

Certified Equational Reasoning via Ordered Completion^{*}

Christian Sternagel¹[0000-0001-9864-1014] and
Sarah Winkler¹[0000-0001-8114-3107]

Department of Computer Science, University of Innsbruck, Austria
{christian.sternagel,sarah.winkler}@uibk.ac.at

Abstract. On the one hand, equational reasoning is a fundamental part of automated theorem proving with ordered completion as a key technique. On the other hand, the complexity of corresponding, often highly optimized, automated reasoning tools makes implementations inherently error-prone. As a remedy, we provide a formally verified certifier for ordered completion based techniques. This certifier is code generated from an accompanying Isabelle/HOL formalization of ordered rewriting and ordered completion incorporating an advanced ground joinability criterion. It allows us to rigorously validate generated proof certificates from several domains: ordered completion, satisfiability in equational logic, and confluence of conditional term rewriting.

Keywords: Equational reasoning · Ordered Completion · Ground Joinability · Certification.

1 Introduction

Equational reasoning constitutes a main area of automated theorem proving in which completion has evolved as a fundamental technique [8]. Completion aims to transform a given set of equations into a terminating and confluent rewrite system that induces the same equational theory. Thus, on success, such a rewrite system can be used to decide equivalence of terms with respect to the initial set of equations. The original completion procedure may fail due to unorientable equations. As a remedy to this problem, ordered completion—also known as unifying completion—was developed [3]. As the name suggests, unifying completion always yields a result (which may however be infinite and thus take infinitely many inference steps to compute). This time, the result is an ordered rewrite system (given by a ground total reduction order, a set of rules which are oriented with respect to this order, and a set of equations) that is still terminating, but in general only ground confluent (that is, confluent on ground terms). Thus, the resulting system can be used to decide equivalence of *ground* terms with respect to the initial set of equations. This suffices for many practical purposes: A well-known success story of ordered completion is the

^{*}This work is supported by Austrian Science Fund (FWF) projects T789 and P27502.

solution of the long-standing Robbins conjecture [10], followed by applications to other problems from (Boolean) algebra [11]. More recent applications include the use of ordered completion in algebraic data integration [14] and in confluence proofs of conditional term rewrite systems [20].

As an introductory example, let us illustrate ordered completion on the following set of equations describing a group where all elements are self-inverse:

$$f(x, y) \approx f(y, x) \quad f(x, f(y, z)) \approx f(f(x, y), z) \quad f(x, x) \approx 0 \quad f(x, 0) \approx x$$

Using ordered completion, the tool MædMax [24] transforms it into the following rules (\rightarrow) and equations (\approx), together with a suitable ground total reduction order $>$ that orients all rules from left to right.

$$\begin{array}{llll} f(x, f(x, y)) \rightarrow f(0, y) & f(x, f(y, x)) \rightarrow f(0, y) & f(x, x) \rightarrow 0 & f(x, 0) \rightarrow x \\ f(f(x, y), z) \rightarrow f(x, f(y, z)) & f(0, x) \rightarrow x & & \\ f(x, f(y, z)) \approx f(y, f(x, z)) & f(x, y) \approx f(y, x) & & \end{array}$$

This ordered rewrite system can be used to decide a given equation between ground terms, by checking whether the unique normal forms (with respect to ordered rewriting using $>$) of both terms coincide.

Automated reasoning tools are highly sophisticated pieces of software, not only because they implement complex calculi, but also due to their high degree optimization. Consequently, their implementation is inherently error-prone.

To improve their trustability we follow a two-staged certification approach and (1) add the relevant concepts and results regarding ordered completion to a formal library using the proof assistant Isabelle/HOL [12] (version Isabelle2019), and from there (2) code generate [5] a trusted certifier that is correct by construction. Our formalization strengthens the originally proposed procedure [3] by using a relaxed version of the inference system, while incorporating a stronger ground joinability criterion [9]. Our certifier allows us to rigorously validate generated proof certificates from several domains: ordered completion, satisfiability in equational logic, and confluence of conditional term rewriting.


More specifically, our contributions are as follows:

- We extend the existing *Isabelle Formalization of Rewriting*¹ (IsaFoR for short) by ordered rewriting and a generalization of the ordered completion calculus oKB [3], and prove the latter correct for finite completion runs with respect to ground total reduction orders (Section 3).
- We establish ground totality of the Knuth-Bendix order and the lexicographic path order in IsaFoR (Section 3).
- We formalize two criteria for ground joinability [3,9] known from the literature, that allow us to apply our previous results to concrete completion runs (Section 4). In fact, we present a slightly more powerful version of the latter, and fix an error in its proof, as described below.

¹ <http://cl-informatik.uibk.ac.at/isafor>

- We apply ordered completion to satisfiability in equational logic and infeasibility of conditions in conditional rewriting (Section 5).
- We extend the XML-based *certification problem format* (CPF for short) [17] by certificates for ordered completion and formalize corresponding executable check functions that verify the supplied derivations (Section 6).
- Finally, we extend the completion tool **MædMax** [24], as well as the confluence tool **ConCon** [20] by certificate generation and evaluate our approach on existing benchmarks (Section 7).

As a result, **CeTA** (the certifier accompanying **IsaFoR**) can now certify (a) ordered completion proofs and (b) satisfiability proofs of equational logic produced by the tool **MædMax**, as well as (c) conditional confluence proofs by **ConCon** where infeasibility of critical pairs is established via equational logic. To the best of our knowledge, **CeTA** constitutes the first proof checker in all of these domains.

In the remainder we provide hyperlinks (marked by ) to an HTML rendering of our formalization.

This work is an extension of an earlier workshop paper [19]. Further note that the **IsaFoR** formalization of the results in this paper is, apart from very basic results on (ordered) rewriting, entirely disjoint from our previous formalization together with Hirokawa and Middeldorp [6]. On the one hand, we consider a relaxed completion inference system where more inferences are allowed. This is possible since we are only interested in finite completion runs. On the other hand, we employ a stronger ground joinability criterion. Another major difference is that our new formalization enables actual certification of ordered completion based techniques, which is not the case for our work with Hirokawa and Middeldorp.

2 Preliminaries

In the sequel, we use standard notation from term rewriting [2]. Let $\mathcal{T}(\mathcal{F}, \mathcal{V})$ denote the *set of all terms* over a signature \mathcal{F} and an infinite set of variables \mathcal{V} , and $\mathcal{T}(\mathcal{F})$ the *set of all ground terms* over \mathcal{F} (that is, terms without variables). A *substitution* σ is a mapping from variables to terms. As usual, we write $t\sigma$ for the *application* of σ to the term t . A *variable permutation* (or *renaming*) π is a bijective substitution such that $\pi(x) \in \mathcal{V}$ for all $x \in \mathcal{V}$. Given an equational system (ES) \mathcal{E} , we write $\mathcal{E}^{\leftrightarrow}$ to denote its *symmetric closure* $\mathcal{E} \cup \{t \approx s \mid s \approx t \in \mathcal{E}\}$. A *reduction order* is a proper and well-founded order on terms which is closed under contexts and substitutions. It is *\mathcal{F} -ground total* if it is total on $\mathcal{T}(\mathcal{F})$. In the remainder we often focus on the Knuth-Bendix order (KBO), written $>_{\text{kbo}}$, and the lexicographic path order (LPO), written $>_{\text{lpo}}$. Given a reduction order $>$ and an ES \mathcal{E} , the term rewrite system (TRS) $\mathcal{E}_>$ consists of all rules $s\sigma \rightarrow t\sigma$ such that $s \approx t \in \mathcal{E}^{\leftrightarrow}$ and $s\sigma > t\sigma$.

Given a reduction order $>$, an *extended overlap* consists of two variable-disjoint variants $\ell_1 \approx r_1$ and $\ell_2 \approx r_2$ of equations in $\mathcal{E}^{\leftrightarrow}$ such that $p \in \text{Pos}_{\mathcal{F}}(\ell_2)$ and ℓ_1 and $\ell_2|_p$ are unifiable with most general unifier μ . An extended overlap which in addition satisfies $r_1\mu \not\approx \ell_1\mu$ and $r_2\mu \not\approx \ell_2\mu$ gives rise to the *extended critical pair* $\ell_2[r_1]_p\mu \approx r_2\mu$. The set $\text{CP}_>(\mathcal{E})$ consists of all extended critical pairs between

equations in \mathcal{E} . A relation on terms is (*ground*) *complete*, if it is terminating and confluent (on ground terms). A TRS \mathcal{R} is (ground) complete whenever the induced rewrite relation $\rightarrow_{\mathcal{R}}$ is. Finally, we say that a TRS \mathcal{R} is a presentation of an ES \mathcal{E} , whenever $\leftrightarrow_{\mathcal{E}}^* = \leftrightarrow_{\mathcal{R}}^*$ (that is, their equational theories coincide).

A substitution σ is *grounding* for a term t if $\sigma(x) \in \mathcal{T}(\mathcal{F})$ for all $x \in \text{Var}(t)$. Two terms s and t are called *ground joinable* over a rewrite system \mathcal{R} , denoted $s \downarrow_{\mathcal{R}}^{\mathcal{E}} t$ if $s\sigma \downarrow_{\mathcal{R}} t\sigma$ for all substitutions σ that are grounding for s and t .

For any complete rewrite relation \rightarrow , we denote the (necessarily unique) *normal form* of a term t (that is, the term u such that we have $t \rightarrow^* u$ but $u \not\rightarrow v$ for all terms v) by $t \downarrow$. By an *ordered rewrite system* we mean a pair $(\mathcal{E}, \mathcal{R})$, consisting of an ES \mathcal{E} and a TRS \mathcal{R} , together with a reduction order $>$. Then, *ordered rewriting* is rewriting with respect to the TRS $\mathcal{R} \cup \mathcal{E}_{>}$. Note that ordered rewriting is always terminating if $\mathcal{R} \subseteq >$. Take commutativity $x * y \approx y * x$ for example, which causes nontermination when used as a rule in a TRS. Nevertheless, the ordered rewrite system $(\{x * y \approx y * x\}, \emptyset)$ together with KBO, say with precedence $* > a > b$, is terminating and we can for example rewrite $a * b$ to $b * a$ since applying the substitution $\{x \mapsto a, y \mapsto b\}$ to the commutativity equation results in a KBO-oriented instance.

3 Formalized Ordered Completion

Ordered completion is commonly presented as a set of inference rules, parameterized by a fixed reduction order $>$. This way of presentation conveniently leaves a lot of freedom to implementations. We use the following inference system, with some differences to the original formulation [3] that we discuss below.

Definition 1 (Ordered Completion \checkmark). *The inference system oKB of ordered completion operates on pairs $(\mathcal{E}, \mathcal{R})$ of equations \mathcal{E} and rules \mathcal{R} over a common signature \mathcal{F} . It consists of the following inference rules, where \mathcal{S} abbreviates $\mathcal{R} \cup \mathcal{E}_{>}$ and π is a renaming.*

$$\begin{array}{l}
\text{deduce} \quad \frac{\mathcal{E}, \mathcal{R}}{\mathcal{E} \cup \{s\pi \approx t\pi\}, \mathcal{R}} \quad \text{if } s \xrightarrow[\mathcal{R} \cup \mathcal{E}^{\leftrightarrow}]{} t \quad \text{compose} \quad \frac{\mathcal{E}, \mathcal{R} \uplus \{s \rightarrow t\}}{\mathcal{E}, \mathcal{R} \cup \{s\pi \rightarrow u\pi\}} \quad \text{if } t \rightarrow_{\mathcal{S}} u \\
\text{orient} \quad \frac{\mathcal{E} \uplus \{s \approx t\}, \mathcal{R}}{\mathcal{E}, \mathcal{R} \cup \{s\pi \rightarrow t\pi\}} \quad \text{if } s > t \quad \text{simplify} \quad \frac{\mathcal{E} \uplus \{s \approx t\}, \mathcal{R}}{\mathcal{E} \cup \{u\pi \approx t\pi\}, \mathcal{R}} \quad \text{if } s \rightarrow_{\mathcal{S}} u \\
\quad \frac{\mathcal{E} \uplus \{s \approx t\}, \mathcal{R}}{\mathcal{E}, \mathcal{R} \cup \{t\pi \rightarrow s\pi\}} \quad \text{if } t > s \quad \quad \frac{\mathcal{E} \uplus \{s \approx t\}, \mathcal{R}}{\mathcal{E} \cup \{s\pi \approx u\pi\}, \mathcal{R}} \quad \text{if } t \rightarrow_{\mathcal{S}} u \\
\text{delete} \quad \frac{\mathcal{E} \uplus \{s \approx s\}, \mathcal{R}}{\mathcal{E}, \mathcal{R}} \quad \text{collapse} \quad \frac{\mathcal{E}, \mathcal{R} \uplus \{t \rightarrow s\}}{\mathcal{E} \cup \{u\pi \approx s\pi\}, \mathcal{R}} \quad \text{if } t \rightarrow_{\mathcal{S}} u
\end{array}$$

We write $(\mathcal{E}, \mathcal{R}) \vdash (\mathcal{E}', \mathcal{R}')$ if $(\mathcal{E}', \mathcal{R}')$ is obtained from $(\mathcal{E}, \mathcal{R})$ by employing one of the above inference rules. A finite sequence of inference steps

$$(\mathcal{E}_0, \emptyset) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash \dots \vdash (\mathcal{E}_n, \mathcal{R}_n)$$

is called a *run*. Definition 1 differs from the original formulation of ordered completion [3] (as well as the formulation in our previous work together with Hirokawa and Middeldorp [6]) in two ways. First, *collapse* and *simplify* do not have an encompassment condition.² This omission is possible since we only consider *finite* runs. Second, we allow variants of rules and equations to be added. This relaxation tremendously simplifies certificate generation in tools, where facts are renamed upon generation to avoid the maintenance and processing of many renamed versions of the same equation or rule. Also note that the *deduce* rule admits the addition of equations that originate from arbitrary peaks. In practice, tools usually limit its application to extended critical pairs.

The following two results establish that the rules resulting from a finite oKB run are oriented by the reduction order $>$ and that the induced equational theories before and after completion coincide.

Lemma 1 (✓). *If $(\mathcal{E}, \mathcal{R}) \vdash^* (\mathcal{E}', \mathcal{R}')$ then $\mathcal{R} \subseteq >$ implies $\mathcal{R}' \subseteq >$.* □

Lemma 2 (✓). *If $(\mathcal{E}, \mathcal{R}) \vdash^* (\mathcal{E}', \mathcal{R}')$ then $\leftrightarrow_{\mathcal{E} \cup \mathcal{R}}^* = \leftrightarrow_{\mathcal{E}' \cup \mathcal{R}'}^*$.* □

If the employed reduction order is \mathcal{F} -ground total then the above two results imply the following conversion equivalence involving *ordered rewriting* with respect to the final system.

Lemma 3 (✓). *Suppose $>$ is \mathcal{F} -ground total and $\mathcal{R} \subseteq >$. If $(\mathcal{E}, \mathcal{R}) \vdash^* (\mathcal{E}', \mathcal{R}')$ such that \mathcal{E}' , \mathcal{R}' , and $>$ are over the signature \mathcal{F} , then $\leftrightarrow_{\mathcal{E} \cup \mathcal{R}}^* = \leftrightarrow_{\mathcal{E}' \cup \mathcal{R}'}^*$ holds for conversions between terms in $\mathcal{T}(\mathcal{F})$.* □

This result is a key ingredient to our correctness results in Section 4. In order to apply it, however, we need ground total reduction orders. To this end, we formalized the following two results in *IsaFoR*.

Lemma 4 (✓). *If $>$ is a total precedence on \mathcal{F} then $>_{\text{kbo}}$ is \mathcal{F} -ground total.* □

Lemma 5 (✓). *If $>$ is a total precedence on \mathcal{F} then $>_{\text{lpo}}$ is \mathcal{F} -ground total.* □

In addition, we proved that for any given KBO $>_{\text{kbo}}$ (LPO $>_{\text{lpo}}$) defined over a total precedence $>$ there exists a minimal constant, that is, a constant c such that $t \geq_{\text{kbo}} c$ ($t \geq_{\text{lpo}} c$) holds for all $t \in \mathcal{T}(\mathcal{F})$ (which will be needed in Section 4). In earlier work by Becker et al. [4] ground totality of a lambda-free higher-order variant of KBO is formalized in *Isabelle/HOL*. However, for our purposes it makes sense to work with the definition of KBO that is already widely used in *IsaFoR*.

By Lemma 3, any two ground terms convertible in the initial equational theory are convertible with respect to ordered rewriting in the system obtained from an oKB run. The remaining key issue is to decide when the current ordered rewrite system is ground confluent, such that a tool implementing oKB can stop. Instead of defining a fairness criterion as done by Bachmair et al. [3], we use the following criterion for correctness involving ground joinability.

² The encompassment condition demands that if a rule or equation $\ell \approx r$ is used to rewrite a term $t = C[\ell\sigma]$ then C is non-empty or σ is not a renaming.

Lemma 6 (✓). *If for all equations $s \approx t$ in \mathcal{E} we have $s > t$ or $t \approx s$ in \mathcal{E} and $\text{CP}_{>}(\mathcal{E}) \subseteq \downarrow_{\mathcal{E}_{>}}^g$ then \mathcal{E} is ground confluent with respect to $>$. \square*

Note that the symmetry condition on \mathcal{E} above is just a convenient way to express the split of \mathcal{E} into rewrite rules with fixed orientation, and equations applicable in both directions, which allows us to treat an ordered rewrite system as a single set of equations. Lemmas 3 and 6 combine to the following correctness result.

Corollary 1 (✓). *If $>$ is \mathcal{F} -ground total and $(\mathcal{E}_0, \emptyset) \vdash^* (\mathcal{E}, \mathcal{R})$ such that \mathcal{E}' , \mathcal{R}' , and $>$ are over the signature \mathcal{F} and $\text{CP}_{>}(\mathcal{R} \cup \mathcal{E}^{\leftrightarrow}) \subseteq \downarrow_{\mathcal{R} \cup \mathcal{E}^{\leftrightarrow}}^g$, then $\mathcal{S} = \mathcal{R} \cup \mathcal{E}^{\leftrightarrow}$ is ground complete and $\leftrightarrow_{\mathcal{E}_0}^* = \leftrightarrow_{\mathcal{S}}^*$ holds for conversions between terms in $\mathcal{T}(\mathcal{F})$.*

Before we can apply this result in order to obtain ground completeness we need to be able to discharge its ground joinability assumption on extended critical pairs. This is the topic of the next section.

4 Formalized Ground Joinability Criteria

In general, ground joinability is undecidable even for terminating rewrite systems [7]. Below, we formalize two sufficient criteria.

4.1 A Simple Criterion

We start with the criterion that Bachmair et al. [3] proposed when they introduced ordered completion.

Lemma 7 (✓). *Suppose $>$ is a ground total reduction order over \mathcal{F} with a minimal constant. Then, $\mathcal{E}_{>}$ is \mathcal{F} -ground complete whenever for all $s \approx t \in \text{CP}_{>}(\mathcal{E}^{\leftrightarrow})$ it holds that $s \downarrow_{\mathcal{E}_{>}} t$, or $s \approx t = (s' \approx t')\sigma$ for some $s' \approx t' \in \mathcal{E}^{\leftrightarrow}$. \square*

A minimal constant c is needed to turn arbitrary ordered rewrite steps into ordered rewrite steps over $\mathcal{T}(\mathcal{F})$: when performing an ordered rewrite step using an equation $u \approx v$ with $V = \text{Var}(v) \setminus \text{Var}(u) \neq \emptyset$, a step over $\mathcal{T}(\mathcal{F})$ is obtained by instantiating all variables in V to c . We illustrate the criterion on an example.

Example 1. The following equational system \mathcal{E}_0 is derived by ConCon while checking infeasibility of a critical pair of the conditional rewrite system Cops #361:

$$\begin{array}{lll} x \div y \approx \langle 0, y \rangle & x \div y \approx \langle s(q), r \rangle & x - 0 \approx x \\ 0 - y \approx 0 & s(x) - s(y) \approx x - y & s(x) > s(y) \approx x > y \\ s(x) > 0 \approx \text{true} & s(x) \leq s(y) \approx x \leq y & 0 \leq x \approx \text{true} \end{array}$$

In an ordered completion run, MædMax transforms \mathcal{E}_0 into the following rules \mathcal{R} and equations \mathcal{E} :

$$\begin{array}{lll} x - 0 \rightarrow x & 0 - x \rightarrow 0 & s(x) - s(y) \rightarrow x - y \\ 0 \leq x \rightarrow \text{true} & s(x) \leq s(y) \rightarrow x \leq y & x \div y \rightarrow \langle 0, y \rangle \end{array}$$

$$\begin{array}{lll}
s(x) > 0 \rightarrow \text{true} & s(x) > s(y) \rightarrow x > y & \\
\langle s(x), y \rangle \approx \langle s(q), r \rangle & \langle 0, y \rangle \approx \langle s(q), r \rangle & \langle 0, x \rangle \approx \langle 0, y \rangle
\end{array}$$

Ground confluence of this system can be established by means of Lemma 7. For example, the extended overlap between the first two equations gives rise to the extended critical pair $\langle 0, y \rangle \approx \langle s(x), y \rangle$, which is just an instance of the second equation (and similarly for the other extended critical pairs).

4.2 Ground Joinability via Order Closures

The criterion discussed in Subsection 4.1 is rather weak. For instance, it cannot handle associativity and commutativity, as illustrated next [9, Example 1.1].

Example 2. Consider the system \mathcal{E} consisting of the three equations

$$(1) \quad (x * y) * z \approx x * (y * z) \quad (2) \quad x * y \approx y * x \quad (3) \quad x * (y * z) \approx y * (x * z)$$

and the reduction order $>_{\text{kbo}}$ with $w_0 = 1$ and $w(*) = 0$. The first equation can be oriented from left to right, whereas the other ones are unorientable.

We obtain the following extended critical peak from equations (2) and (1):

$$z * (x * y) \leftarrow (x * y) * z \rightarrow x * (y * z)$$

The resulting extended critical pair is neither an instance of an equation in \mathcal{E} nor joinable. Thus the criterion of Lemma 7 does not apply.

However, this extended critical pair is ground joinable, which we show in the following. The reduction order $>_{\text{kbo}}$ is contained in an \mathcal{F}' -ground total one on any extension of the signature $\mathcal{F}' \supseteq \mathcal{F}$ (using the well-order theorem and incrementality of KBO). Thus, for any grounding substitution σ the terms $x\sigma$, $y\sigma$, and $z\sigma$ are totally ordered. Suppose for instance that $x\sigma > z\sigma > y\sigma$. Then there is an ordered rewrite sequence witnessing joinability:

$$\begin{array}{ccc}
z\sigma * (x\sigma * y\sigma) & & x\sigma * (y\sigma * z\sigma) \\
\swarrow (2) & & \searrow (2) \\
z\sigma * (y\sigma * x\sigma) & & y\sigma * (x\sigma * z\sigma) \\
\searrow (3) & & \swarrow (3) \\
y\sigma * (z\sigma * x\sigma) & &
\end{array}$$

If, on the other hand, $x\sigma = y\sigma > z\sigma$ holds, there is a joining sequence as well:

$$\begin{array}{ccc}
& & x\sigma * (x\sigma * z\sigma) = x\sigma * (y\sigma * z\sigma) \\
& & \swarrow (2) \\
& & x\sigma * (z\sigma * x\sigma) \\
\swarrow (3) & & \\
z\sigma * (x\sigma * y\sigma) = z\sigma * (x\sigma * x\sigma) & &
\end{array}$$

By ensuring the existence of a joining sequence for all possible relationships between $x\sigma$, $y\sigma$, and $z\sigma$, ground joinability can be established. Using this approach to show that all extended critical pairs are ground joinable, it can be verified that \mathcal{E} is in fact ground complete.

The ground joinability test by Martin and Nipkow [9] is based on the idea illustrated in Example 2 above: perform a case analysis by considering ordered

rewriting using all extensions of $>$ to instantiations of variables. Below, we give the corresponding formal definitions used in `lsaFoR`. For any relation R on terms, let $\sigma(R)$ denote the relation such that $s\sigma\sigma(R)t\sigma$ holds if and only if sRt .

Definition 2 (✓). *A closure \mathcal{C} is a mapping between relations on terms that satisfies the following properties:*

- (1) *If $s\mathcal{C}(R)t$ then $s\sigma\mathcal{C}(\sigma(R))t\sigma$, for all relations R , substitutions σ , and terms s and t .*
- (2) *If $R\subseteq R'$ then $\mathcal{C}(R)\subseteq\mathcal{C}(R')$, for all relations on terms R and R' .*

The closure \mathcal{C} is compatible with a relation on terms R if $\mathcal{C}(R)\subseteq R$ holds.

In the remainder of this section we assume \mathcal{F} to be the signature of the input problem, we consider an \mathcal{F} -ground total reduction order $>$ as well as a closure \mathcal{C} that is compatible with $>$. Furthermore, we assume for every finite set of variables $V\subseteq\mathcal{V}$ and every equivalence relation \equiv on V a representation function rep_\equiv such that for any $x\in V$ we have $x\equiv\text{rep}_\equiv(x)$, $\text{rep}_\equiv(x)\in V$ and $x\equiv y$ implies $\text{rep}_\equiv(x)=\text{rep}_\equiv(y)$. Given an equivalence relation \equiv on V , let $\hat{\equiv}$ denote the substitution such that $\hat{\equiv}(x)=\text{rep}_\equiv(x)$ for all $x\in V$.

Definition 3 (✓). *Given an ES \mathcal{E} and a reduction order $>$, terms s and t are \mathcal{C} -joinable, written $s\downarrow_{\mathcal{E}}^{\mathcal{C}}t$, if for all equivalence relations \equiv on $\mathcal{V}\text{ar}(s,t)$ and every order \succ on the equivalence classes of \equiv it holds that*

$$s\hat{\equiv}\xrightarrow[\mathcal{E}_{\mathcal{C}(\succ)}]{*}\cdot\overset{\equiv}{\leftarrow}\xrightarrow{\mathcal{E}}\cdot\overset{*}{\leftarrow}\xrightarrow[\mathcal{E}_{\mathcal{C}(\succ)}]{*}t\hat{\equiv}\quad (1)$$

Example 3. For instance, consider the terms $s=z*(x*y)$ and $t=x*(y*z)$ from Example 2. One possible equivalence relation \equiv on $\mathcal{V}\text{ar}(s,t)=\{x,y,z\}$ is given by the equivalence classes $\{x,y\}$ and $\{z\}$; one possible order on these is $\hat{\equiv}(x)\succ\hat{\equiv}(z)$ (corresponding to the second example for an order on the instantiations $x\sigma$ and $z\sigma$ in Example 2). By taking \mathcal{C} to be the KBO closure (see Definition 5 below), we have $x*z\mathcal{C}(\succ)z*x$ and $x*(z*x)\mathcal{C}(\succ)z*(x*x)$. Using the ES \mathcal{E} from Example 2 we thus obtain the ordered rewrite sequence

$$t\hat{\equiv}=x*(x*z)\xrightarrow[\mathcal{E}_{\mathcal{C}(\succ)}]{}x*(z*x)\xrightarrow[\mathcal{E}_{\mathcal{C}(\succ)}]{}z*(x*x)=s\hat{\equiv}$$

Ground joinability follows from \mathcal{C} -joinability. Since this is the key result for the ground joinability criterion of this subsection, we also sketch its proof.

Lemma 8 (✓). *If $s\downarrow_{\mathcal{E}}^{\mathcal{C}}t$ then $s\downarrow_{\mathcal{E}}^{\mathcal{G}}t$.*

Proof. We assume $s\downarrow_{\mathcal{E}}^{\mathcal{C}}t$ and consider a grounding substitution σ to show $s\sigma\downarrow_{\mathcal{E}}t\sigma$. There is some equivalence relation \equiv on $\mathcal{V}\text{ar}(s,t)$ such that $x\equiv y$ holds if and only if $\sigma(x)=\sigma(y)$ for all $x,y\in\mathcal{V}\text{ar}(s,t)$. Note that this implies $s\sigma=s\hat{\equiv}\sigma$ and $t\sigma=t\hat{\equiv}\sigma$.

We can define an order \succ on the equivalence classes of \equiv such that $[x]_\equiv\succ[y]_\equiv$ if and only if $\sigma(x)\succ\sigma(y)$. Hence $\sigma(\succ)\subseteq>$ holds, and by Definition 2(2) we have $\mathcal{C}(\sigma(\succ))\subseteq\mathcal{C}(\succ)$. Compatibility implies $\mathcal{C}(\succ)\subseteq>$, and thus $\mathcal{C}(\sigma(\succ))\subseteq>$.

From Definition 2(1) we can show that $u \rightarrow_{\mathcal{E}_{\mathcal{C}(\succ)}} v$ implies $u\sigma \rightarrow_{\mathcal{E}_{\mathcal{C}(\sigma(\succ))}} v\sigma$ for all terms u and v . So using the assumption $s \downarrow_{\mathcal{E}}^{\mathcal{C}} t$ we can apply σ to a conversion of the form (1) to obtain

$$s\sigma = s \hat{=} \sigma \xrightarrow[\mathcal{E}_{\mathcal{C}(\sigma(\succ))}]{} \cdot \xrightarrow[\mathcal{E}]{} \cdot \xleftarrow[\mathcal{E}_{\mathcal{C}(\sigma(\succ))}]{} t \hat{=} \sigma = t\sigma \quad (2)$$

Ordered rewriting is monotone with respect to the order, and hence $\mathcal{C}(\sigma(\succ)) \subseteq \succ$ implies $\rightarrow_{\mathcal{E}_{\mathcal{C}(\sigma(\succ))}} \subseteq \rightarrow_{\mathcal{E}_{\succ}}$. Thus (2) implies the existence of a conversion

$$s\sigma \xrightarrow[\mathcal{E}_{\succ}]{} \cdot \xrightarrow[\mathcal{E}_{\succ}]{} \cdot \xleftarrow[\mathcal{E}_{\succ}]{} t\sigma$$

where the $\leftrightarrow_{\mathcal{E}_{\succ}}$ step exists as any two \mathcal{F} -ground terms are comparable in \succ . \square

Note that the proof above uses the monotonicity assumption for closures (Definition 2(2)), which is not present in [9]. The following counterexample illustrates that monotonicity is indeed necessary.

Example 4. Consider the ES $\mathcal{E} = \{f(x) \approx a\}$ and suppose that $\succ = \mathcal{C}(\succ)$ is an LPO with precedence $a > b > c > f$. Moreover, take $s = f(b)$ and $t = f(c)$. Any order \succ as in Definition 3 is empty since $\text{Var}(s, t) = \emptyset$. As \mathcal{C} is not required to be monotone, the relation $\mathcal{C}(\succ)$ may contain $(f(b), a)$ and $(f(c), a)$. Then $s \rightarrow_{\mathcal{E}_{\mathcal{C}(\succ)}} a$ and $t \rightarrow_{\mathcal{E}_{\mathcal{C}(\succ)}} a$ imply $s \downarrow_{\mathcal{E}}^{\mathcal{C}} t$ even though $s \downarrow_{\mathcal{E}_{\succ}}^{\mathcal{G}} t$ does not hold.

Below, we define an inductive predicate \mathbf{gj} which is used to conclude ground joinability of a given equation.

Definition 4 (✓). *Given an ES \mathcal{E} and a reduction order \succ , \mathbf{gj} is defined inductively by the following rules:*

delete		$\mathbf{gj}(t, t)$
closure	$s \downarrow_{\mathcal{E}}^{\mathcal{C}} t \implies$	$\mathbf{gj}(s, t)$
step	$s \leftrightarrow_{\mathcal{E}} t \implies$	$\mathbf{gj}(s, t)$
rewrite left	$s \xrightarrow[\mathcal{E}_{\succ}]{} u$ and $\mathbf{gj}(u, t) \implies$	$\mathbf{gj}(s, t)$
rewrite right	$t \xrightarrow[\mathcal{E}_{\succ}]{} u$ and $\mathbf{gj}(s, u) \implies$	$\mathbf{gj}(s, t)$
congruence	$\mathbf{gj}(s_i, t_i)$ for all $1 \leq i \leq n \implies$	$\mathbf{gj}(f(s_1, \dots, s_n), f(t_1, \dots, t_n))$

This test differs from the one due to Martin and Nipkow [9] by the two rewrite rules, which were added to allow for more efficient checks, as illustrated next.

Example 5. Consider the ES \mathcal{E}

$$f(x) \approx f(y) \qquad g(x, y) \approx f(x)$$

together with a KBO that can orient the second equation (for instance, one can take as precedence $g > f > c$ and let all function symbol weights as well as w_0

be 1). Then $\text{gj}(f(x), f(z))$ holds by the **step rule**, $\text{gj}(g(x, y), f(z))$ follows by an application of **rewrite left**, and $\text{gj}(g(x, y), g(z, w))$ by **rewrite right**. By Lemma 9 below it thus follows that the equation $g(x, y) \approx g(z, w)$ is ground joinable.

However, the criterion by Martin and Nipkow [9] lacks the rewrite steps. Hence ground joinability of $g(x, y) \approx g(z, w)$ can only be established by applying the closure rule. This amounts to checking ground joinability with respect to 81 relations between the four variables. Since the number of variable relations is in general exponential, the criterion stated in Definition 4 can in practice be exponentially more efficient than the test by Martin and Nipkow [9].

Using Lemma 8 it is not hard to show the following correctness results.

Lemma 9 (✓). *Suppose for all $s \approx t$ in \mathcal{E} we have $s > t$ or $t \approx s$ in \mathcal{E} . Then $\text{gj}(s, t)$ implies $s \downarrow_{\mathcal{E}}^g t$. □*

Lemma 10 (✓). *If for all $s \approx t$ in \mathcal{E} we have $s > t$ or $t \approx s$ in \mathcal{E} and $\text{CP}_{>}(\mathcal{E}) \subseteq \downarrow_{\mathcal{E}}^g$ then \mathcal{E} is ground confluent with respect to $>$. □*

This test can not only handle Example 2 but also the group theoretic problem from the introduction. Moreover, it subsumes Lemma 7 since whenever for some equation $s \approx t$ we have $s \downarrow_{\mathcal{E}}^g t$ by Lemma 7 then $\text{gj}(s, t)$ holds.

Closures for Knuth-Bendix Orders. Definition 2 requires abstract properties on closures. In the following we define closures for KBO as used in `IsaFoR/CeTA`.

Similar to the already existing definition of KBO in `IsaFoR` [16] we define the closure $>_{\text{kbo}}^R$ as follows.

Definition 5 (✓). *Let R be a relation on terms, $>$ a precedence on \mathcal{F} , and (w, w_0) a weight function. The KBO closure $>_{\text{kbo}}^R$ is a relation on terms inductively defined as follows: $s >_{\text{kbo}}^R t$ if $s R t$, or $|s|_x \geq |t|_x$ for all $x \in \mathcal{V}$ and either*

- (a) $w(s) > w(t)$, or
- (b) $w(s) = w(t)$ and one of
 - (1) $s \notin \mathcal{V}$ and $t \in \mathcal{V}$, or
 - (2) $s = f(s_1, \dots, s_n)$, $t = g(t_1, \dots, t_m)$ and $f > g$, or
 - (3) $s = f(s_1, \dots, s_n)$, $t = f(t_1, \dots, t_n)$ and there is some $i \leq n$ such that $s_j = t_j$ for all $1 \leq j < i$ and $s_i >_{\text{kbo}}^R t_i$

Note that even though Definition 5 resembles the usual definition of KBO, it defines a *closure* of a relation R in a KBO-like way rather than a reduction order. For instance, if $x \succ z$, as in Example 3, then $x * z >_{\text{kbo}}^> z * x$ holds.

We prove that $>_{\text{kbo}}^R$ is indeed a closure that is compatible with $>_{\text{kbo}}$ based on the same weight function and precedence.

Lemma 11. *Let R be a relation on terms, $>$ a precedence on \mathcal{F} , and (w, w_0) a weight function. Then all of the following hold:*

- (a) *If $s >_{\text{kbo}} t$ then $s >_{\text{kbo}}^R t$ for all terms s and t . ✓*
- (b) *If $R \subseteq R'$ then $>_{\text{kbo}}^R \subseteq >_{\text{kbo}}^{R'}$. ✓*
- (c) *If $s >_{\text{kbo}}^R t$ then $s\sigma >_{\text{kbo}}^{\sigma(R)} t\sigma$, for all substitutions σ , and terms s and t . ✓*
- (d) *The closure $>_{\text{kbo}}^R$ is compatible with $>_{\text{kbo}}$. ✓*

5 Applications

Ground complete rewrite systems can be used to decide equivalence of ground terms with respect to their induced equational theory. Here we highlight applications of this decision problem.

Deciding Ground Equations. Suppose we obtain the ordered rewrite system $(\mathcal{E}, \mathcal{R})$ and the reduction order $>$ by applying ordered completion to an initial set of equations \mathcal{E}_0 . Then it is easy to decide whether two ground terms s and t are equivalent with respect to \mathcal{E}_0 (that is, whether $s \leftrightarrow_{\mathcal{E}_0}^* t$): it suffices to check if the (necessarily unique) normal forms of s and t with respect to $\mathcal{R} \cup \mathcal{E}_>$ coincide. Also if all variables of a non-ground goal equation are universally quantified, the goal can be decided by substituting fresh constants for its variables.

Equations with Existential Variables. The following trick by Bachmair et al. [3] allows us to reduce equations with existentially quantified variables to the ground case: Let \mathcal{E} be a set of equations and $s \approx t$ a goal equation where all variables are existentially quantified. This corresponds to the question whether there is a substitution σ such that $s\sigma \leftrightarrow_{\mathcal{E}}^* t\sigma$ holds. We employ three fresh function symbols **eq**, **true**, and **false**, and define $\mathcal{E}_{s,t}^{\text{eq}}$ to denote \mathcal{E} extended by the two equations $\text{eq}(x, x) \approx \text{true}$ and $\text{eq}(s, t) \approx \text{false}$.

If a ground complete system equivalent to $\mathcal{E}_{s,t}^{\text{eq}}$ is found—for instance discovered by ordered completion—then it can be used to decide the goal, as stated next.

Lemma 12 (✓). *Let s, t , and \mathcal{E} all be over signature \mathcal{F} and let \mathcal{S} be a ground complete TRS such that $\leftrightarrow_{\mathcal{E}_{s,t}^{\text{eq}}}^* \subseteq \leftrightarrow_{\mathcal{S}}^*$ on $\mathcal{T}(\mathcal{F})$. If $s\sigma \leftrightarrow_{\mathcal{E}}^* t\sigma$ then $\text{true} \downarrow_{\mathcal{S}} = \text{false} \downarrow_{\mathcal{S}}$.*

Proof. Since $s\sigma \leftrightarrow_{\mathcal{E}}^* t\sigma$, there is a conversion $s\sigma \leftrightarrow_{\mathcal{E}_{s,t}^{\text{eq}}}^* t\sigma$ by construction of $\mathcal{E}_{s,t}^{\text{eq}}$. Moreover, (appealing to an earlier formalization about signature extensions [18]) there exists an \mathcal{F} -grounding substitution τ such that $s\tau \leftrightarrow_{\mathcal{E}_{s,t}^{\text{eq}}}^* t\tau$. So we have

$$\text{true} \xleftarrow[\mathcal{E}_{s,t}^{\text{eq}}]{} \text{eq}(s\tau, s\tau) \xleftarrow[\mathcal{E}_{s,t}^{\text{eq}}]{*} \text{eq}(s\tau, t\tau) \xrightarrow[\mathcal{E}_{s,t}^{\text{eq}}]{} \text{false}$$

and by the assumed conversion inclusion an \mathcal{S} -conversion between **true** and **false**. Several applications of ground confluence of \mathcal{S} yield joinability of $\text{true} \downarrow_{\mathcal{S}}$ and $\text{false} \downarrow_{\mathcal{S}}$. Since both of these terms are normal forms they coincide. \square

Infeasibility of Conditions. A decision procedure for ground equations can also be harnessed to prove infeasibility of conditions in conditional term rewriting. Here a condition c is a sequence of pairs of terms $s_1 \approx t_1, \dots, s_k \approx t_k$ and we say that c is infeasible whenever there is no substitution such that $s_i\sigma \rightarrow_{\mathcal{R}}^* t_i\sigma$ holds for all $1 \leq i \leq k$. Now, it is obviously a sound overapproximation to ensure that there is no σ such that $s_i\sigma \leftrightarrow_{\mathcal{R}}^* t_i\sigma$ for all $1 \leq i \leq k$. This suggests that completion methods might be applicable.

But there are still two complications before we are able to achieve an infeasibility check: (1) the rules of a conditional term rewrite system (CTRS for short) \mathcal{R} may be guarded by conditions, making \mathcal{R} an unsuitable input for ordered completion, and (2) the conditions c are most of the time not ground. As is conventional when adopting TRS methods to conditional rewriting, we solve (1) by dropping all conditions from the rules of \mathcal{R} , resulting in the unconditional TRS \mathcal{R}_u whose rewrite relation overapproximates the one of \mathcal{R} . Of course if we can establish that there is no σ such that $s_i\sigma \rightarrow_{\mathcal{R}_u}^* t_i\sigma$ for all $1 \leq i \leq k$, then we also obtain infeasibility of c with respect to the CTRS \mathcal{R} . In order to solve (2) we use a fresh function symbol c and apply Lemma 12 to decide the equation $s = c(s_1, \dots, s_k) \approx c(t_1, \dots, t_k) = t$ by applying ordered completion to $\mathcal{R}_{u, s, t}^{\text{eq}}$. If $s \not\rightarrow_{\mathcal{R}_{u, s, t}^{\text{eq}}}^* t$ we can conclude infeasibility of c .

Checking for infeasibility is for example useful when analyzing the confluence of a conditional rewrite system, since whenever we encounter a conditional critical pair whose conditions are infeasible, we can ignore it entirely. Since 2019 the Confluence Competition (CoCo)³ also features a dedicated infeasibility category.

6 Certification

In this section we describe the proof certificates for the different certifiable properties and summarize the corresponding Isabelle/HOL check functions.

Here, *check functions* are the formal connection between general, abstract results and concrete certificates. For example, a check function for a KBO termination proof takes a certificate, containing a concrete TRS, a specific precedence, and fixed weight functions, as input. It checks that the KBO instance is admissible and orients all rules of the TRS from left to right. By appealing to the abstract result that compatibility of a TRS with an admissible KBO implies termination, it then concludes termination of the concrete instance.

Only check functions that are both executable and proven sound are allowed in the certifier. The latter means that success of the check function implies a concrete instance of the corresponding general result (in our example success proves termination of the given TRS). In case of failure it is customary for CeTA check functions to give a human readable reason for why a certificate is rejected.

Ordered Completion Certificates. Here, the certificate consists of

- a set of initial equations \mathcal{E}_0 ,
- an ordered completion result $(\mathcal{E}, \mathcal{R})$ together with a reduction order $>$, and
- a sequence of inference steps according to Definition 1.

The corresponding check function verifies that (1) the inference steps form a valid run $(\mathcal{E}_0\pi, \emptyset) \vdash^* (\mathcal{E}, \mathcal{R})$ for some renaming π , (2) all extended critical pairs are joinable, by default according to Lemma 10, and (3) the reduction order is admissible, in case of KBO.

Next, we illustrate such an ordered completion proof by an example.

³ <http://project-coco.uibk.ac.at/2019/>

Example 6. The certificate corresponding to Example 1 contains the equations \mathcal{E}_0 , the resulting system $(\mathcal{E}, \mathcal{R})$, and the reduction order $>_{\text{kbo}}$ with precedence $> > \mathbf{s} > \leq > \mathbf{true} > - > \div > \langle \cdot, \cdot \rangle > 0$, $w_0 = 1$, and $w(0) = 2$, $w(\div) = w(\mathbf{true}) = w(\mathbf{s}) = 1$, and all other symbols having weight 0. In addition, a sequence of inference steps explains how $(\mathcal{E}, \mathcal{R})$ is obtained from \mathcal{E}_0 :

$$\begin{array}{ll}
\text{simplify}_{\text{left}} & x \div y \approx \langle \mathbf{s}(q), r \rangle \text{ to } \langle 0, y \rangle \approx \langle \mathbf{s}(q), r \rangle \\
\text{deduce} & \langle 0, x \rangle \leftarrow \langle \mathbf{s}(u), v \rangle \rightarrow \langle 0, y \rangle \\
\text{deduce} & \langle \mathbf{s}(x), y \rangle \leftarrow \langle 0, u \rangle \rightarrow \langle \mathbf{s}(q), r \rangle \\
\text{deduce} & x > y \leftarrow \mathbf{s}(x) > \mathbf{s}(y) \rightarrow \mathbf{s}(\mathbf{s}(x)) > \mathbf{s}(\mathbf{s}(y)) \\
\text{orient}_{\text{lr}} & 0 \leq x \rightarrow \mathbf{true} \\
\text{orient}_{\text{rl}} & \mathbf{s}(\mathbf{s}(x)) > \mathbf{s}(\mathbf{s}(y)) \rightarrow x > y \tag{*} \\
\dots & \\
\text{orient}_{\text{lr}} & \mathbf{s}(x) - \mathbf{s}(y) \rightarrow x - y \\
\text{orient}_{\text{lr}} & 0 - x \rightarrow 0 \\
\text{orient}_{\text{lr}} & \mathbf{s}(x) \leq \mathbf{s}(y) \rightarrow x \leq y \\
\text{collapse} & \mathbf{s}(\mathbf{s}(x)) > \mathbf{s}(\mathbf{s}(y)) \rightarrow x > y \text{ to } x > y \approx x > y \\
\text{collapse} & \mathbf{s}(\mathbf{s}(x)) > \mathbf{s}(0) \rightarrow \mathbf{true} \text{ to } \mathbf{s}(x) > 0 \approx \mathbf{true} \\
\text{simplify}_{\text{left}} & \mathbf{s}(x) > 0 \approx \mathbf{true} \text{ to } \mathbf{true} \approx \mathbf{true} \\
\text{delete} & x > y \approx x > y \\
\text{delete} & \mathbf{true} \approx \mathbf{true}
\end{array}$$

The first **collapse** step using rule (*) above illustrates our relaxed inference rule, it would not have been possible according to the original inference system [3] due to the encompassment condition since $\mathbf{s}(\mathbf{s}(x)) > \mathbf{s}(\mathbf{s}(y)) \not\approx \mathbf{s}(\mathbf{s}(x)) > \mathbf{s}(y)$.

We briefly comment on the differences to the certification of standard Knuth-Bendix completion as already present in CeTA [16]. For standard completion, the certificate contains the initial set of equations \mathcal{E}_0 , the resulting TRS \mathcal{R} together with a termination proof, and stepwise \mathcal{E}_0 -conversions from ℓ to r for each rule $\ell \rightarrow r \in \mathcal{R}$. The certifier first checks the termination proof to guarantee termination of \mathcal{R} . Then, confluence of \mathcal{R} can be guaranteed by ensuring that all critical pairs are joinable. At this point it is easy to verify the inclusion $\leftrightarrow_{\mathcal{E}_0}^* \subseteq \leftrightarrow_{\mathcal{R}}^*$: for each equation $s \approx t \in \mathcal{E}_0$ the \mathcal{R} -normal forms of s and t are computed and checked for syntactic equality. The converse inclusion $\leftrightarrow_{\mathcal{R}}^* \subseteq \leftrightarrow_{\mathcal{E}_0}^*$ is taken care of by the provided \mathcal{E}_0 -conversions. Overall, we obtain that \mathcal{R} is a complete presentation of \mathcal{E}_0 without mentioning a specific inference system.

Unfortunately, the same approach does not work for ordered completion: The inclusion $\leftrightarrow_{\mathcal{E}_0}^* \subseteq \leftrightarrow_{\mathcal{R} \cup \mathcal{E}_>}^*$ cannot be established by rewriting equations in \mathcal{E}_0 to normal form, since they may contain variables but $\mathcal{R} \cup \mathcal{E}_>$ is only ground confluent. Moreover, since ground joinability is undecidable no complete check can be performed. Therefore, we instead ask for certificates that contain explicit inference steps, as described above.

Equational Satisfiability Certificates. We use the term “satisfiability” of unit equality problems in line with the terminology of TPTP [22]: given a set of equations \mathcal{E}_0 and a ground goal *inequality* $s \not\approx t$, show that this axiomatization

is satisfiable. To this end, completion-based tools try to find a ground complete presentation \mathcal{S} of \mathcal{E}_0 and verify that $s \downarrow_{\mathcal{S}} \neq t \downarrow_{\mathcal{S}}$.

A certificate for this application extends an ordered completion certificate by the goal terms. The corresponding check function verifies that

- the presented ordered completion proof is valid as described above,
- the goal inequality is ground,
- the signature of \mathcal{E}_0 , \mathcal{E} , and \mathcal{R} is included in the signature of $>$, and
- the terms in the goal have different normal forms.

We chose the symbols mentioned by the reduction order to be the considered signature \mathcal{F} . In comparison to picking the signature of \mathcal{E}_0 , this has the advantage that it is easy to add additional function symbols. Moreover, since KBO precedences in the CPF input are lists of function symbols, no additional checks are required to ensure \mathcal{F} -ground totality of the constructed reduction order.

As a side note, unsatisfiability proofs are much easier to certify: a tool only needs to output a conversion between the two goal terms. Support for the corresponding certificates has already been added to **CeTA** earlier [21].

Infeasibility Certificates. Actually we check (generalized) nonreachability [15] of a target t from a source s with respect to a TRS \mathcal{R} , that is, the property that, given a TRS \mathcal{R} and two terms s and t , there is no substitution σ with $s\sigma \rightarrow_{\mathcal{R}}^* t\sigma$.

The corresponding certificates list function symbols **eq**, **true**, and **false**, together with an equational satisfiability certificate. The check function first constructs, using **eq**, **true**, and **false** from the certificate the TRS $\mathcal{R}_{s,t}^{\text{eq}}$ and then verifies that the equation **true** \approx **false** is not satisfiable according to the supplied equational satisfiability certificate with $\mathcal{R}_{s,t}^{\text{eq}}$ as initial set of equations.

7 Experiments

Below we summarize experiments with our certifier on different problem sets. More details are available from the accompanying website.⁴

Ordered Completion. Martin and Nipkow [9] give 10 examples. The criterion of Lemma 10 with KBO applies to 7 of those and **MædMax** produces corresponding proofs. Six of these proofs are certified by **CeTA**. The missing example uses a trick also used by Waldmeister [1]: certain *redundant* equations need not be considered for critical pair computation. This simplification is not yet supported by **CeTA**.

We also ran **MædMax** on the 138 problems [13] for standard completion collected from the literature. Using KBO, **MædMax** can complete 55 of them, and 52 of those are certified. (Using LPO and KBO, 91 are completed.) For the three remaining (AC) group examples, **MædMax** uses a stronger criterion [23] which is currently not supported by **CeTA**. Overall, this amounts to 58% certification coverage of all ordered completion proofs by **MædMax**.

⁴ <http://cl-informatik.uibk.ac.at/experiments/okb/>

Satisfiable Unit Equality Problems. There are 144 unit equality problems (UEQ) in the TPTP 7.2.0 [22] benchmark that are classified as satisfiable, of which `MædMax` using KBO only can prove 11. All these proofs are certified by `CeTA`. With its general strategy `MædMax` can handle 14 problems, but two of those require duplicating rules, such that KBO is not applicable, and one has multiple goals, which is currently not supported by `CeTA`.

Infeasibility Problems. There are 148 oriented CTRSs in version 807 of the `Cops`⁵ benchmark (that is, the version of `Cops` where the highest problem number is 807) of `CoCo`. Here *oriented* means that a condition $s \approx t$ is satisfied by a substitution σ , whenever $s\sigma \rightarrow_{\mathcal{R}}^* t\sigma$. (This is the class of systems `ConCon` is specialized to, hence we restrict our experiments to the above 148 systems.)

Out of those 148 CTRSs, the previous version of `ConCon` (1.7) can prove (non)confluence of 109 with and of 112 without certification. The new version of `ConCon` (1.8), extended by infeasibility checks via ordered completion with `MædMax`, can handle 111 CTRSs with and 114 without certification. We thus obtain two new certified proofs, namely for `Cops` #340 and #361.

8 Conclusion

We presented our Isabelle/HOL formalization of ordered completion and two accompanying ground joinability criteria—now part of `IsaFoR` 2.37. It comes with check functions for ordered completion proofs, equational satisfiability proofs, and infeasibility proofs for conditional term rewriting. Formalizing soundness of these check functions allowed us to add support for corresponding certificates to the certifier `CeTA` that is code generated from `IsaFoR`. To the best of our knowledge, `CeTA` constitutes the first proof checker for ordered completion proofs. Indeed, it already helped us to detect a soundness error in `MædMax`, where in certain corner cases some extended critical pairs were ignored. Our experiments show that we can certify 58% of ordered completion proofs (corresponding to 94% of the KBO proofs) and 85% of the satisfiability proofs produced by `MædMax` (100% for KBO). The number of certified proofs of `ConCon` increased by two.

Moreover, `CeTA` is the only certifier used in the Confluence Competition; by certifying infeasibility proofs our work thus helps to validate more tool output. Regarding the recent `CoCo` 2019, certification currently covers roughly 83% of the benchmarks in the two categories (`CTRS` and `TRS`) that have certified counterparts (`CPF-CTRS` and `CPF-TRS`).

In the future, we plan to add support for closures of LPO and extend our certifier to verify proofs of pure, not necessarily unit, equality formulas, as well as ground confluence proofs by tools participating in the confluence competition.

Acknowledgments. We thank the anonymous referees for their constructive comments and various suggestions for improvements.

⁵ <http://cops.uibk.ac.at?q=1..807>

References

1. Avenhaus, J., Hillenbrand, T., Löchner, B.: On using ground joinable equations in equational theorem proving. *J. Symb. Comput.* **36**(1–2), 217–233 (2003). [https://doi.org/10.1016/S0747-7171\(03\)00024-5](https://doi.org/10.1016/S0747-7171(03)00024-5)
2. Baader, F., Nipkow, T.: *Term Rewriting and All That*. Cambridge University Press (1998). <https://doi.org/10.1017/CBO9781139172752>
3. Bachmair, L., Dershowitz, N., Plaisted, D.A.: Completion without failure. In: Kaci, H.A., Nivat, M. (eds.) *Resolution of Equations in Algebraic Structures, Rewriting Techniques*, vol. 2, pp. 1–30. Academic Press (1989). <https://doi.org/10.1016/B978-0-12-046371-8.50007-9>
4. Becker, H., Blanchette, J.C., Waldmann, U., Wand, D.: A transfinite Knuth-Bendix order for lambda-free higher-order terms. In: *Proc. 26th CADE*. LNCS, vol. 10395, pp. 432–453. Springer (2017). https://doi.org/10.1007/978-3-319-63046-5_27
5. Haftmann, F., Nipkow, T.: Code generation via higher-order rewrite systems. In: *Proc. 10th FLOPS*. LNCS, vol. 6009, pp. 103–117. Springer (2010). https://doi.org/10.1007/978-3-642-12251-4_9
6. Hirokawa, N., Middeldorp, A., Sternagel, C., Winkler, S.: Infinite runs in abstract completion. In: *Proc. 2nd FSCD*. LIPIcs, vol. 84, pp. 19:1–19:16 (2017). <https://doi.org/10.4230/LIPIcs.FSCD.2017.19>
7. Kapur, D., Narendran, P., Otto, F.: On ground-confluence of term rewriting systems. *Inform. Comput.* **86**(1), 14–31 (1990). [https://doi.org/10.1016/0890-5401\(90\)90023-B](https://doi.org/10.1016/0890-5401(90)90023-B)
8. Knuth, D.E., Bendix, P.: Simple word problems in universal algebras. In: Leech, J. (ed.) *Computational Problems in Abstract Algebra*, pp. 263–297. Pergamon Press (1970). <https://doi.org/10.1016/B978-0-08-012975-4>
9. Martin, U., Nipkow, T.: Ordered Rewriting and Confluence. In: *Proc. 10th CADE*. LNCS, vol. 449, pp. 366–380 (1990). https://doi.org/10.1007/3-540-52885-7_100
10. McCune, W.: Solution of the Robbins problem. *J. Autom. Reasoning* **19**(3), 263–276 (1997). <https://doi.org/10.1023/A:1005843212881>
11. McCune, W., Veroff, R., Fitelson, B., Harris, K., Feist, A., Wos, L.: Short single axioms for Boolean algebra. *J. Autom. Reasoning* **29**(1), 1–16 (2002). <https://doi.org/10.1023/A:1020542009983>
12. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL – A Proof Assistant for Higher-Order Logic, LNCS, vol. 2283. Springer (2002). <https://doi.org/10.1007/3-540-45949-9>
13. Sato, H., Winkler, S.: Encoding dependency pair techniques and control strategies for maximal completion. In: *Proc. 25th CADE*. LNCS, vol. 9195, pp. 152–162 (2015). https://doi.org/10.1007/978-3-319-21401-6_10
14. Schultz, P., Wisnesky, R.: Algebraic data integration. *J. Funct. Program.* **27**(e24), 51 pages (2017). <https://doi.org/10.1017/S0956796817000168>
15. Sternagel, C., Sternagel, T.: Certifying confluence of almost orthogonal CTRSs via exact tree automata completion. In: *Proc. 1st FSCD*. LIPIcs, vol. 52, pp. 29:1–29:16. Schloss Dagstuhl (2016). <https://doi.org/10.4230/LIPIcs.FSCD.2016.29>
16. Sternagel, C., Thiemann, R.: Formalizing Knuth-Bendix orders and Knuth-Bendix completion. In: *Proc. 24th RTA*. LIPIcs, vol. 21, pp. 287–302. Schloss Dagstuhl (2013). <https://doi.org/10.4230/LIPIcs.RTA.2013.287>
17. Sternagel, C., Thiemann, R.: The certification problem format. In: *Proc. 11th UITP*. EPTCS, vol. 167, pp. 61–72 (2014). <https://doi.org/10.4204/EPTCS.167.8>

18. Sternagel, C., Thiemann, R.: Signature extensions preserve termination. In: Proc. 19th CSL. LNCS, vol. 6247, pp. 514–528. Springer (2010). https://doi.org/10.1007/978-3-642-15205-4_39
19. Sternagel, C., Winkler, S.: Certified ordered completion. In: Proc. 7th IWC (2018), [arXiv:1805.10090](https://arxiv.org/abs/1805.10090)
20. Sternagel, T., Middeldorp, A.: Conditional confluence (system description). In: Proc. RTA/TLCA 2014. LNCS, vol. 8560, pp. 456–465 (2014). https://doi.org/10.1007/978-3-319-08918-8_31
21. Sternagel, T., Winkler, S., Zankl, H.: Recording completion for certificates in equational reasoning. In: Proc. 4th CPP. pp. 41–47 (2015). <https://doi.org/10.1145/2676724.2693171>
22. Sutcliffe, G.: The TPTP Problem Library and Associated Infrastructure: The FOF and CNF Parts. *J. Autom. Reasoning* **43**(4), 337–362 (2009). <https://doi.org/10.1007/s10817-009-9143-8>
23. Winkler, S.: A ground joinability criterion for ordered completion. In: Proc. 6th IWC. pp. 45–49 (2017)
24. Winkler, S., Moser, G.: **MædMax**: A maximal ordered completion tool. In: Proc. 9th IJCAR. LNCS, vol. 10900, pp. 472–480 (2018). https://doi.org/10.1007/978-3-319-94205-6_31