MædMax: A Maximal Ordered Completion Tool

Sarah Winkler(✉) and Georg Moser

University of Innsbruck, Innsbruck, Austria
{sarah.winkler,georg.moser}@uibk.ac.at

Abstract. The equational reasoning tool MædMax implements maximal ordered completion. This new approach extends the maxSMT-based method for standard completion developed by Klein and Hirokawa (2011) to ordered completion and equational theorem proving. MædMax incorporates powerful ground completeness checks and supports certification of its proofs by an Isabelle-based certifier. It also provides an order generation mode which can be used to synthesize term orderings for other tools. Experiments show the potential of our approach.

1 Introduction

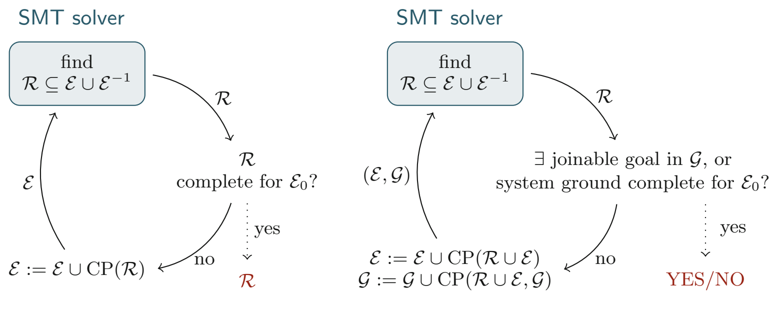
Equational reasoning has been one of the main research areas of theorem proving endeavors ever since Knuth and Bendix proposed completion [8]. To remedy the fact that completion may fail if unorientable equations are encountered, *ordered* completion was developed [3]. The ideas of this method have since been pervasive in automated deduction whenever equations are involved. Completion and paramodulation procedures are typically based on a given-clause-algorithm [9,14], which implies that facts are processed one at a time. The reduction order—a notoriously critical parameter—is typically fixed once and for all.

Maximal completion follows a very different approach. The idea is to maintain a single pool of equations. One then tries to orient as many equations as possible, by solving a maxSMT problem. If a terminating rewrite system $\mathcal{R}$ obtained in this way joins all its critical pairs as well as the input equalities, it is complete. Otherwise the critical pairs of $\mathcal{R}$ are added to $\mathcal{E}$ and the procedure is reiterated, as sketched in Fig. 1(a). In this way the proof search is guided by a maxSMT solver and steered towards systems with desirable properties. Maximal completion gave rise to the simple yet efficient and powerful completion tool Maxcomp [7]. It was later shown that the tool's search process can be significantly improved by using more complex objective functions, instead of merely maximizing the number of oriented equations [11].

The tool MædMax is an ordered completion and equational theorem proving tool based on a similar approach. As input it takes a set of equalities $\mathcal{E}_0$ and a

S. Winkler—Supported by FWF project T789.

D. Galmiche et al. (Eds.): IJCAR 2018, LNAI 10900, pp. 472–480, 2018.
https://doi.org/10.1007/978-3-319-94205-6_31



(a) standard completion (b) theorem proving

Fig. 1. Maximal completion.

goal equality, and tries to decide whether the goal follows from $\mathcal{E}_0$. To this end, it attempts to derive the goal from $\mathcal{E}_0$ or generate an equivalent ground-complete system. It is known that such a system can in particular be used to disprove the goal [1].

Figure 1(b) visualizes our approach (for the common case of a goal without variables). We maintain a set of equalities $\mathcal{E}$ and a set of goals $\mathcal{G}$, which is considered a disjunction. By orienting equations in $\mathcal{E}$ we find a terminating rewrite system $\mathcal{R}$. If a goal in $\mathcal{G}$ can be joined using $\mathcal{R}$, or the system is ground complete then the goal can be decided. Otherwise, critical pairs are added to both $\mathcal{E}$ and $\mathcal{G}$ and the procedure is repeated. Thus in contrast to the given clause algorithm many equations are processed at once, and the proof search is steered towards systems that have desirable properties.

Our experiments show that MædMax is particularly suited to prove (ground) completeness and satisfiability, due to sophisticated joinability criteria. If a proof is found then MædMax can output an equational proof that is checkable by the Isabelle-based verifier CeTA, thus offering a high degree of reliability. The tool also provides an order generation mode, where the reduction order deemed most suitable according to the optimized criteria is displayed. Finally, we illustrate practical relevance by means of examples in recent applications to data integration [12].

The remainder of this paper is organized as follows. In Sect. 2 we recall some relevant concepts and notations. Ordered maximal completion is presented in Sect. 3. Implementation details are highlighted in Sect. 4. In Sect. 5 we report on experimental results and conclude.

2 Preliminaries

In the sequel standard notation from term rewriting is used [2]. We consider the set of terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$ over a signature $\mathcal{F}$ and a set of variables $\mathcal{V}$, while $\mathcal{T}(\mathcal{F})$ denotes the set of all ground terms. An equational system (ES) $\mathcal{E}$ is a set of equations $\ell \approx r$ over $\mathcal{T}(\mathcal{F}, \mathcal{V})$, and a term rewrite system (TRS) $\mathcal{R}$ is a set of equations denoted as $\ell \to r$, where $\ell \notin \mathcal{V}$ and $\mathcal{V}\mathrm{ar}(r) \subseteq \mathcal{V}\mathrm{ar}(\ell)$. For an ES $\mathcal{E}$ we write $\mathcal{E}^{\pm}$ to denote $\mathcal{E} \cup \{t \approx s \mid s \approx t \in \mathcal{E}\}$. A TRS $\mathcal{R}$ is terminating if there are no infinite rewrite sequences $t_0 \to_{\mathcal{R}} t_1 \to_{\mathcal{R}} t_2 \to_{\mathcal{R}} \dots$, and (ground) confluent if $s \;{}_{\mathcal{R}}^{*}\!\!\leftarrow \cdot \to_{\mathcal{R}}^{*} t$ implies $s \to_{\mathcal{R}}^{*} \cdot {}_{\mathcal{R}}^{*}\!\!\leftarrow t$ for all (ground) terms s and t. A TRS which is terminating and (ground) confluent is (ground) complete.

One common method to establish termination of a TRS $\mathcal{R}$ is to find a reduction order $>$ which is compatible with $\mathcal{R}$. A reduction order $>$ is ground total if $s > t$, $t > s$, or $s = t$ holds for all ground terms s and t. It is known that LPO and KBO are reduction orders enjoying this property whenever they are based on a total precedence, and polynomial interpretations can be extended to such an order. For a reduction order $>$ and an ES $\mathcal{E}$, the TRS $\mathcal{E}^{>}$ consists of all rules $s\sigma \to t\sigma$ such that $s \approx t \in \mathcal{E}^{\pm}$ and $s\sigma > t\sigma$ [3]. A TRS $\mathcal{R}$ is moreover ground complete for an ES $\mathcal{E}_0$ if $\mathcal{R}$ is ground complete and the relations $\leftrightarrow_{\mathcal{R}}^{*}$ and $\leftrightarrow_{\mathcal{E}_0}^{*}$ coincide when restricted to ground terms. Given a terminating TRS $\mathcal{R}$ and a term t, we write $t{\downarrow_{\mathcal{R}}}$ to denote some normal form of t. For an ES $\mathcal{E}$, we write $\mathcal{E}{\downarrow_{\mathcal{R}}}$ for the set of all equations $s{\downarrow_{\mathcal{R}}} \approx t{\downarrow_{\mathcal{R}}}$ such that $s \approx t \in \mathcal{E}$ and $s{\downarrow_{\mathcal{R}}} \neq t{\downarrow_{\mathcal{R}}}$. An equation $s \approx t$ is ground joinable in $\mathcal{R}$ if $s\sigma \downarrow_{\mathcal{R}} t\sigma$ for all grounding substitutions σ, where $\downarrow_{\mathcal{R}}$ abbreviates $\to_{\mathcal{R}}^{*} \cdot {}_{\mathcal{R}}^{*}\!\!\leftarrow$.

We use the following notion of *extended critical pairs* [3]: Given a reduction order $>$ and $\ell_1 \approx r_1$ and $\ell_2 \approx r_2$ in $\mathcal{E}^{\pm}$, the equation $\ell_2\sigma[r_1\sigma]_p \approx r_2\sigma$ is an extended critical pair if p is a function symbol position in ℓ_2, the terms $\ell_2|_p$ and ℓ_1 are unifiable with most general unifier σ, and neither $r_1\sigma > \ell_1\sigma$ nor $r_2\sigma > \ell_2\sigma$ hold. The set of extended critical pairs of an ES $\mathcal{E}$ with respect to $>$ is denoted by $\mathrm{CP}_{>}(\mathcal{E})$.

3 Maximal Ordered Completion

We now formalize the approach of maximal ordered completion and theorem proving sketched in Fig. 1(b).

Let $\mathfrak{R}$ be a function mapping an ES $\mathcal{E}$ to a set of terminating TRSs such that for all $\mathcal{R} \in \mathfrak{R}(\mathcal{E})$ we have (1) $\mathcal{R} \subseteq \mathcal{E}^{\pm}$ and (2) there is a ground total reduction order $>$ extending $\to_{\mathcal{R}}$. Moreover, let S be a function from ESs to ESs such that $S(\mathcal{E}) \subseteq \leftrightarrow_{\mathcal{E}}^{*}$ for every ES $\mathcal{E}$. We consider define maximal ordered completion without goals. Our procedure is defined via the following relation φ which maps an ES $\mathcal{E}$ to a tuple $(\mathcal{R}, \mathcal{E}', >)$ consisting of a TRS $\mathcal{R}$, an ES $\mathcal{E}'$, and a reduction order $>$.

Definition 1. *Given a set of input equalities $\mathcal{E}_0$ and an ES $\mathcal{E}$, let*

$$\varphi(\mathcal{E}) = \begin{cases} (\mathcal{R}, \mathcal{E}{\downarrow_{\mathcal{R}}}, >) & \text{if } \mathcal{R} \cup (\mathcal{E}{\downarrow_{\mathcal{R}}})^{>} \text{ is ground complete for } \mathcal{E}_0 \\ & \text{for some } \mathcal{R} \in \mathfrak{R}(\mathcal{E}), \text{ and} \\ \varphi(\mathcal{E} \cup S(\mathcal{E})) & \text{otherwise.} \end{cases}$$

The idea is to recursively apply Definition 1 to a set of initial equations $\mathcal{E}_0$. Note that in general φ may be neither defined nor unique. In MædMax the set $S(\mathcal{E})$ is chosen such that $S(\mathcal{E}) \subseteq \bigcup_{\mathcal{R} \in \mathfrak{R}(\mathcal{E})} \mathrm{CP}_{>}(\mathcal{R} \cup \mathcal{E}{\downarrow_{\mathcal{R}}}){\downarrow_{\mathcal{R}}}$.

Example 1. Consider the following ES $\mathcal{E}_0$ axiomatizing a Boolean ring, where multiplication is denoted by concatenation.

(1)	$(x+y)+z \approx x+(y+z)$	(2)	$x+y \approx y+x$	(3)	$\mathsf{0}+x \approx x$
(4)	$x(y+z) \approx xy+xz$	(5)	$(xy)z \approx x(yz)$	(6)	$xy \approx yx$
(7)	$(x+y)z \approx xz+yz$	(8)	$xx \approx x$	(9)	$x+x \approx \mathsf{0}$
(10)	$\mathsf{1}x \approx x$				

Let $\mathcal{R}_1$ be the TRS $\{(1),(3),(\overline{4}),(5),(\overline{7}),(8),(9),(10)\}$ obtained by orienting distributivity from right to left and all other equations (except for commutativity) from left to right, and $\mathfrak{R}(\mathcal{E}_0)$ the singleton set containing $\mathcal{R}_1$. This choice orients a maximal number of equations. Now the set $S(\mathcal{E}_0)$ may consist of the following extended critical pairs of rules among $\mathcal{R}_1$ and the unorientable commutativity equations:

(11)	$x+(y+z) \approx y+(x+z)$	(12)	$x(yz) \approx y(xz)$	(13)	$x+\mathsf{0} \approx x$
(14)	$y+(x+y) \approx x$	(15)	$x(yx) \approx xy$	(16)	$x\mathsf{1} \approx x$
(17)	$y+(y+x) \approx x$	(18)	$x(xy) \approx xy$	(19)	$\mathsf{0}x \approx \mathsf{0}$

Note that $\mathcal{R}_1$-joinable critical pairs such as $x+(x+\mathsf{0}) \approx \mathsf{0}$ or $x\mathsf{0} \approx y\mathsf{0}$ are not included. We have $\varphi(\mathcal{E}_0) = \varphi(\mathcal{E}_1)$ for $\mathcal{E}_1 = \mathcal{E}_0 \cup S(\mathcal{E}_0)$. Now $\mathfrak{R}(\mathcal{E}_1)$ may contain $\mathcal{R}_2 = \{(1),(3),(4),(5),(7),\dots,(10),(13),\dots,(19)\}$. This TRS is LPO-terminating, so there is a ground-total reduction order $>$ that contains $\rightarrow_{\mathcal{R}_2}$. We have $\mathcal{E}_1{\downarrow}_{\mathcal{R}_2} = \{(2),(6),(11),(12)\}$, and it can be shown that for $\mathcal{E} = \mathcal{E}_1{\downarrow}_{\mathcal{R}_2}$ the system $\mathcal{R}_2 \cup \mathcal{E}^>$ is ground complete. Despite its simplicity, neither WM [1] nor E [14] or Vampire [9] can show satisfiability of this example (considering $\mathcal{E}_0$ as a set of axioms in first-order logic with equality, in the case of the latter).

We next extend our approach to theorem proving, akin to Fig. 1(b). Let S_G be a binary function on ESs such that $S_G(\mathcal{G},\mathcal{E}) \subseteq \leftrightarrow^*_{\mathcal{E}\cup\mathcal{G}} \setminus \leftrightarrow^*_{\mathcal{E}}$ for all ESs $\mathcal{E}$ and $\mathcal{G}$. In our implementation, $S_G(\mathcal{G},\mathcal{E})$ contains extended critical pairs between an equation in $\mathcal{G}$ and an equation in $\mathcal{E}$. The following relation ψ maps a pair of ESs $\mathcal{E}$ and $\mathcal{G}$ to YES or NO.

Definition 2. *Given a set of input equalities $\mathcal{E}_0$, an initial ground goal $s_0 \approx t_0$ and ESs $\mathcal{E}$ and $\mathcal{G}$, let*

$$\psi(\mathcal{E},\mathcal{G}) = \begin{cases} \text{YES} & \text{if } s \downarrow_{\mathcal{R}\cup\mathcal{E}^>} t \text{ for some } s \approx t \in \mathcal{G} \text{ and } \mathcal{R} \in \mathfrak{R}(\mathcal{E}), \\ \text{NO} & \text{if } \mathcal{R} \cup \mathcal{E}{\downarrow}_{\mathcal{R}}{}^> \text{ is ground complete for } \mathcal{E}_0 \\ & \text{but } s_0 \not\downarrow_{\mathcal{R}\cup\mathcal{E}^>} t_0, \text{ for some } \mathcal{R} \in \mathfrak{R}(\mathcal{E}), \text{ and} \\ \psi(\mathcal{E}',\mathcal{G}') & \text{for } \mathcal{G}' = S_G(\mathcal{G},\mathcal{R}\cup\mathcal{E}) \text{ and } \mathcal{E}' = S(\mathcal{E}). \end{cases}$$

For a set of input equations $\mathcal{E}_0$ and an initial goal $s_0 \approx t_0$, the procedure is started with the initial call $\psi(\mathcal{E}_0, \{s_0 \approx t_0\})$. Note that the parameter $\mathcal{G}$ of ψ denotes a disjunction of goals, not a conjunction. Due to the declarative nature of the completion and theorem proving procedures described by Definitions 1 and 2 the following correctness result is straightforward.

Theorem 1. *Let $\mathcal{E}_0$ be an ES and $s_0 \approx t_0$ be a ground goal.*

1. *If $\varphi(\mathcal{E}_0) = (\mathcal{R}, \mathcal{E}, >)$ then $\mathcal{R} \cup \mathcal{E}^>$ is ground complete for $\mathcal{E}_0$.*
2. *If $\psi(\mathcal{E}_0, \{s_0 \approx t_0\})$ is defined then $\psi(\mathcal{E}_0, \{s_0 \approx t_0\}) =$ YES if and only if $s_0 \leftrightarrow^*_{\mathcal{E}_0} t_0$.* □

4 Implementation

MædMax is available as a command-line tool and via a web interface.[1] It is implemented in OCaml and accepts input problems in the TPTP [17] as well as the trs format.[2] We describe how the three main phases of our approach are implemented (see Definitions 1 and 2): (1) finding the terminating TRSs $\mathfrak{R}(\mathcal{E})$, (2) success checks, and (3) selection of new equations and goals, i.e., computation of $S(\mathcal{E})$ and $S_G(\mathcal{G}, \mathcal{E})$. Also some further particular features of the tool get highlighted. Many settings can be controlled via a command line option, we refer to the website for details. In the default *auto* mode the settings are determined heuristically.

Finding Rewrite Systems. In phase (1), MædMax computes the set of TRSs $\mathfrak{R}(\mathcal{E})$ for a given ES $\mathcal{E}$ by solving an optimization problem whose objective function can be controlled via a strategy. Assuming we want to find a TRS $\mathcal{R} \in \mathfrak{R}(\mathcal{E})$ the search may involve the following criteria, as well as their (possibly weighted) sums:

(a) maximize the oriented equations in $\mathcal{E}$ (i.e., the size of $\mathcal{R}$),
(b) maximize the equations in $\mathcal{E}$ that are reducible by $\mathcal{R}$,
(c) minimize critical pairs among rules in $\mathcal{R}$, or
(d) maximize reducible critical pairs among rules in $\mathcal{R}$.

Maxcomp relied on criterion (a), and later a combination of (a), (b), and (c) was found most suitable for standard completion [11]. Our tool uses by default criterion (b), which was most effective in experiments, but switches to (a) in cases where the proof search is considered stuck.

In practice the optimization problem is solved by encoding the optimization constraints in SMT and solving a maxSMT problem using Yices [5]. In order to orient equations, MædMax uses (SMT encodings of) LPO, KBO, and linear polynomial interpretations, as well as a dynamic choice among these at runtime depending on which satisfies the criteria best.

Success Checks. In phase (2), MædMax succeeds if (a) a goal can be joined or (b) a ground complete system was found. In the latter case, a ground goal is decided by checking syntactic equality of the two term's normal forms. For non-ground goals basic and normalizing narrowing is supported. MædMax establishes

[1] http://cl-informatik.uibk.ac.at/software/maedmax/.
[2] https://www.lri.fr/~marche/tpdb/format.html.

ground confluence by verifying ground joinability of extended critical pairs. However, depending on the signature this may be nontrivial: while the property is decidable for some orders when enriching the signature with infinitely many constants, it is undecidable for a finite given signature [4]. Our tool supports the criteria of [10,18] to that end, and both kinds of signatures.

Selection. In the selection phase (3), MædMax picks new equations and goals, given the current set of equations $\mathcal{E}$ and a TRS $\mathcal{R} \in \mathfrak{R}(\mathcal{E})$. To that end, it computes the set $S(\mathcal{E})$ containing n new equations and $S_G(\mathcal{G}, \mathcal{E})$ containing m new goals, where by default $n = 12$ and $m = 2$. In the *auto* mode the number n gets adjusted depending on the current state to avoid dealing with too many equations. The selection heuristic prefers small equations and old, but not yet reducible equations.

Order Generation Mode. MædMax can also be run for a couple of iterations and output the term ordering that is deemed best according to the criteria mentioned above (maximal number of oriented equations, etc.).

Certification. MædMax can output proofs in an XML format (CPF) that are checkable by the Isabelle-based certifier CeTA. For the case where a goal is proven (answer YES), certification follows the approach of [16]: The XML certificate gives a stepwise derivation of the goal from the input equations [16] which is checked by CeTA. Due to recent work [6] also some NO answers are checkable, though ground joinability support in CeTA is limited since the criterion from [18] is not supported.

Optimizations. Fingerprint indexing [13] is used to speed up both rewriting and overlap computation. In order to deal with associative and commutative symbols, the approach of [1] is incorporated. In the *auto* mode the tool also triggers restarts: if a current state is considered stuck, the procedure is restarted but where the input problem is extended by a number of small lemmas found so far.

5 Evaluation

All details of the following experiments can be obtained from the website. The tests were run single threaded on an Intel® Core™ i7-5930K CPU at 3.50 GHz with 12 cores, with varying timeouts as indicated below.

Table 1 compares MædMax with Waldmeister (WM) [1], E [14], and Vampire [9] on different problem sets. The first two lines refer to satisfiable/unsatisfiable problems in TPTP's unit equality division [17]. The third row refers to the 23 problems for which a ground complete system is given in [10] (which are hence all satisfiable, in the TPTP terminology). The fourth row refers to 731 problems generated by the conditional confluence tool ConCon to check infeasibility of critical pairs [15, Sect. 7.5], which are partially satisfiable and partially unsatisfiable. The last row refers to 139 satisfiable problems for standard completion [11].

Table 1. Experimental results.

	MædMax	WM	E	Vampire
TPTP UEQ SAT (600 s)	14	9	12	11
TPTP UEQ UNSAT (600 s)	621	832	692	724
Examples in [10] (60 s)	7	4	4	3
ConCon examples (60 s)	704	657	705	704
KB examples (60 s)	91	45	84	48

For TPTP problems the timeout was set to 600 s; for the latter two data sets 60s were chosen since larger timeouts did not induce any changes. Table 1 shows that MædMax outperforms other tools on satisfiable examples.[3] On unsatisfiable examples MædMax does not prevail, but all CPF proofs of unsatisfiability produced by MædMax have been certified by CeTA, they can be found on-line. For 14 problems the proof output for CeTA cannot be accomplished within a timeout of 1200 s, though.

On average, MædMax spends most of its running time on finding the TRSs $\mathfrak{R}(\mathcal{E})$ (20%), critical pair computation (33%), and overlap computation (33%). Only about 1% of the time is actually spent in the SMT solver.

We tested the order generation mode of MædMax with E since it accepts precedence and weight parameters for LPO and KBO as command line options. To that end MædMax was run for 10 s, and the devised reduction order was passed to E. In this way, E solved 605 unsatisfiable and 10 satisfiable TPTP UEQ problems. Though the number of solved examples is lowered wrt. Table 1 the average time is reduced, and different problems could be solved.

We conclude with a practical application example. The tool AQL[4] performs functorial data integration by means of a category-theoretic approach [12], taking advantage of (ground) completion. The following example problem was communicated by the authors.

Example 2. We consider database tables yIsAL and yIsAW relating amphibians to land and water animals, respectively. They are described by 400 ground equations over symbols $\mathsf{yIsAL}, \mathsf{yIsALL}, \mathsf{yIsAW}, \mathsf{yIsAWW}$ and 449 constants of the form $\mathsf{a}_i, \mathsf{w}_i, \mathsf{l}_i$ representing data entities. We give six example equations to convey an impression:

$$\begin{array}{ccc} \mathsf{yIsAW}(\mathsf{a}_1) \approx \mathsf{w}_{29} & \mathsf{yIsAW}(\mathsf{a}_{78}) \approx \mathsf{w}_{16} & \mathsf{yIsAW}(\mathsf{a}_{61}) \approx \mathsf{w}_{30} \\ \mathsf{yIsAL}(\mathsf{a}_{37}) \approx \mathsf{l}_{80} & \mathsf{yIsAL}(\mathsf{a}_{84}) \approx \mathsf{l}_{6} & \mathsf{yIsAL}(\mathsf{a}_{29}) \approx \mathsf{l}_{47} \end{array}$$

In addition, the equation $\mathsf{yIsALL}(\mathsf{yIsAL}(x)) \approx \mathsf{yIsAWW}(\mathsf{yIsAW}(x))$ describes a mapping to a second database schema. A ground complete presentation of the

[3] The Maxcomp version presented in [11] solves 91 KB examples within 60 s, too, but 98 problems in 600 s. For the other tools the numbers hardly change with a larger timeout. Maxcomp is not applicable to the other problem sets though.

[4] http://categoricaldata.net/aql.html.

entire system thus constitutes a representation of the data, translated to the second schema. MædMax discovers a complete presentation of 889 rules in less then 20 s, while AQL's internal completion prover fails. MædMax' automatic mode switches to linear polynomials for such systems with many symbols, which turned out to be faster than LPO or KBO in this situation.

For Example 2 even a complete system can be found. In general ground completeness is achieved by MædMax for those problems encountered by AQL, as required. Further details can be found on the website.

6 Conclusion

We presented the tool MædMax implementing maximal ordered completion, a novel approach to ordered completion and equational theorem proving. Our experiments show that this approach outperforms other tools on satisfiable problems. For unsatisfiable problems MædMax can produce output verifiable by the trusted proof checker CeTA, thus offering a very high degree of reliability. We believe that MædMax is particularly suited to problems with large signatures like Example 2, due to its ability to search for a reduction order which induces a small number of critical pairs and hence fewer steps to completion.

Acknowledgements. The authors thank Ryan Wisnesky for sharing AQL problems, and the anonymous referees for their helpful comments.

References

1. Avenhaus, J., Hillenbrand, T., Löchner, B.: On using ground joinable equations in equational theorem proving. JSC **36**(1–2), 217–233 (2003). https://doi.org/10.1016/S0747-7171(03)00024-5
2. Baader, F., Nipkow, T.: Term Rewriting and All That. Cambridge University Press, Cambridge (1998). https://doi.org/10.1017/CBO9781139172752
3. Bachmair, L., Dershowitz, N., Plaisted, D.A.: Completion without failure. In: Aït Kaci, H., Nivat, M. (eds.) Resolution of Equations in Algebraic Structures, Rewriting Techniques of Progress in Theoretical Computer Science, vol. 2, pp. 1–30. Academic Press, Cambridge (1989)
4. Comon, H., Narendran, P., Nieuwenhuis, R., Rusinowitch, M.: Deciding the confluence of ordered term rewrite systems. ACM TOCL **4**(1), 33–55 (2003). https://doi.org/10.1145/601775.601777
5. Dutertre, B., de Moura, L.: A fast linear-arithmetic solver for DPLL(T). In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 81–94. Springer, Heidelberg (2006). https://doi.org/10.1007/11817963_11
6. Hirokawa, N., Middeldorp, A., Sternagel, C., Winkler, S.: Infinite runs in abstract completion. In: Proceedings of the 2nd FSCD. LIPIcs, vol. 84, pp. 19:1–19:16 (2017). https://doi.org/10.4230/LIPIcs.FSCD.2017.19
7. Klein, D., Hirokawa, N.: Maximal completion. In: Proceedings of the 22nd RTA. LIPIcs, vol. 10, pp. 71–80 (2011). https://doi.org/10.4230/LIPIcs.RTA.2011.71

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, Oxford (1970). https://doi.org/10.1016/B978-0-08-012975-4
9. Kovács, L., Voronkov, A.: First-order theorem proving and Vampire. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 1–35. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39799-8_1
10. Martin, U., Nipkow, T.: Ordered rewriting and confluence. In: Stickel, M.E. (ed.) CADE 1990. LNCS, vol. 449, pp. 366–380. Springer, Heidelberg (1990). https://doi.org/10.1007/3-540-52885-7_100
11. Sato, H., Winkler, S.: Encoding dependency pair techniques and control strategies for maximal completion. In: Felty, A.P., Middeldorp, A. (eds.) CADE 2015. LNCS, vol. 9195, pp. 152–162. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21401-6_10
12. Schultz, P., Wisnesky, R.: Algebraic data integration. JFP **27**(e24), 51 (2017). https://doi.org/10.1017/S0956796817000168
13. Schulz, S.: Fingerprint indexing for paramodulation and rewriting. In: Gramlich, B., Miller, D., Sattler, U. (eds.) IJCAR 2012. LNCS, vol. 7364, pp. 477–483. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31365-3_37
14. Schulz, S.: System description: E 1.8. In: McMillan, K., Middeldorp, A., Voronkov, A. (eds.) LPAR 2013. LNCS, vol. 8312, pp. 735–743. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-45221-5_49
15. Sternagel, T.: Reliable confluence analysis of conditional term rewrite systems. Ph.D. thesis. University of Innsbruck (2017)
16. Sternagel, T., Winkler, S., Zankl, H.: Recording completion for certificates in equational reasoning. In: Proceedings of CPP 2015, pp. 41–47 (2015). https://doi.org/10.1145/2676724.2693171
17. Sutcliffe, G.: The TPTP problem library and associated infrastructure: the FOF and CNF parts. JAR **43**(4), 337–362 (2009). https://doi.org/10.1007/s10817-009-9143-8
18. Winkler, S.: A ground joinability criterion for ordered completion. In: Proceedings of 6th IWC, pp. 45–49 (2017)