = t$ for some substitution σ. A renaming substitution is a substitution that is a permutation on variables (i.e. a bijective mapping from $\mathcal{V}$ to $\mathcal{V}$); renaming substitutions are denoted by $\sigma_\alpha, \rho_\alpha, \ldots$. The symbol $\mathbb{N}$ ($\mathbb{N}^*$) stands for the set of (resp. the finite sequences of) natural numbers. We denote the set of *(variable) positions* in a term t by $\mathrm{Pos}(t)$ (resp. $\mathrm{Pos}_{\mathcal{V}}(t)$). The *root position* is denoted by ϵ and the *subterm* at a position p by $t|_p$. A subterm $t|_p$ is a *direct* (*variable*) subterm of t if $p \in \mathbb{N}$ (resp. $t|_p \in \mathcal{V}$). A *hole* is a special constant, denoted by $\Box$. A *context* is a term containing exactly one hole. For a context C and a term t, we denote by $C[t]$ the term obtained by replacing the hole in C by t. A context C is also written as $C[\,]$. Especially, we write $C[\,]_p$ to specify the position of the hole in $C[\,]$. We write $t[\,]_p$ to denote the context obtained by substituting the hole at the position p in a term t.

A *rewrite rule* $l \to r$ satisfies $l \notin \mathcal{V}$; we don't assume, however, the other usual variable restriction $\mathcal{V}(r) \subseteq \mathcal{V}(l)$ in this paper. A *term rewriting system* (*TRS*) is given by $\langle \mathcal{F}, \mathcal{R} \rangle$ where $\mathcal{R}$ is a finite set of rewrite rules over $\mathcal{F}$. When $\langle \mathcal{F}, \mathcal{R} \rangle$ is abbreviated to $\mathcal{R}$, some appropriate $\mathcal{F}$ is fixed. Let $\mathcal{R}$ be a TRS. If there exist $l \to r \in \mathcal{R}$, a substitution σ and a context $C[\,]_p$ such that $s = C[l\sigma]_p$ $t = C[r\sigma]_p$, we have a *rewrite step* $s \to_{\mathcal{R}} t$. The subscript $\mathcal{R}$ may be abbreviated when it is clear from the context. When we need to make (some of) $p, l \to r, \sigma$ explicit, we write $s \to^{p, l \to r, \sigma} t$, etc. A rewrite step $s \to^{p} t$ is *root* if $p = \epsilon$. A non-root rewrite step is denoted by $s \to^{>\epsilon} t$. A term s is a *normal form* if $s \to t$ for no t; the set of normal forms is denoted by *NF*. The symmetric closure of $\to$ is denoted by $\leftrightarrow$, its transitive closure by $\xrightarrow{+}$, its reflexive transitive closure by $\xrightarrow{*}$. its equivalence closure by $\xleftrightarrow{*}$. A successive composition of rewrite steps $s_1 \to \cdots \to s_n$ is called a *rewrite sequence*, which may be abbreviated as $s_1 \xrightarrow{*} s_n$. These notations are reused for other similar relations as well and could be combined. Terms s and t are *convertible* if $s \xleftrightarrow{*} t$. A TRS $\mathcal{R}$ satisfies *uniqueness of normal forms w.r.t. conversion* (UNC) if there are no convertible distinct normal forms, i.e. $s \xleftrightarrow{*} t$ with $s, t \in \mathit{NF}$ implies $s = t$. A finite set of equations is called an equational system (ES for short). We identify equations $l \approx r$ and $r \approx l$. A rewrite step $s \leftrightarrow_E t$ by an equation $l \approx r \in E$ is defined in the same way as for a rewrite rule. For a TRS $\mathcal{R}$, the associated ES $\{l \approx r \mid l \to r \in \mathcal{R}\}$ is denoted by $E_{\mathcal{R}}$.

2.2 UNC of shallow TRSs

A term t is *shallow* if $\mathrm{Pos}_{\mathcal{V}}(t) \subseteq \{\epsilon\} \cup \mathbb{N}$, i.e. $t \in \mathcal{C} \cup \mathcal{V}$ or t contains a variable only as a direct subterm. For example, terms $x, \mathsf{a}, \mathsf{g}(y), \mathsf{f}(x, \mathsf{g}(\mathsf{a}))$ are shallow but $\mathsf{f}(x, \mathsf{g}(y))$ is not. A TRS $\mathcal{R}$ is shallow if l, r are shallow for all $l \to r \in \mathcal{R}$. In [17], a decision procedure for the UNC property of shallow TRSs is given. We now explain some crucial characterizations of UNC, some notions and notations in [17] that will be also used in our decision procedure.

The first step of our decision procedure, as well as that of [17], is to translate shallow TRSs to flat TRSs. A term t is *flat* if $\mathit{height}(t) \leq 1$; a TRS $\mathcal{R}$ is flat if l, r are flat for all $l \to r \in \mathcal{R}$. Clearly, flat TRSs are shallow. On the other hand, a term $\mathsf{f}(x, \mathsf{g}(\mathsf{a}))$ is shallow but not flat. It is known that one can transform shallow TRSs into flat TRSs preserving (non-)UNC; we refer details to [20].

▶ **Example 1.** Let $\mathcal{R}_{shallow} = \{\mathsf{f}(x,y) \to \mathsf{g}(\mathsf{h}(\mathsf{a}))\}$. As $height(\mathsf{g}(\mathsf{h}(\mathsf{a}))) = 2$, the TRS $\mathcal{R}_{shallow}$ is not flat. Now, by the UNC-preserving flatting translation [20], we obtain a flat TRS $\mathcal{R} = \{\mathsf{b} \to \mathsf{h}(\mathsf{a}), \mathsf{f}(x,y) \to \mathsf{g}(\mathsf{b})\}$ from $\mathcal{R}_{shallow}$. Here, b is a newly introduced constant.

Our procedure, as well as the one of [17], employs this transformation. *Henceforth, we focus on flat TRSs.*

The *pattern* of direct subterms in a term $t = f(t_1, \dots, t_n) \in \mathrm{T}_f$, denoted by $Patt(t)$, is the set $\{\{i,j\} \mid 1 \le i,j \le n, i \ne j, t|_i = t|_j\}$; its subset $\{\{i,j\} \in Patt(t) \mid t|_i \in \mathcal{V}\}$ is denoted by $Patt_{\mathcal{V}}(t)$. The following property of root rewrite steps of flat TRSs will be heavily used:

▶ **Lemma 2** ([17]). *Let $\mathcal{R}$ be a flat TRS and t a term. Then, $t \to_{\mathcal{R}}^{\epsilon, l \to r} t'$ for some t' iff (1) $\mathrm{root}(l) = \mathrm{root}(t)$, (2) $l|_i = t|_i$ for all $i \in \mathbb{N}$ with $l_i \in \mathcal{C}$, and (3) $Patt_{\mathcal{V}}(l) \subseteq Patt(t)$.*

An ES is flat if so are all equations in it. An important ingredient of the decision procedure is the *completion* of flat ESs [8]: Given a flat ES E, one can construct a closure $\widehat{E}$ of E with respect to the following rules:

$$\frac{g \approx d, l \approx r}{d\sigma \approx r\sigma} \quad \text{if } l, g \notin \mathcal{V},\ \sigma = \mathrm{mgu}(l,g) \tag{1}$$

$$\frac{l \approx d, y \approx r}{d \approx r\sigma} \quad \text{if } y \in \mathcal{V}, l \in \mathcal{C} \cup \mathcal{V}, \sigma = \{y := l\} \tag{2}$$

$$\frac{C[a] \approx d, a \approx b}{C[b] \approx d} \quad \text{if } a, b \in \mathcal{C} \tag{3}$$

Here, mgu stands for a most general unifier. Then, $\widehat{E}$ is a flat ES that is equivalent to E (i.e. $\xleftrightarrow{*}_E = \xleftrightarrow{*}_{\widehat{E}}$) and is ground complete w.r.t. the ordered rewriting [7]. We won't go into the detail of the latter property, but remark that only we concern in this paper is that, from the latter property, for any given terms s, t, it is decidable whether $s \xleftrightarrow{*}_{\widehat{E}} t$ holds. Our procedure, as well as the one of [17], heavily uses the completion $\widehat{E_{\mathcal{R}}}$ of $E_{\mathcal{R}}$ $(= \{l \approx r \mid l \to r \in \mathcal{R}\})$.

▶ **Example 3.** Let $\mathcal{R} = \{\mathsf{a} \to \mathsf{b}, \mathsf{a} \to \mathsf{f}(x,\mathsf{c},\mathsf{d}), \mathsf{c} \to \mathsf{g}(\mathsf{d}), \mathsf{h}(\mathsf{a}) \to \mathsf{d}, \mathsf{e} \to \mathsf{g}(\mathsf{e})\}$. Then, for example, one obtains $\widehat{E_{\mathcal{R}}} = \{\mathsf{a} \approx \mathsf{b}, \mathsf{a} \approx \mathsf{f}(x,\mathsf{c},\mathsf{d}), \mathsf{b} \approx \mathsf{f}(x,\mathsf{c},\mathsf{d}), \mathsf{f}(x,\mathsf{c},\mathsf{d}) \approx \mathsf{f}(y,\mathsf{c},\mathsf{d}), \mathsf{c} \approx \mathsf{g}(\mathsf{d}), \mathsf{h}(\mathsf{a}) \approx \mathsf{d}, \mathsf{h}(\mathsf{b}) \approx \mathsf{d}, \mathsf{h}(\mathsf{a}) \approx \mathsf{h}(\mathsf{b}), \mathsf{e} \approx \mathsf{g}(\mathsf{e})\}$. It is decidable whether $s \xleftrightarrow{*}_{\widehat{E_{\mathcal{R}}}} t$ (equivalently, $s \xleftrightarrow{*}_{\mathcal{R}} t$) for any given terms s, t.

We have one further point to explain about the use of $\widehat{E_{\mathcal{R}}}$ in the decision procedures. A TRS $\mathcal{R}$ (or an ES E) is *inconsistent* if $x \xleftrightarrow{*}_{\mathcal{R}} y$ (resp. $x \xleftrightarrow{*}_E y$) for some distinct variables x, y; it is *consistent* if it is not inconsistent. Clearly, UNC of a TRS $\mathcal{R}$ implies consistency of $\mathcal{R}$. It is also easy to see that a TRS (or an ES) is inconsistent iff there exists a term t convertible to $x \in \mathcal{V} \setminus \mathcal{V}(t)$. For a flat ES E, this characterization can be strengthened as follows [8]: E is inconsistent iff there exists $x \approx t \in \widehat{E}$ such that $x \notin \mathcal{V}(t)$. Thus, one can check whether $\mathcal{R}$ is consistent using $\widehat{E_{\mathcal{R}}}$.

In the beginning of the decision procedures, one computes a completion $\widehat{E_{\mathcal{R}}}$ of $E_{\mathcal{R}}$. Then, one checks if there exists an equation $x \approx t \in \widehat{E_{\mathcal{R}}}$ such that $x \notin \mathcal{V}(t)$. If this is the case, one knows that $\widehat{E_{\mathcal{R}}}$ is inconsistent, and hence so is $\mathcal{R}$. As inconsistency implies non-UNC, one can conclude $\mathcal{R}$ is not UNC. Thus, the rest of the procedure only deals with the case $\mathcal{R}$ is consistent. For this reason, *we focus on the case that $\mathcal{R}$ is consistent in Sections 4–6.*

The following properties are easily obtained using the definition of $\widehat{E}$ [8]. These properties will be used in subsequent sections without mentioning.

▶ **Proposition 4** ([8]). *Let E be a consistent flat ES, and $\widehat{E}$ its completion. (1) If $s \xleftrightarrow{*}_E t$, then there exists a rewrite sequence $s \xleftrightarrow{*}_{\widehat{E}} t$ that has at most one root rewrite step. (2) For any $c, c' \in \mathcal{C}$ with $c \neq c'$, $c \xleftrightarrow{*}_{\widehat{E}} c'$ iff $c \approx c' \in \widehat{E}$. (3) Suppose $c \xleftrightarrow{*}_{\widehat{E}} c'$. Then, $c \approx r \in \widehat{E}$ iff $c' \approx r \in \widehat{E}$.*

As we focus on the UNC property of $\mathcal{R}$, a pair $\langle s, t\rangle$ of distinct normal forms s, t such that $s \xleftrightarrow{*}_{\mathcal{R}} t$ is called a *witness* (of non-UNC). We call a witness $\langle s, t\rangle$ is *equivalent to a constant* c if $s \xleftrightarrow{*} c$ and $t \xleftrightarrow{*} c$. If $|s| + |t|$ is minimal among all witnesses, $\langle s, t\rangle$ is a *minimal witness*. The set of all subterms of any minimal witness is denoted by $\mathit{SubMinWit}_{\mathcal{R}}$. Clearly, $\mathcal{R}$ has UNC property iff there exists no (minimal) witness. It is not at all easy to see what actually $\mathit{SubMinWit}_{\mathcal{R}}$ is, but it satisfies the following useful claims that will be used later.

▶ **Proposition 5** ([17]). *Let $f(t_1, \ldots, t_n) \in \mathit{SubMinWit}_{\mathcal{R}}$. If $t_i \xleftrightarrow{*} t_j$ then $t_i = t_j$.*

▶ **Lemma 6.** *Let $f(t_1, \ldots, t_n) \in \mathit{SubMinWit}_{\mathcal{R}}$. If $t_i \xleftrightarrow{*} c \in \mathcal{C} \cap \mathit{NF}$ then $t_i = c$.*

2.3 Existing Decision Procedure

Here, we briefly describe the decision procedure of [17]. It is based on these two lemmas:

▶ **Lemma 7** ([17]). *Let $\mathcal{R}$ be a TRS. One can add a finite number of constants to $\mathcal{R}$ to get the TRS $\mathcal{R}'$ which meets the following condition: a witness exists in $\mathcal{R}$ iff a ground witness exists in $\mathcal{R}'$.*

▶ **Lemma 8** ([17]). *Let $\mathcal{R}$ be a flat TRS. If there exists a witness, there exists a witness $\langle s, t\rangle$ such that* $\mathrm{height}(s), \mathrm{height}(t) \leq \max(1, |\mathcal{C}|)$.

Think of a given shallow TRS $\mathcal{R}_{shallow}$. As we explained above, one can get a flat TRS $\mathcal{R}$ from $\mathcal{R}_{shallow}$ preserving (non-)UNC, and its completion $\widehat{E_{\mathcal{R}}}$. From Lemmas 7 and 8, we simply need to check there exists a ground witness $\langle s, t\rangle$ s.t. $\mathrm{height}(s), \mathrm{height}(t) \leq \max(1, |\mathcal{C}|)$, adding a finite number of constants to $\mathcal{R}$. Since there are only finitely many ground terms that satisfy height $\leq \max(1, |\mathcal{C}|)$, one can construct all of them. Thus, it remains to check there exists any pair of such terms that is a witness – this can be decided using $\mathcal{R}$ (whether its components are normal forms) and $\widehat{E_{\mathcal{R}}}$ (whether it consists of convertible terms).

3 New Decision Procedure

In this section, we describe our new decision procedure for the UNC property of shallow TRSs and motivate later sections where we prove its correctness.

3.1 The Whole Procedure

Below, the rewrite step $\leftrightarrow$ of $\widehat{E_{\mathcal{R}}}$ will be abbreviated as $\leftrightarrow$ and NF denotes the set of normal forms w.r.t. $\to_{\mathcal{R}}$.

The whole decision procedure is given in Figure 1. Apart from the same part as the existing procedure (Step 1) and simple checking, the procedure contains two main ingredients – construction of the set $\mathit{CP}_{\mathit{NF}}$ and that of the set CW. We are going to explain the details of these constructions shortly. Actually, these two steps are closely related to the correctness proofs of the previous algorithm [17]. In [17], the authors divide the UNC problem into two main cases according to whether there exists a witness equivalent to a constant. The Step 3 checks whether there exists such a witness and the Step 4 checks whether there exists a witness that is not equivalent to any constant.

Input: a shallow TRS
Output: UNC or Non-UNC

Step 1 Transform the input shallow TRS into a flat TRS $\mathcal{R}$ preserving the UNC property, and calculate its completion $\widehat{E_{\mathcal{R}}}$. If $\mathcal{R}$ is inconsistent (this can be detected when calculating $\widehat{E_{\mathcal{R}}}$), then return Non-UNC.
Step 2 Calculate CP_{NF} by the Constant Propagation Algorithm.
Step 3 If there exists $(\hat{c}, \hat{r}, t, h) \in CP_{NF}$ such that t has a direct variable subterm, then return Non-UNC. If there exist $(\hat{c}, \hat{r}_s, s, h_s), (\hat{c}, \hat{r}_t, t, h_t) \in CP_{NF}$ such that $s \neq t$, then return Non-UNC.
Step 4 Calculate CW. If there exist $\langle s, t\rangle \in CW$ such that $s \neq t$ and $s, t \in NF$, then return Non-UNC.
Step 5 Return UNC.

Figure 1 Proposed decision procedure for UNC of shallow TRSs.

3.2 Constant Propagation Algorithm

Here, we describe how to construct CP_{NF} by Constant Propagation Algorithm, which determine whether there exists any minimal counterexample that is equivalent to a constant.

We first need a couple of notion and notation.

▶ **Definition 9** (equivalence relation $\simeq$ on flat terms). *We define an equivalence relation $\simeq$ on flat terms like this: $s \simeq t$ iff either (1) $s, t \in \mathcal{V}$ and $s = t$, (2) $s, t \in \mathcal{C}$ and $s \overset{*}{\leftrightarrow} t$, or (3) $s = f(s_1, \ldots, s_n)$, $t = f(t_1, \ldots, t_n)$ $(n \geq 1)$ such that (a) $s_i \in \mathcal{C}$ iff $t_i \in \mathcal{C}$ for all $1 \leq i \leq n$, (b) $s_i \overset{*}{\leftrightarrow} t_i$ for all $s_i \in \mathcal{C}$, and (c) there exists a renaming substitution σ_α such that $\sigma_\alpha(s_i) = t_i$ for all $s_i \in \mathcal{V}$.*

It is easy to see that $\simeq$ is indeed an equivalence relation. The $\simeq$-equivalence class of a flat term t is denoted by $[\![t]\!]$.

▶ **Definition 10.** *We fix a representative element of $[\![c]\!]$ $(c \in \mathcal{C})$ and denote it by $\hat{c}$. We denote the set $\{\hat{c} \mid c \in \mathcal{C}\}$ by $\widehat{\mathcal{C}}$. For an arbitrary flat term t, we define $\hat{x} = x$ and $\hat{t} = f(\hat{t}_1, \ldots, \hat{t}_n)$ for $t = f(t_1, \ldots, t_n)$, in addition.*

The idea of our algorithms comes from the observation that, for any term t which is equivalent to a constant c, we have $t \overset{*}{\to} c$ by the ordered rewriting of $\widehat{E_{\mathcal{R}}}$, and c (or maybe another constant that is equivalent to c) is a $\widehat{E_{\mathcal{R}}}$-normal form of t. Our algorithm incrementally searches normal forms of each constant, tracing the inverse direction of the $\widehat{E_{\mathcal{R}}}$-rewriting sequences.

▶ **Definition 11** (Constant Propagation (CP) Algorithm). *Suppose a flat TRS $\mathcal{R}$ and its completion $\widehat{E_{\mathcal{R}}}$ are given. The algorithm incrementally computes a set of quadruples CP_{NF} by the pseudo-code presented in Figure 2.*

Actually, the fourth element of quadruples is unused to compute the result; the sole purpose of adding auxiliary parameter H is to use it in our proof below.

▶ **Example 12.** Let $\mathcal{R}$ and $\widehat{E_{\mathcal{R}}}$ be given as in Example 3. The constant propagation algorithm runs as follows:

Input: a flat TRS $\mathcal{R}$ and its completion $\widehat{E_{\mathcal{R}}}$
Output: CP_{NF}

Step 1: $H := 0$; $\widehat{\mathcal{C}} := \{\hat{c}_1, \ldots, \hat{c}_m\}$
For each $i = 1, \ldots, m$: $Y_{\hat{c}_i} := \{\hat{r} \mid r \in \mathrm{T}_f,\ \hat{c}_i \approx r \in \widehat{E_{\mathcal{R}}}\}$
Step 2: $CP_{NF} := \{(\hat{c}, \hat{c}, c, 0) \mid c \in \mathcal{C} \cap NF\}$
Step 3: Repeat the following **(main loop)**:
$H := H + 1$; $X_{tmp} := \emptyset$
Calculate a function $\chi_H : \widehat{\mathcal{C}} \cup \mathcal{V} \to \mathcal{P}(\mathrm{T})$ as follows :

$$\chi_H(\hat{c}) = \begin{cases} \{u \mid (\hat{c}, \_, u, \_) \in CP_{NF}\} & \text{if } \exists (\hat{c}, \_, u, \_) \in CP_{NF} \\ \{\hat{c}\} & \text{otherwise} \end{cases}$$

$$\chi_H(x) = \{x\}$$

For each $i = 1, \ldots, m$, calculate as follows :
For each $\hat{r} = f(\hat{u}_1, \ldots, \hat{u}_n) \in Y_{\hat{c}_i}$, calculate as follows :

$$X_{i,\hat{r}} := \left\{ (\hat{c}_i, \hat{r}, f(u'_1, \ldots, u'_n), H) \;\middle|\; \begin{array}{l} u'_j \in \chi_H(\hat{u}_j)\ (1 \le j \le n) \\ f(u'_1, \ldots, u'_n) \in NF \end{array} \right\}$$

$$Y_{\hat{c}_i} := \begin{cases} Y_{\hat{c}_i} \backslash \{\hat{r}\} & \text{if } X_{i,\hat{r}} \neq \emptyset \\ Y_{\hat{c}_i} & \text{otherwise} \end{cases}$$

$$X_{tmp} := X_{i,\hat{r}} \cup X_{tmp}$$

$CP_{NF} := X_{tmp} \cup CP_{NF}$
If $X_{tmp} = \emptyset$, exit the main loop

Figure 2 Constant Propagation Algorithm.

1. Choose a representative element among convertible constants. As we have $\mathsf{a} \overset{*}{\leftrightarrow} \mathsf{b}$, let us pick up a as their representative, i.e. $\hat{\mathsf{a}} = \hat{\mathsf{b}} = \mathsf{a}$ (picking b leads no problem). Since there are no other distinct convertible constants, we have $\widehat{\mathcal{C}} = \{\mathsf{a}, \mathsf{c}, \mathsf{d}, \mathsf{e}\}$. Thus, we set $Y_{\mathsf{a}} = \{\mathsf{f}(x, \mathsf{c}, \mathsf{d})\}$, $Y_{\mathsf{c}} = \{\mathsf{g}(\mathsf{d})\}$, $Y_{\mathsf{d}} = \{\mathsf{h}(\mathsf{a})\}$ and $Y_{\mathsf{e}} = \{\mathsf{g}(\mathsf{e})\}$ in Step 1.
2. In Step 2, as $\mathcal{C} \cap NF = \{\mathsf{b}, \mathsf{d}\}$, we initialize $CP_{NF} := \{(\mathsf{a}, \mathsf{a}, \mathsf{b}, 0), (\mathsf{d}, \mathsf{d}, \mathsf{d}, 0)\}$. Intuitively, this expresses that a term b (d) is one of the convertible normal forms of the constant a (resp. d).
3. Now we run into the first loop of the Step 3. We have $\chi_1(\mathsf{a}) = \{\mathsf{b}\}$ and $\chi_1(x) = \{x\}$ for $x \neq \mathsf{a}$. Now, we check whether this replacement mapping χ_1 can make elements of $Y_{\mathsf{a}} \cup Y_{\mathsf{c}} \cup Y_{\mathsf{d}} \cup Y_{\mathsf{e}}$ a normal form. Then, we find that $\mathsf{g}(\mathsf{d}) \in NF$ is obtained from $\mathsf{g}(\mathsf{d}) \in Y_{\mathsf{c}}$ and $\mathsf{h}(\mathsf{b}) \in NF$ is obtained from $\mathsf{h}(\mathsf{a}) \in Y_{\mathsf{d}}$. Thus, we updates the sets as: $CP_{NF} := CP_{NF} \cup \{(\mathsf{c}, \mathsf{g}(\mathsf{d}), \mathsf{g}(\mathsf{d}), 1), (\mathsf{d}, \mathsf{h}(\mathsf{a}), \mathsf{h}(\mathsf{b}), 1)\}$, $Y_{\mathsf{c}} := \emptyset$ and $Y_{\mathsf{d}} := \emptyset$. Intuitively, in this step, we found a normal form $\mathsf{g}(\mathsf{d})$ ($\mathsf{h}(\mathsf{b})$) equivalent to a constant c (resp. d).
4. Now we run into the second loop of the Step 3. We have $\chi_2(\mathsf{a}) = \{\mathsf{b}\}$, $\chi_2(\mathsf{c}) = \{\mathsf{g}(\mathsf{d})\}$, $\chi_2(\mathsf{d}) = \{\mathsf{d}, \mathsf{h}(\mathsf{b})\}$ and $\chi_2(x) = \{x\}$ for $x \notin \{\mathsf{a}, \mathsf{c}, \mathsf{d}\}$. Again, we check whether this replace mapping can make remaining elements of $Y_{\mathsf{a}} \cup Y_{\mathsf{e}}$ a normal form. Then, we find that $\mathsf{f}(x, \mathsf{g}(\mathsf{d}), \mathsf{d}), \mathsf{f}(x, \mathsf{g}(\mathsf{d}), \mathsf{h}(\mathsf{b})) \in NF$ are obtained from $\mathsf{f}(x, \mathsf{c}, \mathsf{d}) \in Y_{\mathsf{a}}$. Thus, we update the sets as: $CP_{NF} := CP_{NF} \cup \{(\mathsf{a}, \mathsf{f}(x, \mathsf{c}, \mathsf{d}), \mathsf{f}(x, \mathsf{g}(\mathsf{d}), \mathsf{d}), 2), (\mathsf{a}, \mathsf{f}(x, \mathsf{c}, \mathsf{d}), \mathsf{f}(x, \mathsf{g}(\mathsf{d}), \mathsf{h}(\mathsf{b})), 2)\}$ and $Y_{\mathsf{a}} := \emptyset$.
5. The third round of the loop of the Step 3 finds no new normal forms, and $X_{tmp} = \emptyset$.

Thus, we exit the loop.
Finally, we have $CP_{NF} =$

$$\left\{ \begin{array}{lll} (\mathsf{a}, \mathsf{a}, \mathsf{b}, 0), & (\mathsf{a}, \mathsf{f}(x, \mathsf{c}, \mathsf{d}), \mathsf{f}(x, \mathsf{g}(\mathsf{d}), \mathsf{d}), 2), & (\mathsf{a}, \mathsf{f}(x, \mathsf{c}, \mathsf{d}), \mathsf{f}(x, \mathsf{g}(\mathsf{d}), \mathsf{h}(\mathsf{b})), 2), \\ (\mathsf{c}, \mathsf{g}(\mathsf{d}), \mathsf{g}(\mathsf{d}), 1), & (\mathsf{d}, \mathsf{d}, \mathsf{d}, 0), & (\mathsf{d}, \mathsf{h}(\mathsf{a}), \mathsf{h}(\mathsf{b}), 1) \end{array} \right\}.$$

Observe that, for any quadruples $(c, r, u, h) \in CP_{NF}$, we have $c \overset{*}{\leftrightarrow} u$ and $u \in NF$. Thus, from the final CP_{NF}, one easily see some witnesses equivalent to a constant: e.g. $\langle \mathsf{d}, \mathsf{h}(\mathsf{b}) \rangle$ (equivalent to d), $\langle \mathsf{b}, \mathsf{f}(x, \mathsf{g}(\mathsf{d}), \mathsf{d}) \rangle$ and $\langle \mathsf{f}(x, \mathsf{g}(\mathsf{d}), \mathsf{d}), \mathsf{f}(x, \mathsf{g}(\mathsf{d}), \mathsf{h}(\mathsf{b})) \rangle$ (equivalent to a). One also obtains a witness $\langle \mathsf{f}(x, \mathsf{g}(\mathsf{d}), \mathsf{d}), \mathsf{f}(y, \mathsf{g}(\mathsf{d}), \mathsf{d})$ (equivalent to a), since renaming x to y leads $\mathsf{f}(x, \mathsf{g}(\mathsf{d}), \mathsf{d}) \overset{*}{\leftrightarrow} \mathsf{a} \overset{*}{\leftrightarrow} \mathsf{f}(y, \mathsf{g}(\mathsf{d}), \mathsf{d})$.

To characterize the situation where we couldn't find any witness from CP_{NF}, we introduce the following property.

▶ **Definition 13** (consistency of CP_{NF})**.** *CP_{NF} is* consistent *if (i) there exists no $(\hat{c}, \hat{r}, t, h) \in CP_{NF}$ such that t has a direct variable subterm, and (ii) no $(\hat{c}, \hat{r}_s, s, h_s), (\hat{c}, \hat{r}_t, t, h_t) \in CP_{NF}$ such that $s \neq t$. It is* inconsistent *if it is not consistent.*

Before ending this subsection, we introduce one notation that will be used below.

▶ **Definition 14.** *We define $\mathrm{T}^{CP} = \{t \mid (\hat{c}, \hat{r}, t, h) \in CP_{NF}\}$.*

3.3 Construction of CW

CP_{NF} finds only constant-equivalent witnesses, so we need to know yet whether there exist any minimal witnesses that are *not* equivalent to a constant.

From Step 3, we may suppose that CP_{NF} is consistent. Assume a term t is equivalent to a constant c. We denote a term $\tilde{t}$ such that $\tilde{t} \in \mathrm{T}^{CP}$ and $c \overset{*}{\leftrightarrow} \tilde{t}$. If such a term exists, it must be unique from the assumption.

▶ **Definition 15.** *(1) Define $\delta''(t)$ for $t \in \mathcal{C} \cup \mathcal{V}$ as follows: $\delta''(t) = \tilde{t}$ if $t \in \mathcal{C}$, and $\delta''(t) = t$ otherwise. (2) Define $\psi''(t)$ for non-constant flat terms t as follows: $\psi''(t) = t$ if $t \in \mathcal{V}$, and $\psi''(t) = f(\delta''(t_1), \ldots, \delta''(t_n))$ if $t = f(t_1, \ldots, t_n)$ $(n \geq 1)$. (3) Finally, define the set CW:*

$$CW = \{\langle \psi''(l), \psi''(r) \rangle \mid l \approx r \in \widehat{E_{\mathcal{R}}},\ l, r \notin \mathcal{C}\}.$$

▶ **Example 16.** Let us consider $\mathcal{R}$ of Example 1. Through the Step 1 of the Figure 1, we obtain a completion of $\mathcal{R}$ as $\widehat{E_{\mathcal{R}}} = \{\mathsf{b} \approx \mathsf{h}(\mathsf{a}), \mathsf{f}(x, y) \approx \mathsf{g}(\mathsf{b}), \mathsf{f}(x, y) \approx \mathsf{f}(x_1, y_1)\}$. By the Step 2, we obtain $CP_{NF} := \{(\mathsf{a}, \mathsf{a}, \mathsf{a}, 0), (\mathsf{b}, \mathsf{h}(\mathsf{a}), \mathsf{h}(\mathsf{a}), 1)\}$. The conditions of Step 3 fails, and thus we run into Step 4. Note here that from CP_{NF}, we have $\tilde{\mathsf{a}} = \mathsf{a}$ and $\tilde{\mathsf{b}} = \mathsf{h}(\mathsf{a})$. Thus, we obtain $CW = \{\langle \mathsf{f}(x, y), \mathsf{g}(\mathsf{h}(\mathsf{a})) \rangle, \langle \mathsf{f}(x, y), \mathsf{f}(x_1, y_1) \rangle\}$. Since none of $\mathsf{f}(x, y), \mathsf{g}(\mathsf{h}(\mathsf{a})), \mathsf{f}(x, y), \mathsf{f}(x_1, y_1)$ is a normal form of $\mathcal{R}$, we conclude that $\mathcal{R}$ (and hence $\mathcal{R}_{shallow}$) is UNC.

In the subsequent sections, we prove the correctness of our decision procedure, and report its complexity analysis and the result of experiments.

4 Constant Propagation Class

In the previous section, we introduced CP_{NF} that finds witnesses that are equivalent to constants. In this section, we introduce a notion of constant propagation class, a key notion that acts as a mediator between CP_{NF} and constant-equivalent witnesses.

In subsequent sections, we fix a TRS $\mathcal{R}$ that is consistent and flat, and its completion $\widehat{E_\mathcal{R}}$. For convenience, we also use $\mathcal{R}$ and $\widehat{E_\mathcal{R}}$ as if they are closed under renaming, i.e. we assume $\mathcal{R}$ ($\widehat{E_\mathcal{R}}$) includes all the rules (resp. equations) whose variables are renamed, and use $\widehat{E_\mathcal{R}}$ as if it contains trivial equations $c \approx c$ ($c \in \mathcal{C}$).

▶ **Definition 17** (constant expansion). *A term t is obtained from s by a* constant expansion*, written as $s \rightharpoonup_{\widehat{E_\mathcal{R}}} t$, if there exists $c \in \mathcal{C}$ with $c \approx r \in \widehat{E_\mathcal{R}}$ and a position p such that $s|_p = c$, $t = s[r]_p$ and $\mathcal{V}(r) \cap \mathcal{V}(s) = \emptyset$.*

Henceforth, we will omit the subscript $\widehat{E_\mathcal{R}}$ of $s \rightharpoonup_{\widehat{E_\mathcal{R}}} t$. Clearly, $t \rightharpoonup t'$ implies $t \leftrightarrow t'$.

▶ **Example 18.** Let $\mathcal{R}$ and $\widehat{E_\mathcal{R}}$ be given as in Example 3 (enhanced by trivial equations and renamed rules, as explained). Then we have $\mathsf{a} \rightharpoonup \mathsf{a}$, $\mathsf{a} \rightharpoonup \mathsf{b}$, $\mathsf{a} \rightharpoonup \mathsf{f}(x, \mathsf{c}, \mathsf{d}) \rightharpoonup \mathsf{f}(x, \mathsf{c}, \mathsf{h}(\mathsf{a})) \rightharpoonup \mathsf{f}(x, \mathsf{c}, \mathsf{h}(\mathsf{f}(y, \mathsf{c}, \mathsf{d})))$, and $\mathsf{a} \rightharpoonup \mathsf{f}(x, \mathsf{c}, \mathsf{d}) \rightharpoonup \mathsf{f}(x, \mathsf{c}, \mathsf{h}(\mathsf{b})) \rightharpoonup \mathsf{f}(x, \mathsf{c}, \mathsf{h}(\mathsf{a})) \rightharpoonup \mathsf{f}(x, \mathsf{g}(\mathsf{d}), \mathsf{h}(\mathsf{a}))$.

For any $r \in \mathrm{T}_f$, we have $r \overset{*}{\rightharpoonup} t$ iff $r \overset{*}{\rightharpoonup}{}^{>\epsilon} t$, and hence $r \overset{*}{\rightharpoonup} t$ implies $\mathrm{root}(r) = \mathrm{root}(t)$. This motivates us to introduce a term class parameterized by $c \in \mathcal{C}$ and $r \in \mathrm{T}_f$ as follows.

▶ **Definition 19** (constant propagation class). *Let $c \in \mathcal{C}, r \in T$. Define the* constant propagation class (CPC) *of the pair $\langle c, r\rangle$ as follows:*

$$\mathrm{CP}(c, r) = \{t \mid r \in \mathrm{T}_f,\ c \approx r \in \widehat{E_\mathcal{R}},\ \exists t' \ s.t.\ c \rightharpoonup r \overset{*}{\rightharpoonup} t' \leqslant t\}$$

Remark that $t \in \mathrm{CP}(c, r)$ implies that $t \in \mathrm{T}_f$ and $\mathrm{root}(t) = \mathrm{root}(r)$.

▶ **Example 20.** Let $\widehat{E_\mathcal{R}}$ be as in Example 18. Then, $\mathsf{f}(\mathsf{b}, \mathsf{c}, \mathsf{h}(\mathsf{f}(x, \mathsf{c}, \mathsf{d}))) \in \mathrm{CP}(\mathsf{a}, \mathsf{f}(x, \mathsf{c}, \mathsf{d}))$ as $\mathsf{a} \rightharpoonup \mathsf{f}(x, \mathsf{c}, \mathsf{d}) \overset{*}{\rightharpoonup} \mathsf{f}(x, \mathsf{c}, \mathsf{h}(\mathsf{f}(y, \mathsf{c}, \mathsf{d}))) \leqslant \mathsf{f}(\mathsf{b}, \mathsf{c}, \mathsf{h}(\mathsf{f}(x, \mathsf{c}, \mathsf{d})))$.

Next lemma shows that the class $\mathrm{CP}(c, r)$ is invariant under renaming.

▶ **Lemma 21.** *Let $c \in \mathcal{C}$, $r \in \mathrm{T}_f$, and assume $c \approx r \in \widehat{E_\mathcal{R}}$. Then, $\mathrm{CP}(c, r) = \{t \mid \exists t', \sigma_\alpha \ s.t.\ c \rightharpoonup \sigma_\alpha(r) \overset{*}{\rightharpoonup} t' \leqslant t\}$, and hence $\mathrm{CP}(c, r) = \mathrm{CP}(c, \sigma_\alpha(r))$.*

The invariance of the $\mathrm{CP}(c, r)$ can be extended further than renaming. Firstly, $\mathrm{CP}(c, r)$ and $[\![c]\!]$ share the following property.

▶ **Lemma 22.** *Let $t \in \mathrm{CP}(c, r) \cup [\![c]\!]$. Then, (1) $c \overset{*}{\rightharpoonup} t' \leqslant t$ for some t', and (2) $c \overset{*}{\leftrightarrow} t$.*

We are now going to show that any CPC is preserved under $\simeq$ (Theorem 25).

▶ **Lemma 23.** *Let $c, c' \in \mathcal{C}$ and $r, r' \in \mathrm{T}_f$. (1) $c \simeq c'$ iff $c \approx c' \in \widehat{E_\mathcal{R}}$. (2) If $r \simeq r'$ then $c \approx r \in \widehat{E_\mathcal{R}}$ iff $c \approx r' \in \widehat{E_\mathcal{R}}$.*

▶ **Lemma 24.** *Let $c, c' \in \mathcal{C}$ and $r, r' \in \mathrm{T}_f$. (1) If $c \simeq c'$ then $c \rightharpoonup c'$. (2) If $c \simeq c'$, then $c \rightharpoonup r$ iff $c' \rightharpoonup r$. (3) If $c \simeq c'$ and $r \simeq r'$, then $c \rightharpoonup r$ iff $c' \rightharpoonup r'$. (4) If $r \simeq r'$ then there exists σ_α such that $\sigma_\alpha(r') \overset{*}{\rightharpoonup} r$.*

▶ **Theorem 25** (preservation of CPCs by $\simeq$). *CPCs are preserved by $\simeq$, i.e. if $c \simeq c'$ and $r \simeq r'$ then $\mathrm{CP}(c, r) = \mathrm{CP}(c', r')$. In particular, $\mathrm{CP}(c, r) = \mathrm{CP}(\hat{c}, \hat{r})$.*

Proof. Use above four lemmas and Proposition 4. See appendix for the detail. ◀

Before ending the section, we relate $\mathit{SubMinWit}_\mathcal{R}$ and CPCs.

▶ **Theorem 26.** *Let $c \in \mathcal{C}$ and $t \in \mathit{SubMinWit}_\mathcal{R}$ such that $c \overset{*}{\leftrightarrow} t$. Then, there exists a term r such that $t \in \mathrm{CP}(c, r) \cup [\![c]\!]$. Hence, $t \in \mathrm{CP}(\hat{c}, \hat{r}) \cup [\![\hat{c}]\!]$.*

Proof. The proof proceeds by induction on $h = height(t)$. Let $c \in \mathcal{C}$ and $t \in SubMinWit_{\mathcal{R}}$, and assume $c \overset{*}{\leftrightarrow} t$.

1. (B.S.) Suppose $h = 0$. Then $t \in \mathcal{C} \cup \mathcal{V}$ holds. It follows from the consistency of $\mathcal{R}$ that $t \in \mathcal{C}$. Since $c \overset{*}{\leftrightarrow} t$, we have $c \simeq t$. Thus $t \in [\![c]\!]$.
2. (I.S.) By $h > 0$, we have $t \notin [\![c]\!]$. From $c \overset{*}{\leftrightarrow} t$ and Proposition 4, it follows that we have

$$c \leftrightarrow^{\epsilon} f(s'_1, \ldots, s'_n) \overset{*}{\leftrightarrow}{}^{>\epsilon} f(t_1, \ldots, t_n) = t.$$

   Let $c \leftrightarrow f(s_1, \ldots, s_n) \in \widehat{E_{\mathcal{R}}}$ be the rule used at the rewrite step $c \leftrightarrow^{\epsilon} f(s'_1, \ldots, s'_n)$. Then, there exists a substitution σ such that $f(\sigma(s_1), \ldots, \sigma(s_n)) = f(s'_1, \ldots, s'_n)$ and $\sigma(s_i) \overset{*}{\leftrightarrow} t_i$ $(1 \leq i \leq n)$. Now, define u_i and σ_i such that $s_i \overset{*}{\rightharpoonup} u_i$, $\sigma_i(u_i) = t_i$ $(i = 1, \ldots, n)$ so that $\bigcup_i \sigma_i$ is well-defined, according to the following case distinction:
   - **a.** Case $s_i = c_i \in \mathcal{C}$. Then, $\sigma(s_i) = c_i \overset{*}{\leftrightarrow} t_i$ holds. From consistency of $\mathcal{R}$, we know $t_i \notin \mathcal{V}$. By $c_i \overset{*}{\leftrightarrow} t_i$ and $t_i \in SubMinWit_{\mathcal{R}}$, it follows from induction hypothesis that there exists r_i such that $t_i \in \mathrm{CP}(c_i, r_i) \cup [\![c_i]\!]$.
     - **i.** Case $t_i \in [\![c_i]\!]$. Put $u_i = t_i$ and $\sigma_i = \{\}$ (the identity substitution). Then $s_i = c_i \simeq t_i = u_i$, and hence we have $s_i \rightharpoonup u_i$ by Lemma 24. Clearly, $\sigma_i(u_i) = t_i$ and $\mathcal{V}(u_i) = \emptyset$.
     - **ii.** Case $t_i \in \mathrm{CP}(c_i, r_i)$. Then, $r_i \in \mathrm{T}_f$ holds. From Lemma 21, $c_i \rightharpoonup \sigma_{\alpha_i}(r_i) \overset{*}{\rightharpoonup} t'_i \leqslant t_i$ for some t'_i and σ_{α_i} such that the variables in $\sigma_{\alpha_i}(r_i)$ or t'_i are fresh. Put $u_i = t'_i$ and take σ_i as a substitution such that $\sigma_i(t'_i) = t_i$. Clearly, we have $s_i = c_i \overset{*}{\rightharpoonup} u_i$ and $\sigma_i(u_i) = t_i$. Furthermore, variables in $\mathrm{dom}(\sigma_i)$ and $\mathcal{V}(u_i)$ are fresh.
   - **b.** Case $r_i \in \mathcal{V}$. Put $u_i = r_i$ and $\sigma_i = \{u_i := t_i\}$. Clearly, we have $r_i \overset{*}{\rightharpoonup} u_i$ and $\sigma_i(u_i) = t_i$.

   Now, we show the substitution $\sigma = \bigcup_{1 \leq i \leq n} \sigma_i$ is well-defined. From the construction, it is clear that it suffices to show $t_p = t_q$ whenever $s_p = s_q \in \mathcal{V}$ $(1 \leq p, q \leq n)$. If $s_p = s_q$, then $t_p \overset{*}{\leftrightarrow} \sigma(s_p) = \sigma(s_q) \overset{*}{\leftrightarrow} t_q$ holds. Then, since $t \in SubMinWit_{\mathcal{R}}$, we have $t_p = t_q$ by Proposition 5. Hence, σ is well-defined. Now we have $c \rightharpoonup f(s_1, \ldots, s_n) \overset{*}{\rightharpoonup} f(u_1, \ldots, u_n)$ and $\sigma(f(u_1, \ldots, u_n)) = t$. Thus, $t \in \mathrm{CP}(c, f(s_1, \ldots, s_n))$.

Thus, there exists r such that $t \in \mathrm{CP}(c, r) \cup [\![c]\!]$. Also, $t \in \mathrm{CP}(\hat{c}, \hat{r}) \cup [\![\hat{c}]\!]$ by Theorem 25. ◀

5 Correctness of Constant Propagation Algorithm

In this section, we describe the correctness of Constant Propagation Algorithm given in Figure 2, which checks whether there exists a minimal witness that is equivalent to a constant.

Because of the main loop, termination of the algorithm needs to be clarified.

▶ **Lemma 27.** *CP algorithm terminates.*

The following properties of elements in CP_{NF} are immediate from the definition.

▶ **Lemma 28.** *Let $t \in \mathcal{C} \cap NF$. Then, $t \in [\![\hat{c}]\!]$ iff $(\hat{c}, \hat{c}, t, 0) \in CP_{NF}$.*

▶ **Lemma 29.** *Let $(\hat{c}, \hat{r}, t, h) \in CP_{NF}$. Then, (1) $\mathrm{height}(t) = h$, (2) $\mathrm{root}(\hat{r}) = \mathrm{root}(t)$, (3) $\hat{r}|_i \in \mathcal{V} \Rightarrow \hat{r}|_i = t|_i$ for each $i \in \mathbb{N}$, (4) $t \in NF$, and (5) $t \notin \mathcal{V}$.*

Further properties are established as well.

▶ **Lemma 30.** *Let $(\hat{c}, \hat{r}, t, h) \in CP_{NF}$. Then, (1) $\hat{c} \rightharpoonup \hat{r} \overset{*}{\rightharpoonup} t$, (2) if $h > 0$ then $t \in \mathrm{CP}(\hat{c}, \hat{r})$, and (3) $\hat{c} \overset{*}{\leftrightarrow} t$.*

From the previous lemma and Lemma 28, $(\hat{c}, \hat{r}, t, h) \in CP_{NF}$ implies $t \in \mathrm{CP}(\hat{c}, \hat{r}) \cup [\![\hat{c}]\!]$. We now consider the reverse direction. In fact, we have already shown the case $t \in [\![\hat{c}]\!]$ in Lemma 28. For the case $t \in \mathrm{CP}(\hat{c}, \hat{r})$, we need a further assumption that $t \in \mathit{SubMinWit}_{\mathcal{R}}$.

▶ **Lemma 31.** *Let $t \in \mathit{SubMinWit}_{\mathcal{R}} \cap \mathrm{CP}(\hat{c}, \hat{r})$. Then, $(\hat{c}, \hat{r}, t', h') \in CP_{NF}$ for some t' and $h' \leq \mathrm{height}(t)$.*

A final property of elements in CP_{NF} we need is a kind of injectivity.

▶ **Lemma 32.** *Suppose $(\hat{c}_s, \hat{r}_s, s, h_s), (\hat{c}_t, \hat{r}_t, t, h_t) \in CP_{NF}$. Then $s = t$ implies $\hat{c}_s = \hat{c}_t$ and $\hat{r}_s = \hat{r}_t$.*

Now we arrive at the main result of this section – CP_{NF} gives a necessary and sufficient criteria to find a minimal witness equivalent to a constant.

▶ **Theorem 33.** *There exists a witness that is equivalent to a constant iff CP_{NF} is inconsistent.*

Proof. ($\Rightarrow$) Let $\langle u, v\rangle$ be a minimal witness that is equivalent to a constant c. We have $c \overset{*}{\leftrightarrow} u$, $c \overset{*}{\leftrightarrow} v$, $u \neq v$ and $u, v \in \mathit{NF}$. From Theorem 26, there exists r_u, r_v such that $u \in \mathrm{CP}(\hat{c}, \hat{r}_u) \cup [\![\hat{c}]\!]$ and $v \in \mathrm{CP}(\hat{c}, \hat{r}_v) \cup [\![\hat{c}]\!]$. We distinguish four cases:

1. Case $u \in [\![\hat{c}]\!]$ and $v \in [\![\hat{c}]\!]$. From Lemma 28, we have $(\hat{c}, \hat{c}, u, 0), (\hat{c}, \hat{c}, v, 0) \in CP_{NF}$. Since $u \neq v$, the claim holds.
2. Case $u \in \mathrm{CP}(\hat{c}, \hat{r}_u)$ and $v \in [\![\hat{c}]\!]$. From Lemma 31, there exists $u' \in \mathrm{CP}(\hat{c}, \hat{r}_u)$ such that $(\hat{c}, \hat{r}_u, u', h'_u) \in CP_{NF}$. From Lemma 28, we have $(\hat{c}, \hat{c}, v, 0) \in CP_{NF}$. Since $\hat{r}_u \notin \mathcal{C}$, $\hat{c} \neq \hat{r}_u$ holds. Hence Lemma 32 leads $u' \neq v$.
3. Case $u \in [\![\hat{c}]\!]$ and $v \in \mathrm{CP}(\hat{c}, \hat{r}_v)$. Similar to the previous case.
4. Case $u \in \mathrm{CP}(\hat{c}, \hat{r}_u)$ and $v \in \mathrm{CP}(\hat{c}, \hat{r}_v)$. If $(\hat{c}, \hat{r}_u) \neq (\hat{c}, \hat{r}_v)$, then $u \neq v$ by Lemma 32. So, suppose otherwise, i.e. $(\hat{c}, \hat{r}_u) = (\hat{c}, \hat{r}_v)$. By $\hat{r}_u = \hat{r}_v$, $\mathrm{root}(u) = \mathrm{root}(\hat{r}_u) = \mathrm{root}(\hat{r}_v) = \mathrm{root}(v)$. Thus, one can let $u = f(t_{u,1}, \ldots, t_{u,n})$, $v = f(t_{v,1}, \ldots, t_{v,n})$ and $\hat{r}_u = \hat{r}_v = f(s_1, \ldots, s_n)$. Then, there exist $u', v' \in \mathrm{T}_f$, $\sigma_u, \sigma_v \in \Sigma$ such that

$$c \rightharpoonup f(s_1, \ldots, s_n) \overset{*}{\rightharpoonup} f(u_1, \ldots, u_n) = u', \quad \sigma_u(u') = u,$$
$$c \rightharpoonup f(s_1, \ldots, s_n) \overset{*}{\rightharpoonup} f(v_1, \ldots, v_n) = v', \quad \sigma_v(v') = v$$

   where $s_i \overset{*}{\rightharpoonup} u_i$, $s_i \overset{*}{\rightharpoonup} v_i$ for all $1 \leq i \leq n$. Since $u \neq v$, $\sigma_u(u_i) \neq \sigma_v(v_i)$ for some $1 \leq i \leq n$. Assume $\sigma_u(u_i) \neq \sigma_v(v_i)$. Suppose $s_i \in \mathcal{C}$. Then, we have $\sigma_u(u_i) \overset{*}{\leftrightarrow} s_i \overset{*}{\leftrightarrow} \sigma_v(v_i)$. Since $\sigma_u(u_i), \sigma_v(v_i) \in \mathit{NF}$, $\langle \sigma_u(u_i), \sigma_v(v_i)\rangle$ is a witness equivalent to the constant s_i. This violates the minimality of $\langle u, v\rangle$. Hence $s_i \in \mathcal{V}$. Now, as $u \in \mathit{SubMinWit}_{\mathcal{R}} \cap \mathrm{CP}(\hat{c}, \hat{r}_u)$, there exists $u' \in \mathrm{CP}(\hat{c}, \hat{r}_u) \cup [\![\hat{c}]\!]$ such that $(\hat{c}, \hat{r}_u, u', h'_u) \in CP_{NF}$ by Lemma 31. Because of $s_i \in \mathcal{V}$, we have $s_i = \hat{r}_u|_i = u'|_i$ by Lemma 29. Hence u' has a direct variable subterm.

($\Leftarrow$) Suppose (i) of the definition of CP_{NF} holds. One can take σ_α such that $\sigma_\alpha(t) \neq t$. Then $\langle t, \sigma_\alpha(t)\rangle$ is a witness as $t, \sigma_\alpha(t) \in \mathit{NF}$ and $t \overset{*}{\leftrightarrow} c \overset{*}{\leftrightarrow} \sigma_\alpha(t)$. Suppose (ii) holds. Then $s, t \in \mathit{NF}$ by Lemma 29 and $s \overset{*}{\leftrightarrow} \hat{c} \overset{*}{\leftrightarrow} t$ by Lemma 30. Thus $\langle s, t\rangle$ is a witness. ◀

Before ending the section, we present a property regarding T^{CP} (see Definition 14).

▶ **Lemma 34.** *Suppose CP_{NF} is consistent. Then, (1) for any $s, t \in \mathrm{T}^{CP} \cup \mathcal{V}$, $s \overset{*}{\leftrightarrow} t$ implies $s = t$. (2) Suppose $c \overset{*}{\leftrightarrow} t$ for $c \in \mathcal{C}$ and $t \in \mathit{SubMinWit}_{\mathcal{R}}$. Then, there exists a unique $s \in \mathrm{T}^{CP}$ such that $c \overset{*}{\leftrightarrow} s$.*

6 Minimal Witness that is Not Equivalent to a Constant

In the previous section, a sufficient criteria for having a minimal witness that is equivalent a constant is obtained. In this section, we turn our attention to the check whether there exists a witness that is not equivalent to a constant.

We use the following result of [17] as our starting point. For each term t, one can assign a variable $x_{\overline{t}}$ in such a way that $x_{\overline{s}} = x_{\overline{t}}$ if and only if $s \overset{*}{\leftrightarrow} t$. Using this convention, the following definition is given.

▶ **Definition 35** ([17]). *Let δ and ψ be defined as follows: (1) $\delta(t) = t$ if t is equivalent to a constant, and $\delta(t) = x_{\overline{t}}$ otherwise. (2) $\psi(t) = t$ if $t \in \mathcal{C} \cup \mathcal{V}$, and $\psi(t) = f(\delta(t_1), \dots, \delta(t_n))$ if $t = f(t_1, \dots, t_n)$ $(n \geq 1)$.*

▶ **Proposition 36** ([17]). *Let $\langle s, t \rangle$ be a minimal witness that is not equivalent to a constant. Then, either $\langle \psi(s), y \rangle$, $\langle y, \psi(t) \rangle$ or $\langle \psi(s), \psi(t) \rangle$ is a witness for some variable y.*

We first refine δ so that the candidates of $\delta(t)$ form a smaller set. As we focus the case that there is no witness that is equivalent a constant, for the rest of the section, we suppose $\mathit{CP}_{\mathit{NF}}$ is consistent. We refine δ to δ' by substituting a unique term $\tilde{t}$ for $\delta'(t)$ (see section 3.3); the existence of such a term is guaranteed for $t \in \mathit{SubMinWit}_{\mathcal{R}}$ by our assumption just given and Lemma 34.

▶ **Definition 37.** *Let δ' and ψ' be defined as follows: (1) $\delta'(t) = \tilde{t}$ if t is equivalent to a constant, and $\delta'(t) = x_{\overline{t}}$ otherwise. (2) $\psi'(t) = t$ if $t \in \mathcal{C} \cup \mathcal{V}$, and $\psi'(t) = f(\delta'(t_1), \dots, \delta'(t_n))$ if $t = f(t_1, \dots, t_n)$ $(n \geq 1)$.*

The following lemma is readily checked.

▶ **Lemma 38.** *Let $t \in \mathit{SubMinWit}_{\mathcal{R}}$. (1) Then, $\mathrm{root}(t) = \mathrm{root}(\psi(t)) = \mathrm{root}(\psi'(t))$. (2) If $t \in \mathrm{T}_f$ and $\psi(t)|_i \in \mathcal{V}$ then $\psi(t)|_i = \psi'(t)|_i$ for all $i \in \mathbb{N}$. (3) If $t \in \mathrm{T}_f$, then $\psi(t)|_i \overset{*}{\leftrightarrow} \psi'(t)|_i$ for all $i \in \mathbb{N}$. (4) We have $\psi(t) \overset{*}{\leftrightarrow} \psi'(t)$.*

A further property of ψ' is as follows.

▶ **Lemma 39.** *Let $\langle s, t \rangle$ be a minimal witness that is not equivalent to a constant. Then, $s \notin \mathcal{V}$ ($t \notin \mathcal{V}$) implies $\psi'(s) \in \mathit{NF}$ (resp. $\psi'(t) \in \mathit{NF}$).*

We can now refine Proposition 36 as follows.

▶ **Lemma 40.** *Let $\langle s, t \rangle$ be a minimal witness that is not equivalent to a constant. Then, either $\langle \psi'(s), y \rangle$, $\langle y, \psi'(t) \rangle$ or $\langle \psi'(s), \psi'(t) \rangle$ is a witness for some variable y.*

The definition of ψ' gives rise to the following characterization of terms.

▶ **Definition 41.** *We define $\mathrm{T}_f^{CP} = \{f(t_1, \dots, t_n) \mid t_i \in \mathrm{T}^{CP} \cup \mathcal{V}$ for all $i\}$.*

The following lemma will be used later.

▶ **Lemma 42.** *Let $\langle s, t \rangle$ be a witness such that $s, t \in \mathrm{T}_f^{CP} \cup \mathcal{V}$. Then, there exists a proof $s \overset{*}{\leftrightarrow} t$ which has precisely one root rewrite step using a non-trivial equation.*

CW (Definition 15) is now used to further restrict the witnesses class. Because $\widehat{E_{\mathcal{R}}}$ is finite, CW is a finite set, and it can be checked whether an element of CW is a witness:

▶ **Lemma 43.** *Let $\langle s, t \rangle \in \mathit{CW}$. Then, $\langle s, t \rangle$ is a witness if and only if $s, t \in \mathit{NF}$ and $s \neq t$.*

We now arrive at the main result of this section that a witness (if it exists) can be found in CW, if there is no minimal witness equivalent to a constant.

▶ **Theorem 44.** *Suppose CP_{NF} is consistent. If there exists a minimal witness that is not equivalent to a constant, then there exists a witness in CW.*

Proof. We show that if there exists a witness $\langle s,t\rangle$ such that $s,t \in (\mathrm{T}_f^{CP} \cap NF) \cup \mathcal{V}$ then there exists a witness $\langle s',t'\rangle \in CW$. Then the claim follows from Lemma 40, as $\psi'(s), \psi'(t) \in \mathrm{T}_f^{CP} \cap NF$ by Lemma 39. Let $\langle s,t\rangle$ be a witness such that $s,t \in (\mathrm{T}_f^{CP} \cap NF) \cup \mathcal{V}$. Then, w.l.o.g. one can suppose (a) $s = f(u_1,\dots,u_m)$ and $t = g(u_{m+1},\dots,u_{m+n})$, or (b) $s = f(u_1,\dots,u_m)$ and $t = u_{m+1} \in \mathcal{V}$. Now, we repeatedly refine the witness $\langle s,t\rangle$ until we get a desired witness $\langle s',t'\rangle \in CW$.

We first describe one step refinement from $\langle s,t\rangle$ to $\langle s',t'\rangle$ for the case (a). Suppose $s,t \in \mathrm{T}_f^{CP} \cap NF$ with $s = f(u_1,\dots,u_m)$ and $t = g(u_{m+1},\dots,v_{m+n})$.

By Lemma 42, there exist $f(\mu_1,\dots,\mu_m)$, $g(\mu_{m+1},\dots,\mu_{m+n})$ such that

$$s \overset{*}{\leftrightarrow}^{>\epsilon} f(\mu_1,\dots,\mu_m) \leftrightarrow^{\epsilon} g(\mu_{m+1},\dots,\mu_{m+n}) \overset{*}{\leftrightarrow}^{>\epsilon} t$$

Let $f(s_1,\dots,s_m) \approx g(s_{m+1},\dots,s_{m+n}) \in \widehat{E_{\mathcal{R}}}$ be the equation used in the root rewrite step. Then, there exists a substitution σ such that $u_i \overset{*}{\leftrightarrow} \mu_i = \sigma(s_i)$ for all $1 \le i \le m+n$. Suppose there exists $s_i \in \mathcal{V}$ either $u_i = \mu_i \in \mathcal{V}$ does not hold, or there exists j such that $s_i \neq s_j$ and $\mu_i = \mu_j$. If there is no such s_i, the refining step stops.

Let $\{k_1,\dots k_p\} = \{j \in \mathbb{N} \mid s_i = s_j\}$. Then, we have $u_{k_1} \overset{*}{\leftrightarrow} u_{k_2} \overset{*}{\leftrightarrow} \cdots \overset{*}{\leftrightarrow} u_{k_p}$. Since $s,t \in \mathrm{T}_f^{CP}$, $u_{k_1},\dots,u_{k_p} \in \mathrm{T}^{CP} \cup \mathcal{V}$. Thus, Lemma 34 yields $u_{k_1} = \cdots = u_{k_p}$. Now, take a fresh variable x, and let u'_j, μ'_j be x for all $j \in \{k_1,\dots k_p\}$ and u_j, μ_j, respectively, otherwise. Then we obtain a proof

$$s' \overset{*}{\leftrightarrow}^{>\epsilon} f(\mu'_1,\dots,\mu'_m) \leftrightarrow^{\epsilon} g(\mu'_{m+1},\dots,\mu'_{m+n}) \overset{*}{\leftrightarrow}^{>\epsilon} t'$$

where $s' = f(u'_1,\dots,u'_m)$ and $t' = g(u'_{m+1},\dots,u'_{m+n})$.

By construction, it is clear that $s',t' \in \mathrm{T}_f^{CP}$. Since $s \in NF$ and $Patt(s') \subseteq Patt(s)$, we have $s' \in NF$ by Lemma 2. Similarly, $t' \in NF$. Thus, $\langle s',t'\rangle$ is a witness such that $s',t' \in \mathrm{T}_f^{CP} \cap NF$.

Next, we describe one step refinement from $\langle s,t\rangle$ to $\langle s',t'\rangle$ for the case (b). So, assume $s = f(u_1,\dots,u_m)$, $t = u_{m+1} \in \mathcal{V}$ with $s \in \mathrm{T}_f^{CP} \cap NF$.

By Lemma 42, there exists $f(\mu_1,\dots,\mu_m)$ such that

$$s = f(u_1,\dots,u_m) \overset{*}{\leftrightarrow}^{>\epsilon} f(\mu_1,\dots,\mu_m) \leftrightarrow^{\epsilon} t$$

Suppose the equation $f(s_1,\dots,s_m) \approx s_{m+1} \in \widehat{E_{\mathcal{R}}}$ was used for the root rewrite. Suppose there exists $s_i \in \mathcal{V}$ such that either $u_i = \mu_i \in \mathcal{V}$ does not hold, or there exists j such that $s_i \neq s_j$ and $\mu_i = \mu_j$. Then, similar to the case (a), one can obtain a witness $\langle s',t'\rangle$ such that $s' \in \mathrm{T}_f^{CP} \cap NF$ and $t' \in \mathcal{V}$.

Since $\langle s',t'\rangle$ is a witness such that $s',t' \in (\mathrm{T}_f^{CP} \cap NF) \cup \mathcal{V}$, we can repeatedly apply the refinement step above. Since one can iterates the refining steps at most $n+m$-times, eventually one obtains a witness $\langle s'',t''\rangle$ so that, for some equation $f(s_1,\dots,s_m) \approx g(t_1,\dots,t_n) \in \widehat{E_{\mathcal{R}}}$ (note the all renaming equations are in $\widehat{E_{\mathcal{R}}}$), $s''|_i \in \mathrm{T}^{CP} \cup \mathcal{V}$, $s''|_i = s_i$ if $s_i \in \mathcal{V}$, and $s''|_i \overset{*}{\leftrightarrow} s_i$ if $s_i \in \mathcal{C}$ for all $1 \le i \le m$, and similarly for all t_i's.

Furthermore, since there are no $(\hat{c}, \hat{r}_s, s, h_s), (\hat{c}, \hat{r}_t, t, h_t) \in CP_{NF}$ such that $s \neq t$, we have $s'' = \psi''(f(s_1,\dots,s_m))$, $t'' = \psi''(g(t_1,\dots,t_n))$. Hence, $\langle s'',t''\rangle \in CW$. ◀

Table 1 Construction of CP_{NF} by CP algorithm.

$\hat{c}$	$\hat{r}$ (or $\hat{c}$)	normal forms	h
a	a	b	0
	$\mathsf{f}(x, \mathsf{c}, \mathsf{d})$		
c	c	-	-
	$\mathsf{g}(\mathsf{d})$		
d	d	d	0
	$\mathsf{h}(\mathsf{a})$		
e	e	-	-
	$\mathsf{g}(\mathsf{e})$		

(CP table after Step 2)

$\hat{c}$	$\hat{r}$ (or $\hat{c}$)	normal forms	h
a	a	b	0
	$\mathsf{f}(x, \mathsf{c}, \mathsf{d})$	$\mathsf{f}(x, \mathsf{g}(\mathsf{d}), \mathsf{d})$	2
		$\mathsf{f}(x, \mathsf{g}(\mathsf{d}), \mathsf{h}(\mathsf{b}))$	2
c	c	-	-
	$\mathsf{g}(\mathsf{d})$	$\mathsf{g}(\mathsf{d})$	1
d	d	d	0
	$\mathsf{h}(\mathsf{a})$	$\mathsf{h}(\mathsf{b})$	1
e	e	-	-
	$\mathsf{g}(\mathsf{e})$		

(Final CP table)

7 Correctness of the Decision Procedure and Its Complexity

Combining preparations in the previous sections, we now show the correctness of the decision procedure for the UNC property of shallow TRSs in Figure 1. The following is immediate.

▶ **Lemma 45.** *The procedure given in Figure 1 terminates.*

Our main theorem follows from Theorems 33 and 44.

▶ **Theorem 46.** *It can be decided whether a given shallow TRS is UNC or not, by the procedure given in Figure 1.*

We now analyze the complexity of our algorithm. Following [17], the complexity of the algorithm is evaluated in terms of the number of rules in the flat TRS $\mathcal{R}$, and we omit the cost of constructing $\widehat{E_{\mathcal{R}}}$.

In Section 3, we give a *set-based* description of the constant propagation algorithm. To evaluate the complexity, we introduce a data structure *CP table* as illustrated in the following example.

▶ **Example 47.** Let $\mathcal{R}$ and $\widehat{E_{\mathcal{R}}}$ be as in Example 12. One can obtain CP_{NF} as in Table 1 according to following procedure:

1. Enumerate all constants in $\hat{\mathcal{C}}$ and fill the first column of the table.
2. For all $\hat{c} \in \hat{\mathcal{C}}$, enumerate all elements of $[\![\hat{c}]\!]$ and all equations $\hat{c} \approx \hat{r} \in \widehat{E_{\mathcal{R}}}$ $(r \notin \mathcal{C})$ to fill the second column.
3. Fill the 1st, 3rd, 5th and 7th rows according to $CP_{NF} := \{(\hat{c}, \hat{c}, c, 0) \mid c \in \mathcal{C} \cap NF\}$.
4. When an element $(\hat{c}, \hat{r}, t, H)$ is added to CP_{NF} in the main loop of the algorithm, fill the third and forth columns with t, H whose first and second columns correspond to $\hat{c} \approx \hat{r}$.

The table is referred to as a *CP table.* A row of the CP table with non-empty third column corresponds an element of CP_{NF}. Thus, $\chi_H(\hat{c})$ is given by look up of the third columns of the rows with $h < H$ and having $\hat{c}$ at the first column.

▶ **Theorem 48.** *The procedure in Figure 1 runs in $O(\alpha|\mathcal{R}|^{4\alpha+5})$, where α is the maximal arity of function symbols and $|\mathcal{R}|$ is the number of rules in $\mathcal{R}$.*

Proof. Let $\mathcal{R}$ be the flat TRS obtained by the transformation from the input shallow TRS, and $\widehat{E_{\mathcal{R}}}$ the completion of $E_{\mathcal{R}}$. Let $N = |\mathcal{R}|$ and $M = |\widehat{E_{\mathcal{R}}}|$.

Let us first evaluate the CP algorithm. If the set X_{tmp} is non-empty, then $X_{i,\hat{r}}$ is non-empty for some $i, \hat{r}$. Thus in each iteration of the main loop, the number of $Y_{\hat{c}_i}$ reduces for some i. As $\sum_i |Y_{\hat{c}_i}| \leq M$, the number of iterations of the main loop is bounded by M.

In each iteration, the calculation of χ_H is replaced with the look up of the CP table for calculating CP_{NF} (Table 1) Each pair $\langle \hat{c}, \hat{r} \rangle$ in the table is from an equation $c \approx r \in \widehat{E_{\mathcal{R}}}$. If one finds two normal forms for the pair $\langle \hat{c}, \hat{r} \rangle$, during the construction of the CP table, then one can stop the construction and output a counter example. Thus, the height of the CP table is bounded by the number of such pairs, i.e. by M. The calculation of a candidate of new normal form $f(u'_1, \ldots, u'_n)$ is done by representing non-variable direct subterms of $\hat{r}$ as a pointer to an entry of the CP table. Then, one has to check the candidate $f(u'_1, \ldots, u'_n)$ is whether a normal form; as as $u'_1, \ldots, u'_n$ are normal forms and $\mathcal{R}$ is flat, this is done in $O(\alpha N)$. Note that each non-variable u'_i can be identified as a pointer to an entry of the CP table. Thus, the calculation of each entry of the CP table is done in $O(\alpha N)$.

Thus, each iteration of the main loop is bounded by $O(\alpha NM)$, and hence the CP algorithm is bounded by $O(\alpha NM^2)$. During the construction of the CP table the checks in Step 3 can be done in $O(1)$. Thus, this accounts the complexity of the Steps 2, 3.

For the Step 4, first note that computing $\tilde{c}$ costs M. Thus, computing $\psi''(t)$ costs at most αM. Since the size of CW is at most M, one needs $O(\alpha M^2)$ for computing the set CW. For each $\langle s, t \rangle \in CW$, checking whether $s, t \in NF$ needs $O(\alpha N)$, and checking whether $s \neq t$ needs $O(\alpha)$, Since the size of CW is at most M, checking the existence of a witness in CW costs $O(\alpha NM)$. Thus, the Step 4 runs in $O(\alpha M^2 + \alpha NM)$.

Thus, the complexity is dominated by Step 1, that is, $O(\alpha NM^2)$. Now it is known that M is bounded by $O(N^{2\alpha+2})$ [17]. Hence, we conclude that the complexity of the algorithm is $O(\alpha N^{4\alpha+5})$. ◀

▶ Remark 49. It is shown in [17] that the complexity of the algorithm given there is $O(|\mathcal{R}|^{2\alpha+2}(|\mathcal{R}| + (|\mathcal{F}| + \beta + 1)^{O(\beta\alpha^{|\mathcal{C}|})})$, where $\beta = O(\max(\alpha, |\mathcal{C}| - 1)))$. This is of the form $O(N(M + L))$ where L is the number of candidates for witnesses (and N, M as in the proof above). This complexity comes from check $s \neq t$ of the candidates $\langle s, t \rangle$ of the witnesses. In their algorithm, the complexity of the candidates construction part $O(\alpha ML)$ does not affect the final complexity. In contrast, our complexity comes from the candidates construction part $O(\alpha NM^2)$. In this view, a large set L of candidates is reduced to a set of size $O(NM)$ in our algorithm, and the witness checking part is omittable in contrast.

8 Implementations and Experiments

We have implemented our decision procedure, as well as the existing one described in [17]. We use the functional programming language SML/NJ for the implementations.

We have prepared 13 shallow TRSs which covers various situations of the algorithm for our experiments. The timeout was set to 300 seconds. When the execution exceeds 300 seconds, we regard the decision was failed, and we write the execution time as ∞. Our computer used for the experiments has Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz and 4GB memory. Standard ML of New Jersey of v110.79 has been used.

A summary of the experiments is shown in Table 2. Here, “YES” stands for UNC and “NO” stands for Non-UNC; when it cannot judge UNC by timeout, the result is shown as “ - ”. WEC stands for “a witness equivalent to a constant.” The results show that our procedure can judge UNC for all examples 1–13. On the other hand, the existing procedure cannot for examples 7–9,11, because the execution time sharply increased as the rules become complicated. Furthermore, except for very simple examples, the proposed procedure was able to run significantly faster than the existing method.

Table 2 A summary of experiments on examples for various situations.

	Procedure of [17]		Proposed procedure		
TRS	**Result**	**Time(s)**	**Result**	**Time(s)**	features of the TRS
$\mathcal{R}_1$	NO	0.003	NO	0.001	signature with only constants
$\mathcal{R}_2$	NO	0.177	NO	0.002	flat with WEC
$\mathcal{R}_3$	NO	0.001	NO	0.000	flat without WEC
$\mathcal{R}_4$	YES	0.001	YES	0.001	flat, simple, UNC
$\mathcal{R}_5$	NO	0.000	NO	0.000	inconsistent
$\mathcal{R}_6$	YES	0.868	YES	0.000	shallow, simple, height 2
$\mathcal{R}_7$	-	∞	YES	0.000	shallow, simple, height 3
$\mathcal{R}_8$	-	∞	NO	0.014	shallow, complex, with WEC
$\mathcal{R}_9$	-	∞	NO	0.013	shallow, complex, without WEC
$\mathcal{R}_{10}$	YES	0.004	YES	0.001	shallow, simple, UNC
$\mathcal{R}_{11}$	-	∞	YES	0.011	shallow, complex, UNC
$\mathcal{R}_{12}$	NO	0.003	NO	0.001	non-linear, Non-UNC
$\mathcal{R}_{13}$	YES	0.900	YES	0.000	non-linear, UNC

Table 3 A summary of experiments on problems from Cops.

	Procedure of [17]	Proposed procedure
Result	**Num of examples**	**Num of examples**
YES	38	94
NO	18	45
timeout	90	7

We have also tested how procedures fare for the problems from the Cops (confluence problems) database[2]. At the time of the experiment, the database consists of 1137 problems, containing 146 shallow TRSs in it. The timeout was set to 60 sec., which is the timeout used in the Confluence Competition. A summary of the experiments is shown in Table 3. Our procedure succeeds 139 examples and have 7 timeouts, while the previous procedure succeeds 56 examples and have 90 timeouts. Our decision procedure has been also incorporated to the confluence tool ACP [5], which have won the category of UNC in the 2019 edition of Confluence Competition (CoCo 2019)[3].

Our implementations as well as the details of the experiments can be found in the webpage `http://www.nue.ie.niigata-u.ac.jp/experiments/fscd20/`.

9 Conclusion

In this paper, we have proposed a new decision procedure for the UNC property of shallow TRSs. We have introduced a constant propagation algorithm that efficiently constructs candidates of counter examples that are equivalent to a constant. Those candidates have been also used to construct candidates of counter examples that are not equivalent to a constant either. Thus, a large enumeration of candidates for counter examples have been avoided, in contrast to the existing algorithm of [17]. The correctness of the proposed procedure has

[2] `https://cops.uibk.ac.at/`
[3] `http://project-coco.uibk.ac.at/2019/`

been proved and its complexity has been analyzed. Furthermore, we have implemented the proposed decision procedure and the existing one, and it has been experimentally confirmed that the proposed procedure runs much faster than the existing procedure.

References

1 T. Aoto, M. Hamana, N. Hirokawa, A. Middeldorp, J. Nagele, N. Nishida, K. Shintani, and H. Zankl. Confluence competition 2018. In *Proc. of 3rd FSCD*, volume 108 of *LIPIcs*, pages 32:1–32:5. Schloss Dagstuhl, 2018. doi:10.4230/LIPIcs.FSCD.2018.32.

2 T. Aoto, N. Hirokawa, J. Nagele, N. Nishida, and H. Zankl. Confluence competition 2015. In *Proc. of 25th CADE*, volume 9195 of *LNCS*, pages 101–104. Springer-Verlag, 2015. doi:10.1007/978-3-319-21401-6_5.

3 T. Aoto and Y. Toyama. On composable properties of term rewriting systems. In *Proc. of 6th ALP and 3rd HOA*, volume 1298 of *LNCS*, pages 114–128. Springer-Verlag, 1997. doi:10.1007/BFb0027006.

4 T. Aoto and Y. Toyama. Automated proofs of unique normal forms w.r.t. conversion for term rewriting systems. In *Proc. of 12th FroCoS*, volume 11715 of *LNAI*, pages 330–347. Springer-Verlag, 2019. doi:10.1007/978-3-030-29007-8_19.

5 T. Aoto, Y. Yoshida, and Y. Toyama. Proving confluence of term rewriting systems automatically. In *Proc. of 20th RTA*, volume 5595 of *LNCS*, pages 93–102. Springer-Verlag, 2009. doi:10.1007/978-3-642-02348-4_7.

6 F. Baader. and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998. doi:10.1017/CBO9781139172752.

7 L. Bachmair, N. Dershowitz, and D. A. Plaisted. Completion without failure. In H. Aït Kaci and M. Nivat, editors, *Resolution of Equations in Algebraic Structures*, volume 2, pages 1–30. Academic Press, 1989. doi:10.1016/B978-0-12-046371-8.50007-9.

8 H. Comon, M. Haberstrau, and J. P. Jouannaud. Syntacticness, cycle-syntacticness, and shallow theories. *Information and Computation*, 111(1):154–191, 1994. doi:10.1006/inco.1994.1043.

9 M. Dauchet and S. Tison. The theory of ground rewrite systems is decidable. In *Proc. of 5th LICS*, pages 242–248. IEEE Computer Society Press, 1990. doi:10.1109/LICS.1990.113750.

10 R. C. de Vrijer. Conditional linearization. *Indagationes Mathematicae*, 10(1):145–159, 1999. doi:10.1016/S0019-3577(99)80012-3.

11 B. Felgenhauer. Deciding confluence and normal form properties of ground term rewrite systems efficiently. *Logical Methods in Computer Science*, 14(4:7):1–35, 2018. doi:10.23638/LMCS-14(4:7)2018.

12 S. Kahrs and C. Smith. Non-ω-overlapping TRSs are UN. In *Proc. of 1st FSCD*, volume 52 of *LIPIcs*, pages 22:1–17. Schloss Dagstuhl, 2016. doi:10.4230/LIPIcs.FSCD.2016.22.

13 A. Middeldorp. *Modular Properties of Term Rewriting Systems.* PhD thesis, Vrije Universiteit, Amsterdam, 1990.

14 A. Middeldorp, J. Nagele, and K. Shintani. Confluence competition 2019. In *Proc. of 25th TACAS*, volume 11429 of *LNCS*, pages 25–40. Springer-Verlag, 2019. doi:10.1007/978-3-030-17502-3_2.

15 I. Mitsuhashi, M. Oyamaguchi, and F. Jacquemard. The confluence problem for flat TRSs. In *Proc. of 8th AISC*, volume 4120 of *LNAI*, pages 68–81. Springer-Verlag, 2006. doi:10.1007/11856290_8.

16 N. Radcliffe and R. M. Verma. Uniqueness of normal forms is decidable for shallow term rewrite systems. In *Proc. of 30th FSTTCS*, volume 8 of *LIPIcs*, pages 284–295. Schloss Dagstuhl, 2010. doi:10.4230/LIPIcs.FSTTCS.2010.284.

17 N. R. Radcliffe, L. F. T. Moreas, and R. M. Verma. Uniqueness of normal forms for shallow term rewrite systems. *ACM Transactions on Computational Logic*, 18(2):17:1–17:20, 2017. doi:10.1145/3060144.

18 F. Rapp and A. Middeldorp. Automating the first-order theory of rewriting for left-linear right-ground rewrite systems. In *Proc. of 1st FSCD*, volume 52 of *LIPIcs*, pages 36:1–17. Schloss Dagstuhl, 2016. doi:10.4230/LIPIcs.FSCD.2016.36.
19 Y. Toyama and M. Oyamaguchi. Conditional linearization of non-duplicating term rewriting systems. *IEICE Transactions on Information and Systems*, E84-D(4):439–447, 2001.
20 J. Zinn and R. M. Verma. A polynomial algorithm for uniqueness of normal forms of linear shallow term rewrite systems. *Applicable Algebra in Engineering, Communication and Computing*, 21(6):459–485, 2010. doi:10.1007/s00200-010-0133-1.

A Omitted Proofs

Proof of Lemma 6. Let $t = f(t_1, \ldots, t_n)$. Then there exists a minimal witness $\langle u, v\rangle$ such that t is a subterm of u or v. Assume $t_i \xleftrightarrow{*} c \in \mathcal{C} \cap \mathit{NF}$ with $t_i \neq c$. From $u, v \in \mathit{NF}$, we know $t_i \in \mathit{NF}$. Thus, $\langle t_i, c\rangle$ is a witness. But, as $|t_i| + |c| = |t_i| + 1 < |t| + 1 \leq |u| + |v|$, this contradicts the minimality of $\langle u, v\rangle$. ◀

Proof of Lemma 21. Note that, by our convention, $c \approx r \in \widehat{E_{\mathcal{R}}}$ iff $c \approx \sigma_\alpha(r) \in \widehat{E_{\mathcal{R}}}$. ($\subseteq$) Clear. ($\supseteq$) Observe $\sigma_\alpha(r) \xrightharpoonup{*} t'$ implies $r \xrightharpoonup{*} \sigma_\alpha^{-1}(t') \leqslant t' \leqslant t$. ◀

Prof of Lemma 22. (1) For $t \in [\![c]\!]$, we have $t \in \mathcal{C}$ by the definition, and thus, $c \leftrightarrow t$ or $c = t$ by Proposition 4 (1). For $t \in \mathrm{CP}(c, r)$, it is clear from the definition. (2) From (1), we have $c \xrightharpoonup{*} t' \leqslant t$. Hence we have $c \xleftrightarrow{*} t'$ and $\sigma(t') = t$ for some σ. Therefore, $c = c\sigma \xleftrightarrow{*} t'\sigma = t$. ◀

Proof of Lemma 23. (1) follows immediately from Proposition 4 and our convention that $c \approx c \in \widehat{E_{\mathcal{R}}}$ for $c \in \mathcal{C}$. (2) is a consequence of (1) and the inference rule (3) for $\widehat{E}$ ([8], p. 160) and by our convention that $\widehat{E_{\mathcal{R}}}$ is closed under renaming. ◀

Proof of Lemma 24. (1) By Lemma 23. (2) By Proposition 4. (3) Use (2) and Lemma 23. (4) Then $r = f(s_1, \ldots, s_n)$, $r' = f(t_1, \ldots, t_n)$ $(n \geq 1)$, and there exists σ_α such that $s_i = \sigma_\alpha(t_i)$ for all $s_i \in \mathcal{V}$, and $s_i \simeq t_i$ for all $s_i \in \mathcal{C}$. From (1), $t_i \rightharpoonup s_i$ for all $s_i \in \mathcal{C}$. Thus, $\sigma_\alpha(r') \xrightharpoonup{*} r$. ◀

Proof of Theorem 25. By the definition of CPC, we only consider the case $c, c' \in \mathcal{C}$. Since $r \simeq r'$, we have either (a) $r, r' \in V$, (b) $r, r' \in \mathcal{C}$ or (c) $r, r' \in \mathrm{T}_f$. For the cases (a), (b), we have $\mathrm{CP}(c, r) = \emptyset = \mathrm{CP}(c, r)$ by the definition. Thus, assume furthermore, $r, r' \in \mathrm{T}_f$. It suffices to show that (1) $c \simeq c'$ implies $\mathrm{CP}(c, r) = \mathrm{CP}(c', r)$ for any $r \in \mathrm{T}_f$, and (2) $r \simeq r'$ implies $\mathrm{CP}(c, r) = \mathrm{CP}(c, r')$ for any $c \in \mathcal{C}$. (1) follows from Proposition 4. (2) Suppose $r \simeq r'$. Then, by Lemma 24, $\sigma_\alpha(r') \xrightharpoonup{*} r$ for some σ_α. Furthermore, from $r \simeq r'$, we have $c \approx r \in \widehat{E_{\mathcal{R}}}$ iff $c \approx r' \in \widehat{E_{\mathcal{R}}}$ by Lemma 23. Suppose $t \in \mathrm{CP}(c, r)$. Then, we have $c \rightharpoonup r \xrightharpoonup{*} t' \leqslant t$ for some t'. Hence $c \rightharpoonup \sigma_\alpha(r') \xrightharpoonup{*} t' \leqslant t$. Thus, by Lemma 21, we obtain $t \in \mathrm{CP}(c, r')$. ◀

Proof of Lemma 27. $k = \sum_{i=1}^{m} |Y_{\hat{c}_i}|$ is finite for $H = 0$, and the algorithm decreases k in each iteration of the main loop. ◀

Proof of Lemma 30. (1) The proof proceeds by induction on h. (B.S.) Then $\hat{r} = \hat{c}$ and $t = c$. Thus $\hat{c} \simeq t$ and the claim follows by Lemma 24. (I.S.) By Lemma 29, let $\hat{r} = f(\hat{u}_1, \ldots, \hat{u}_n)$, $t = f(u'_1, \ldots, u'_n)$. As $\hat{r} \in Y_{\hat{c}}$, we have $\hat{c} \approx r \in \widehat{E_{\mathcal{R}}}$, and thus, $\hat{c} \approx \hat{r} \in \widehat{E_{\mathcal{R}}}$ by Lemma 24. Hence $\hat{c} \rightharpoonup \hat{r}$. Consider the relation between $\hat{u}_i$ and u'_i, according to the following case distinction:

1. Case $\hat{u}_i \in \mathcal{V}$. Then, by Lemma 29, $u'_i = \hat{u}_i$.

2. Case $\hat{u}_i \in \mathcal{C}$. Then, by the definition of χ^H, either (a) $(\hat{u}_i, \hat{r}_i, u'_i, h'_i) \in CP_{NF}$ for some $\hat{r}_i$ and $h'_i < h_i$ or (b) $\hat{u}_i = u'_i$. In the former case, we have $\hat{u}_i \rightharpoonup \hat{r}_i \xrightharpoonup{*} u'_i$ by induction hypothesis. In the latter case, $\hat{u}_i \xrightharpoonup{*} u'_i$ trivially.

Thus, $\hat{c} \rightharpoonup \hat{r} = f(\hat{u}_1, \ldots, \hat{u}_n) \xrightharpoonup{*} f(u'_1, \ldots, u'_n) = t$. (2) If $h > 0$, then $\hat{r} \in \mathrm{T}_f$ by Lemma 29, and thus the claim follows from (1). (3) is also clear from (1). ◀

Proof of Lemma 31. The proof proceeds by induction on $h = \mathrm{height}(t)$. Let $t \in \mathit{SubMinWit}_{\mathcal{R}} \cap \mathrm{CP}(\hat{c}, \hat{r})$. (B.S.) By $t \in \mathrm{CP}(\hat{c}, \hat{r})$, we have $h > 0$. Thus, the claim trivially holds. (I.S.) By the definition of CPC, one can let $\hat{r} = f(s_1, \ldots, s_n)$ and $t = f(t_1, \ldots, t_n)$, and for some $u = f(u_1, \ldots, u_n)$,

$$c \rightharpoonup^{\epsilon} f(s_1, \ldots, s_n) \xrightharpoonup{*}{}^{>\epsilon} f(u_1, \ldots, u_n) \leqslant f(t_1, \ldots, t_n).$$

Let $\sigma(f(u_1, \ldots, u_n)) = f(t_1, \ldots, t_n)$. By $s_i \xleftrightarrow{*} u_i$, we have $\sigma(s_i) \xleftrightarrow{*} \sigma(u_i) = t_i$. Now, define t'_i and h_i $(1 \leq i \leq n)$ according to the following case distinction:

1. Case $s_i = c_i \in \mathcal{C}$. Then, $\sigma(s_i) = c_i \xleftrightarrow{*} t_i$ holds. Since $\mathcal{R}$ is consistent, $t_i \notin \mathcal{V}$. Since $t_i \in \mathit{SubMinWit}_{\mathcal{R}}$, there exists $\hat{r}_i$ such that $t_i \in \mathrm{CP}(\hat{c}_i, \hat{r}_i) \cup [\![\hat{c}_i]\!]$ by Theorem 26.
   - Case $t_i \in [\![\hat{c}_i]\!]$. Put $t'_i = t_i$ and $h_i = 0$. Since $t_i \in \mathcal{C} \cap \mathit{NF}$, we have $(\hat{c}_i, \hat{c}_i, t'_i, h_i) \in CP_{NF}$ by Lemma 28.
   - Case $t_i \in \mathrm{CP}(\hat{c}_i, \hat{r}_i)$. By induction hypothesis, there exists $t'_i \in \mathrm{CP}(\hat{c}_i, \hat{r}_i)$ such that $(\hat{c}_i, \hat{r}_i, t'_i, h_i) \in CP_{NF}$ with $h_i = \mathrm{height}(t'_i) \leq \mathrm{height}(t_i) < h$. By Lemmas 22 and 30, we have $t_i \xleftrightarrow{*} \hat{c}_i \xleftrightarrow{*} t'_i$. Also, by Lemma 29, $t'_i \in \mathit{NF}$.
2. Case $s_i \in \mathcal{V}$. Put $t'_i = s_i$ and $h_i = 0$.

Let $t' = f(t'_1, \ldots, t'_n)$. We now derive $t' \in \mathit{NF}$ from $t \in \mathit{NF}$ using Lemma 2. From the definition, $t'_1, \ldots, t'_n \in \mathit{NF}$. Clearly, $t'|_i = t|_i$ whenever $t'|_i \in \mathcal{C}$. Also, $\mathrm{root}(t) = \mathrm{root}(t')$. Thus, it remains to show $\mathit{Patt}(t') \subseteq \mathit{Patt}(t)$, i.e. $t_i = t_j$ whenever $t'_i = t'_j$. Suppose $t'_i = t'_j$.

- Case $t'_i = t'_j \notin \mathcal{V}$. Then $s_i, s_j \in \mathcal{C}$, and thus, $t_i \xleftrightarrow{*} t'_i$ and $t_j \xleftrightarrow{*} t'_j$. Hence, $t_i \xleftrightarrow{*} t'_i = t'_j \xleftrightarrow{*} t_j$. Then, since $t \in \mathit{SubMinWit}_{\mathcal{R}}$, Proposition 5 leads $t_i = t_j$.
- Case $t'_i = t'_j \in \mathcal{V}$. Then, $s_i = t'_i = t'_j = s_j \in \mathcal{V}$. Thus, we have $t_i \xleftrightarrow{*} \sigma(s_i) = \sigma(s_j) \xleftrightarrow{*} t_j$. Since $t \in \mathit{SubMinWit}_{\mathcal{R}}$, Proposition 5 leads $t_i = t_j$.

Thus, we obtain $t' \in \mathit{NF}$. Also, $h' = \mathrm{height}(t') = 1 + \max\{\mathrm{height}(t'_i)\}_i \leq h$.

Since $h' = 1 + \max_i h'_i$, there exists $1 \leq j \leq n$ such that $h_j = h' - 1$. Then $(\hat{c}_j, \hat{r}_j, t_j, h_j)$ is added to CP_{NF} at $H = h' - 1$ by Lemma 29. Thus, the main loop the CP algorithm is performed at $H = h'$. Consider the main loop for $H = h'$. If $\hat{r} \notin Y_{\hat{c}}$ already, it is clear that there exists t'' and $h'' \in \mathbb{N}$ such that $(\hat{c}, \hat{r}, t'', h'') \in CP_{NF}$ $(h'' < h')$. From Lemma 30, we have $t'' \in \mathrm{CP}(\hat{c}, \hat{r})$. Hence the claim established in this case. Suppose $\hat{r} \in Y_{\hat{c}_i}$. Then, by the construction above, $s_i = t'_i$ whenever $s_i \in \mathcal{V}$, and $(\hat{s}_i, \hat{r}_i, t'_i, h_i) \in CP_{NF}$ $(h_i < h)$ whenever $s_i \in \mathcal{C}$. Also, $t' \in \mathit{NF}$. Therefore, the main loop of the CP algorithm yields $(\hat{c}, \hat{r}, t', h') \in CP_{NF}$. ◀

Proof of Lemma 32. Suppose $(\hat{c}_s, \hat{r}_s, s, h_s), (\hat{c}_t, \hat{r}_t, t, h_t) \in CP_{NF}$ and $s = t$. From Lemma 30, we have $\hat{c}_s \xleftrightarrow{*} s = t \xleftrightarrow{*} \hat{c}_t$. Thus, $\hat{c}_s \approx \hat{c}_t$, and hence $\hat{c}_s = \hat{c}_t$. By Lemma 30, $s \in \mathrm{CP}(\hat{c}_s, \hat{r}_s) \cup [\![\hat{c}_s]\!]$ and $t \in \mathrm{CP}(\hat{c}_s, \hat{r}_t) \cup [\![\hat{c}_s]\!]$. We distinguish four cases:

1. Case $s \in [\![\hat{c}_s]\!]$ and $t \in [\![\hat{c}_s]\!]$. Then, $s, t \in \mathcal{C}$. Thus, $h_s = h_t = 0$ and $\hat{r}_s = \hat{c}_s = \hat{c}_t = \hat{r}_t$.
2. Case $s \in [\![\hat{c}_s]\!], t \in \mathrm{CP}(\hat{c}_s, \hat{r}_t)$. Then $\mathrm{height}(s) > 0 = \mathrm{height}(t)$, which contradicts $s = t$.
3. Case $s \in \mathrm{CP}(\hat{c}_s, \hat{r}_s), t \in [\![\hat{c}_s]\!]$. Similar to the previous case.

4. Case $s \in \mathrm{CP}(\hat{c}_s, \hat{r}_s), t \in \mathrm{CP}(\hat{c}_s, \hat{r}_t)$. From Lemma 30, we have $\hat{c}_s \rightharpoonup \hat{r}_s \xrightharpoonup{*} s$ and $\hat{c}_s \rightharpoonup \hat{r}_t \xrightharpoonup{*} t$. Thus $\mathrm{root}(\hat{r}_s) = \mathrm{root}(s) = \mathrm{root}(t) = \mathrm{root}(\hat{r}_t)$. Thus, one can let $\hat{r}_s = f(u_1, \ldots, u_n)$, $\hat{r}_t = f(v_1, \ldots, v_n)$, $s = f(s_1, \ldots, s_n) = t$. Furthermore, we have $u_i \xrightharpoonup{*} s_i$ and $v_i \xrightharpoonup{*} s_i$ for all $1 \leq i \leq n$. If $s_i \in \mathcal{V}$ then $u_i = s_i = v_i$ by Lemma 29. Suppose $s_i \notin \mathcal{V}$. Then, $u_i, v_i \in \mathcal{C}$, and thus $\hat{u}_i = u_i$ and $\hat{v}_i = v_i$ (as they are subterms of $\hat{r}_s, \hat{r}_t$). Since $u_i \xleftrightarrow{*} s_i \xleftrightarrow{*} v_i$, we have $u_i \approx v_i$, and hence $u_i = v_i$. Thus, we obtain $\hat{r}_s = f(u_1, \ldots, u_n) = f(v_1, \ldots, v_n) = \hat{r}_t$. ◀

Proof of Lemma 34. (1) Suppose $s \xleftrightarrow{*} t$. By Lemma 29, $\mathrm{T}^{CP} \cap \mathcal{V} = \emptyset$. Thus, we can distinguish four cases. The case $s, t \in \mathrm{T}^{CP}$ follows from the assumption, and the case $s, t \in \mathcal{V}$ follows from the consistency of $\mathcal{R}$. If $s \in \mathrm{T}^{CP}$ and $t \in \mathcal{V}$ then $(\hat{c}, \hat{r}, s, h_s) \in CP_{NF}$ for some $\hat{c}, \hat{r}, h_s$, and thus $\hat{c} \xleftrightarrow{*} s \xleftrightarrow{*} t \in \mathcal{V}$ by Lemma 30. This contradicts the consistency of $\mathcal{R}$. The case $t \in \mathrm{T}^{CP}$ and $s \in \mathcal{V}$ follows similarly. (2) Suppose $c \xleftrightarrow{*} t$ for $c \in \mathcal{C}$ and $t \in \mathit{SubMinWit}_{\mathcal{R}}$. Then, by Theorem 26, $t \in \mathrm{CP}(c, r) \cup [\![c]\!]$. Then, by Lemma 31, $(\hat{c}, \hat{r}, s, \_) \in CP_{NF}$ for some s. Then, we have $s \in \mathrm{T}^{CP}$ and $c \xleftrightarrow{*} \hat{c} \xleftrightarrow{*} s$ by Lemma 30. If $s' \in \mathrm{T}^{CP}$ and $c \xleftrightarrow{*} s'$, then $s \xleftrightarrow{*} c \xleftrightarrow{*} s'$, and thus $s = s'$ by (1). ◀

Proof of Lemma 38. (1), (2) are immediate. (3) If $t|_i$ is not equivalent to a constant, from definitions of ψ and ψ', we have $\psi(t)|_i = \psi'(t)|_i$. If $t|_i$ is equivalent to a constant c, we have $c \xleftrightarrow{*} t|_i$. From the definition of ψ, $t|_i = \psi(t)|_i$. Also, from the definition of ψ', $c \xleftrightarrow{*} \psi'(t)|_i$. Hence $\psi(t)|_i \xleftrightarrow{*} \psi'(t)|_i$. (4) Clear from (1) and (3). ◀

Proof of Lemma 39. From Lemma 29, we have $t'_1, \ldots, t'_n \in \mathit{NF}$. Thus it suffices to show there's no root rewrite step from $\psi'(s)$. It is known that $\psi(s) \in \mathit{NF}$ [17]; thus (as $\mathcal{R}$ is flat) the claim follows if we have: (a) $t'_i \in \mathcal{C}$ implies $t'_i = t_i$ for all $1 \leq i \leq n$ and (b) $\mathit{Patt}(\psi'(s)) \subseteq \mathit{Patt}(\psi(s))$. To show (a), suppose $t'_i \in \mathcal{C}$. Then $t'_i \in \mathcal{C} \cap \mathit{NF}$. From Lemma 38, $t_i \xleftrightarrow{*} t'_i$ holds. Also, by definition of ψ, $u_i = t_i$ holds. Thus, $u_i = t_i \xleftrightarrow{*} t'_i \in \mathcal{C} \cap \mathit{NF}$. Since $s = f(u_1, \ldots, u_n) \in \mathit{SubMinWit}_{\mathcal{R}}$, $t_i = u_i = t'_i$ holds by Lemma 6. Next, we show (b). Suppose $t'_i = t'_j$. We distinguish two cases:

- Case $t'_i = t'_j \notin \mathcal{V}$. Then u_i, u_j are equivalent to constants, and hence $u_i = t_i$ and $u_j = t_j$ by the definition of ψ. By $u_i = t_i \xleftrightarrow{*} t'_i = t'_j \xleftrightarrow{*} t_j = u_j$ and $s \in \mathit{SubMinWit}_{\mathcal{R}}$, we obtain $t_i = u_i = u_j = t_j$ by Proposition 5.
- Case $t'_i = t'_j \in \mathcal{V}$. Then $t_i = t'_i = t'_j = t_j$ by definitions of ψ and ψ'. ◀

Proof of Lemma 40. Let $\langle s, t \rangle$ be a minimal witness that is not equivalent to a constant. By Proposition 36, either $\langle \psi(s), y \rangle$, $\langle y, \psi(t) \rangle$ or $\langle \psi(s), \psi(t) \rangle$ is a witness for some variable y. By Lemma 38, $\psi(s) \xleftrightarrow{*} \psi'(s)$ and $\psi(t) \xleftrightarrow{*} \psi'(t)$ hold. We distinguish three cases.

- Case $\langle \psi(s), y \rangle$ is a witness. Then $\psi(s) \notin \mathcal{V}$ as $\mathcal{R}$ is consistent. By the definitions, we have $\psi(s) \notin \mathcal{V}$ iff $s \notin \mathcal{V}$ iff $\psi'(s) \notin \mathcal{V}$, and thus $\psi'(s) \in \mathit{NF} \setminus \mathcal{V}$ by Lemma 39. Since $\psi'(s) \xleftrightarrow{*} \psi(s) \xleftrightarrow{*} y$, $\langle \psi'(s), y \rangle$ is also a witness.
- Case $\langle y, \psi(t) \rangle$ is a witness. Same as the previous case.
- Case $\langle \psi(s), \psi(t) \rangle$ is a witness. If $\psi(s) \in \mathcal{V}$ or $\psi(t) \in \mathcal{V}$, then one can use the same argument as above. So, suppose $\psi(s), \psi(t) \notin \mathcal{V}$. Then, $\psi'(s), \psi'(t) \notin \mathcal{V}$ as above, and thus $\psi'(s), \psi'(t) \in \mathit{NF}$ by Lemma 39. Since $\psi'(s) \xleftrightarrow{*} \psi(s) \xleftrightarrow{*} \psi(t) \xleftrightarrow{*} \psi'(t)$, it remains to show $\psi'(s) \neq \psi'(t)$. If $\mathrm{root}(\psi(s)) \neq \mathrm{root}(\psi(t))$ or if $\psi(s)|_i \neq \psi(t)|_i$ with $\psi(s)|_i \in \mathcal{V}$ or $\psi(t)|_i \in \mathcal{V}$ for some $i \in \mathbb{N}$, then it follows from Lemma 38 that $\psi'(s) \neq \psi'(t)$. Consider the case where roots of $\langle \psi(s), \psi(t) \rangle$ and all direct variable subterms are same. Then, we have $\psi(s)|_i \neq \psi(t)|_i$ and $\psi(s)|_i, \psi(t)|_i \notin \mathcal{V}$ for some $i \in \mathbb{N}$. Then by definition of ψ, we have $s|_i = \psi(s)|_i$ and $t|_i = \psi(t)|_i$. Thus, $s|_i \neq t|_i$. Then, if $\psi'(s)|_i = \psi'(t)|_i$, then $s|_i = \psi(s)|_i \xleftrightarrow{*} \psi'(s)|_i = \psi'(t) \xleftrightarrow{*} \psi'(t)|_i = t|_i$ by Lemma 38. Then $\langle s|_i, t|_i \rangle$ becomes a witness, which contradicts the minimaliy of $\langle s, t \rangle$. ◀

Proof of Lemma 42. From Proposition 4, there exists a proof $s \overset{*}{\leftrightarrow} t$ which has at most one root rewrite step. Clearly, one can assume the root step by a trivial equation have been removed. Suppose $s \overset{*}{\leftrightarrow} t$ does not have a root rewrite step. Then, $s = f(s_1, \dots s_n)$, $t = f(t_1, \dots t_n)$ and $s_i \overset{*}{\leftrightarrow} t_i$ for all $1 \leq i \leq n$ for some f and s_i, t_i $(1 \leq i \leq n)$. Then, $s, t \in \mathrm{T}_f^{CP}$ and hence $s_i, t_i \in \mathrm{T}^{CP} \cup \mathcal{V}$ for all $1 \leq i \leq n$. Then, from Lemma 34, we know $s_i = t_i$ for all $1 \leq i \leq n$. Thus, $s = f(s_1, \dots s_n) = f(t_1, \dots t_n) = t$, but this contradicts that $\langle s, t \rangle$ is a witness. ◀

Proof of Lemma 43. Let $\langle s, t \rangle \in CW$. Then, there exists $l \approx r \in \widehat{E_{\mathcal{R}}}$ such that $\psi''(l) = s$ and $\psi''(r) = t$. It immediately follows from the definition of ψ'' that $\psi''(l) \overset{*}{\leftrightarrow} l$ and $r \overset{*}{\leftrightarrow} \psi''(r)$. Hence $s \overset{*}{\leftrightarrow} t$. The claim is an easy consequence of this. ◀

Proof of Lemma 45. Step 1 terminates and a flat TRS $\mathcal{R}$ and a finite set $\widehat{E_{\mathcal{R}}}$ is computed [8, 20]. Step 2 terminates by Lemma 27, and a finite set CP_{NF} is obtained. As the set CP_{NF} is finite, Step 3 terminates. As a consequence of Step 3, $\tilde{t}$ is defined uniquely. Thus, by the finiteness of CP_{NF} and $\widehat{E_{\mathcal{R}}}$, a finite set CW can be computed. By Lemma 43, one can check whether there exists a witness in CW. Thus, Step 4 terminates. ◀

Proof of Theorem 46. It suffices to decide that the flat TRS $\mathcal{R}$ obtained by the transformation has the UNC property or not. If $\mathcal{R}$ is inconsistent, then $\mathcal{R}$ is not UNC. In this case, UNC is returned at the Step 1 of the procedure. For the rest of the procedure, one can assume $\mathcal{R}$ is consistent. Suppose there exists $(\hat{c}, \hat{r}, t, h) \in CP_{NF}$ such that t has a direct variable subterm or there exist $(\hat{c}, \hat{r}_s, s, h_s), (\hat{c}, \hat{r}_t, t, h_t) \in CP_{NF}$ such that $s \neq t$. Then $\mathcal{R}$ is not UNC by Theorem 33. In this case, Non-UNC is returned as the Step 3 of the procedure. Suppose this does not hold. Then, by Theorem 33, there exists no minimal witness equivalent to a constant. If there exists a minimal witness that is not equivalent to a constant, then there exists a witness in CW by Theorem 44. Thus, in this case, Non-UNC is returned as the Step 4 of the procedure. Suppose otherwise. Clearly, if a witness exists, then there exists a minimal witness. Thus, one can conclude that there is no witness, and hence $\mathcal{R}$ has the UNC property. In this case, UNC is returned as the Step 5 of the procedure. ◀

B Examples in our Experiments

Bellow we present examples used in our experiments given in Table 2.

$$
\begin{aligned}
\mathcal{R}_1 &= \{\mathsf{a} \to \mathsf{b},\ \mathsf{a} \to \mathsf{c},\ \mathsf{c} \to \mathsf{c},\ \mathsf{d} \to \mathsf{c},\ \mathsf{d} \to \mathsf{e}\} \\
\mathcal{R}_2 &= \{\mathsf{f}(x,\mathsf{a}) \to \mathsf{a},\ \mathsf{a} \to \mathsf{b}\} \\
\mathcal{R}_3 &= \{\mathsf{g}(y) \to \mathsf{f}(x,y)\} \\
\mathcal{R}_4 &= \{\mathsf{a} \to \mathsf{b},\ \mathsf{b} \to \mathsf{a},\ \mathsf{f}(x,y) \to \mathsf{a}\} \\
\mathcal{R}_5 &= \{\mathsf{a} \to \mathsf{b},\ \mathsf{b} \to \mathsf{a},\ \mathsf{f}(x,y) \to \mathsf{a},\ \mathsf{f}(x,y) \to z\} \\
\mathcal{R}_6 &= \{\mathsf{f}(\mathsf{g}(\mathsf{a}),y) \to y\} \\
\mathcal{R}_7 &= \{\mathsf{f}(\mathsf{g}(\mathsf{h}(\mathsf{a})),y) \to y\} \\
\mathcal{R}_8 &= \left\{ \begin{array}{l} \mathsf{f}(x,y) \to \mathsf{a},\ \mathsf{f}(\mathsf{k}(\mathsf{l}(\mathsf{a}_1,\mathsf{a}_2),\mathsf{a}_1,\mathsf{a}_3),y) \to \mathsf{a},\ \mathsf{f}(x,\mathsf{u}) \to \mathsf{a},\ \mathsf{a} \to \mathsf{g}(\mathsf{b}_1,\mathsf{u},x),\ \mathsf{b}_1 \to \mathsf{b}, \\ \mathsf{b} \to \mathsf{h}(x,\mathsf{d}),\ \mathsf{h}(y,\mathsf{d}) \to \mathsf{i}(\mathsf{c}),\ \mathsf{u} \to \mathsf{j}(\mathsf{v}) \end{array} \right\} \\
\mathcal{R}_9 &= \left\{ \begin{array}{l} \mathsf{f}(x,y) \to \mathsf{a},\ \mathsf{f}(\mathsf{k}(\mathsf{l}(\mathsf{a}_1,\mathsf{a}_2),\mathsf{a}_1,\mathsf{a}_3),y) \to \mathsf{a},\ \mathsf{f}(x,\mathsf{u}) \to \mathsf{a},\ \mathsf{a} \to \mathsf{g}(\mathsf{b}_1,\mathsf{u}),\ \mathsf{b}_1 \to \mathsf{b} \\ \mathsf{b} \to \mathsf{h}(x,\mathsf{d}),\ \mathsf{h}(y,\mathsf{d}) \to \mathsf{i}(\mathsf{c}),\ \mathsf{u} \to \mathsf{j}(\mathsf{v}),\ \mathsf{m}(x,x) \to \mathsf{n}(y) \end{array} \right\} \\
\mathcal{R}_{10} &= \left\{\ \mathsf{f}(x,y) \to \mathsf{g}(\mathsf{h}(\mathsf{a}))\ \right\} \\
\mathcal{R}_{11} &= \left\{ \begin{array}{l} \mathsf{f}(x,y) \to \mathsf{a},\ \mathsf{f}(\mathsf{k}(\mathsf{l}(\mathsf{a}_1,\mathsf{a}_2),\mathsf{a}_1,\mathsf{a}_3),y) \to \mathsf{a},\ \mathsf{f}(x,\mathsf{u}) \to \mathsf{a},\ \mathsf{a} \to \mathsf{g}(\mathsf{b}_1,\mathsf{u}),\ \mathsf{b}_1 \to \mathsf{b} \\ \mathsf{b} \to \mathsf{h}(x,\mathsf{d}).\ \mathsf{h}(y,\mathsf{d}) \to \mathsf{i}(\mathsf{c}),\ \mathsf{u} \to \mathsf{j}(\mathsf{v}) \end{array} \right\} \\
\mathcal{R}_{12} &= \{\mathsf{f}(x) \to \mathsf{g}(\mathsf{a},x),\ \mathsf{f}(x) \to \mathsf{g}(x,\mathsf{a}),\ \mathsf{g}(x,x) \to \mathsf{f}(x)\} \\
\mathcal{R}_{13} &= \{\mathsf{f}(x,x) \to \mathsf{g}(x),\ \mathsf{f}(\mathsf{a},\mathsf{b}) \to \mathsf{g}(\mathsf{a})\}
\end{aligned}
$$

C Implementation of Existing Decision Procedure

Here, we briefly explain our implementation of the existing decision procedure [17] and illustrate why it suffers a bottleneck of having a sheer number of candidates for the witness.

We use the following algorithm:

1. Transform a shallow TRS into a flat TRS $\mathcal{R}$ preserving UNC.
2. Calculate $\widehat{E_{\mathcal{R}}}$. (Here, the program also judges whether $\mathcal{R}$ is consistent.)
3. Add new constants $\mathcal{C}_{new}$ to $\mathcal{F}$ of the size $|\mathcal{C}_{new}| = 2 * \alpha^{h-1}$, where $h = \max(1, |\mathcal{C}|)$ and $\alpha = \max\{\text{arity}(f) \mid f \in \mathcal{F}\}$.
4. Make all ground terms over $\mathcal{F} \cup \mathcal{C}_{new}$ of height $\leq h$.
5. Check whether there exists a pair of such terms that is a witness.

Below, we provide a (straight) estimation of the number of candidates for the witness for our examples $\mathcal{R}_6$ and $\mathcal{R}_7$. These examples are very similar (see the previous section) but our implementation of the existing procedure succeeds for $\mathcal{R}_6$ but fails for $\mathcal{R}_7$.

▶ **Example 50.** Consider the TRS $\mathcal{R}_6$. The UNC-preserving flatting translation makes the following flat TRS $\{\mathsf{f}(\mathsf{c}_0, x) \to x, \mathsf{g}(\mathsf{a}) \to \mathsf{c}_0\}$. Since we have $h = |\mathcal{C}| = 2$ and $\alpha = 2$, we add $2 * 2^1 = 4$ new constants. Hence, we consider ground terms over function symbols f, g and $2 + 4 = 6$ constants. We have 6 ground terms of height 0. There are 6 ground terms of height 1 having root g and $6 \times 6 = 36$ ground terms of height 1 having root f. Thus, the number of ground terms of height ≤ 2 to be used to constructing the candidates is $6 + 6 + 42 = 48$. This means there are $48 \times 47 = 2,256$ candidates to be checked.

▶ **Example 51.** Consider the TRS $\mathcal{R}_7$. The UNC-preserving flatting translation makes the following flat TRS $\{\mathsf{f}(\mathsf{c}_1, x) \to x, \mathsf{g}(\mathsf{c}_0) \to \mathsf{c}_1, \mathsf{h}(\mathsf{a}) \to \mathsf{c}_0\}$. Since we have $h = |\mathcal{C}| = 3$ and

$\alpha = 2$, we add $2 * 2^2 = 8$ new constants. Hence, we consider ground terms over function symbols f, g and $3 + 8 = 11$ constants. We have 11 ground terms of height 0. There are 11 ground terms of height 1 having root g, and $11 \times 11 = 121$ ground terms of height 1 having root f. Thus, the number of ground terms of height 1 to be used is $11 + 121 = 132$. There are 132 ground terms of height 2 having root g. Now we calculate the number of ground terms of height 2 having root f distinguishing three cases. The number of terms $\mathsf{f}(s,t)$ with $\mathit{height}(s) = 0$, $\mathit{height}(t) = 1$ is $11 \times 121 = 1331$; so is the number of terms $\mathsf{f}(s,t)$ with $\mathit{height}(s) = 1$, $\mathit{height}(t) = 0$. The number of terms $\mathsf{f}(s,t)$ with $\mathit{height}(s) = \mathit{height}(t) = 1$ is $121 \times 121 = 14,642$. Thus, we have $132 + 1331 + 1331 + 14,641 = 17,435$ ground terms of height 2. Hence,the number of ground terms of height ≤ 2 be used to constructing the candidates is $11 + 121 + 17435 = 17567$. This means there are 17567×17566 (≈ 300 millions) candidates to be checked.