Decidable Classes of Inductive Theorems*

Jürgen Giesl[1] and Deepak Kapur[2]

[1] LuFG Informatik II, RWTH Aachen, Ahornstr. 55, 52074 Aachen, Germany,
giesl@informatik.rwth-aachen.de
[2] Computer Science Dept., University of New Mexico, Albuquerque, NM 87131, USA
kapur@cs.unm.edu

Abstract. Kapur and Subramaniam [8] defined syntactical classes of equations where inductive validity is decidable. Thus, their validity can be checked without any user interaction and hence, this allows an integration of (a restricted form of) induction in fully automated reasoning tools such as model checkers. However, the results of [8] were only restricted to equations. This paper extends the classes of conjectures considered in [8] to a larger class of arbitrary quantifier-free formulas (e.g., conjectures also containing negation, conjunction, disjunction, etc.).

1 Introduction

Inductive theorem provers usually require massive manual intervention and they may waste huge amounts of time on proof attempts which fail due to the incompleteness of the prover. Therefore, induction has not yet been integrated in fully automated reasoning systems (i.e., model checkers) used for hardware and protocol verification, static and type analyses, byte-code verification, and proof-carrying codes. Most such push-button systems use a combination of decision procedures for theories such as Presburger arithmetic, propositional satisfiability, and data structures including bit vectors, arrays, and lists. However, extending these tools by the capability to perform induction proofs would be very desirable, since induction is frequently needed to reason about structured and parameterized circuits (e.g., n-bit adders or multipliers), the timing behavior of circuits with feedback loops, and code using loops and/or recursion.

For that reason, Kapur and Subramaniam proposed an approach for integrating induction schemes suggested by terminating function definitions with decision procedures, and gave a syntactical characterization of a class of equations where inductive validity is decidable using decision procedures and the cover set method for mechanizing induction [8, 11]. For those equations, induction proofs can be accomplished without any user interaction and they only fail if the conjecture is not valid. In Section 2, we give a simple characterization which

* *Proceedings of the International Joint Conference on Automated Reasoning, IJCAR 2001*, LNAI 2083, pp. 469-484, Springer-Verlag, 2001. Supported by the Deutsche Forschungsgemeinschaft Grant GI 274/4-1 and the National Science Foundation Grants nos. CCR-9996150 and CDA-9503064.

extends the class of decidable equations in [8]. Subsequently, we further extend the approach to arbitrary quantifier-free formulas, i.e., we define classes of such formulas where inductive validity is decidable. The crucial concept for this characterization are so-called *correctness predicates*. For a quantifier-free conjecture φ, c_φ is a correctness predicate iff for any tuple of (constructor) ground terms q^*, the truth of $c_\varphi(q^*)$ implies the truth of $\varphi[x^*/q^*]$ (cf. [6, 9]). We present a technique for automatically generating correctness predicates in Section 3.

The truth of a correctness predicate is only sufficient, but not necessary for the truth of the corresponding conjecture. In Section 4 we examine for which equations φ the correctness predicate is *exact* (i.e., the truth of $c_\varphi(q^*)$ is both sufficient and necessary for the truth of $\varphi[x^*/q^*]$). We develop a characterization to recognize (a subclass of) these equations automatically. In Section 5 we show that the use of exact correctness predicates allows us to extend the decidable classes of inductive theorems from equations to arbitrary quantifier-free formulas.

Our results are also useful for conventional inductive theorem provers since exact correctness predicates can be used to simplify the proof of conjectures like $\mathsf{double}(y) = y \Rightarrow y = \mathsf{0}$ where inductive provers would fail otherwise.

Even though the paper focuses on constructor systems and the decidable theory of quantifier-free formulas on free constructors, we believe the approach extends to other decidable theories $\mathcal{T}$ as well (e.g., Presburger arithmetic).

2 Equations where Inductive Validity is Decidable

We use term rewrite systems $\mathcal{R}$ (TRSs) as our programming language [1]. In a TRS, all root symbols of left-hand sides are called *defined* and all other function symbols of $\mathcal{R}$ are *constructors*. We only consider constructor systems (CSs), i.e., TRSs where the left-hand sides contain no defined symbols below the root position, even though most of the results in this paper generalize to more general theory-based systems, called $\mathcal{T}$-based systems in [8], with a decidable theory $\mathcal{T}$, in which arguments to defined symbols are terms from $\mathcal{T}$. Moreover, we restrict ourselves to (ground-)convergent and sufficiently complete CSs $\mathcal{R}$, i.e., for every ground term t there exists a unique constructor ground term q such that $t \to^*_\mathcal{R} q$. (A term containing only variables and constructors is called a constructor term; a constructor term without variables is a constructor ground term.)

For induction proofs, we use the concept of *cover sets* [7, 11]. A cover set is a finite set of pairs $\mathcal{C} = \{\langle s^*_1, \{t^*_{1,1}, \ldots, t^*_{1,n_1}\}\rangle, \ldots, \langle s^*_m, \{t^*_{m,1}, \ldots, t^*_{m,n_m}\}\rangle\}$, where s^*_i and $t^*_{i,j}$ are n-tuples of terms (for some $n \geq 0$). $\mathcal{C}$ is *complete* if for every n-tuple q^* of constructor ground terms, there is an s^*_i and a substitution σ such that $s^*_i\sigma = q^*$. Every cover set $\mathcal{C}$ induces a relation $<_\mathcal{C}$ on tuples of constructor ground terms: $p^* <_\mathcal{C} q^*$ iff there exists a pair $\langle s^*_i, \{t^*_{i,1}, \ldots, t^*_{i,n_i}\}\rangle \in \mathcal{C}$ such that $s^*_i\sigma = q^*$ and $t^*_{i,j}\sigma \to^*_\mathcal{R} p^*$. $\mathcal{C}$ is called *well-founded* iff $<_\mathcal{C}$ is well founded.[1]

A quantifier-free formula φ is *inductively valid* (or "valid" for short), denoted "$\mathcal{R} \models_{\mathrm{ind}} \varphi$", iff $\forall y^*\ \varphi$ holds in the initial model of the equations of $\mathcal{R}$ (where y^*

[1] $<_\mathcal{C}$ is well founded if there exists no infinite sequence $\ldots t_3 <_\mathcal{C} t_2 <_\mathcal{C} t_1 <_\mathcal{C} t_0$.

are the variables in φ).[2] For example, consider the following CS:

$$\mathsf{half}(\mathsf{0}) \to \mathsf{0}, \quad \mathsf{half}(\mathsf{s}(\mathsf{0})) \to \mathsf{0}, \quad \mathsf{half}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{s}(\mathsf{half}(x)).$$

This function definition suggests the cover set $\mathcal{C}_{\mathsf{half}} = \{\langle \mathsf{0}, \varnothing\rangle, \langle \mathsf{s}(\mathsf{0}), \varnothing\rangle, \langle \mathsf{s}(\mathsf{s}(x)), \{x\}\rangle\}$. To prove φ by induction w.r.t. $\mathcal{C}_{\mathsf{half}}$ (using the induction variable y), one obtains the base formulas $\varphi[y/\mathsf{0}]$ and $\varphi[y/\mathsf{s}(\mathsf{0})]$ and the step formula $\varphi[y/x] \Rightarrow \varphi[y/\mathsf{s}(\mathsf{s}(x))]$. Here, $\varphi[y/x]$ is the *induction hypothesis* and $\varphi[y/\mathsf{s}(\mathsf{s}(x))]$ is the *induction conclusion*. When proving a conjecture φ containing a term $f(y_1, \ldots, y_n)$, a successful heuristic for the choice of an induction relation is to perform induction w.r.t. $\mathcal{C}_f$ using the induction variables $y_1, \ldots, y_n$, cf. [2, 11].

Kapur and Subramaniam [8] characterized classes of equations where inductive validity is decidable (the decision procedure consists of an induction proof attempt w.r.t. a particular cover set). The observation is that if each induction formula built according to some cover set $\mathcal{C}$ only contains terms from an underlying decidable theory, then validity of the original conjecture can be decided.

Def. 1 and Thm. 2 apply to general $\mathcal{T}$-based systems, but due to lack of space, we focus on the decidable quantifier-free theory of free constructors in this paper. Here, $r[s^*]$ abbreviates $r[y^*/s^*]$ where y^* contains all variables in r.

Definition 1 ($\mathcal{C}$-provability). *Let $\mathcal{R}$ be a convergent sufficiently complete CS and let $\mathcal{C}$ be a complete well-founded cover set. An equation $r_1 = r_2$ is* $\mathcal{C}$-provable *w.r.t. $\mathcal{R}$ iff r_2 is a constructor term, for every $\langle s_i^*, \{t_{i,1}^*, \ldots, t_{i,n}^*\}\rangle \in \mathcal{C}$, s_i^* and all $t_{i,j}^*$ are tuples of constructor terms, and there exists a constructor term context C_i such that $r_1[s_i^*] \to_{\mathcal{R}}^* C_i[r_1[t_{i,1}^*], \ldots, r_1[t_{i,n}^*]]$.*

As an example, let us extend the CS for half by the rules $\mathsf{double}(\mathsf{0}) \to \mathsf{0}$ and $\mathsf{double}(\mathsf{s}(x)) \to \mathsf{s}(\mathsf{s}(\mathsf{double}(x)))$. Then the equation $\mathsf{double}(\mathsf{half}(y)) = y$ is $\mathcal{C}_{\mathsf{half}}$-provable. As required, the term y is a constructor term. Moreover, we obtain

$$\begin{array}{lll}
r_1[s_1] = \mathsf{double}(\mathsf{half}(\mathsf{0})) & \to_{\mathcal{R}}^* \mathsf{0} & \text{and thus, } C_1 = \mathsf{0}, \\
r_1[s_2] = \mathsf{double}(\mathsf{half}(\mathsf{s}(\mathsf{0}))) & \to_{\mathcal{R}}^* \mathsf{0} & \text{and thus, } C_2 = \mathsf{0}, \\
r_1[s_3] = \mathsf{double}(\mathsf{half}(\mathsf{s}(\mathsf{s}(x)))) & \to_{\mathcal{R}}^* \mathsf{s}(\mathsf{s}(\mathsf{double}(\mathsf{half}(x)))) & \text{and thus, } C_3 = \mathsf{s}(\mathsf{s}(\Box)).
\end{array}$$

Since $\mathcal{C}$-provability is decidable, Def. 1 characterizes a decidable class of conjectures. Instead of checking $\mathcal{C}$-provability directly, several sufficient conditions for $\mathcal{C}$-provability were given in [8]. We obtain the following theorem.

Theorem 2 (Decidability of inductive validity for equations). *Let $\mathcal{R}$ be a convergent sufficiently complete CS, let $\mathcal{C}$ be a complete well-founded cover set, and let $r_1 = r_2$ be a $\mathcal{C}$-provable equation. Then inductive validity of $r_1 = r_2$ is decidable (by attempting an induction proof w.r.t. $\mathcal{C}$).*

Proof. The decision procedure works by constructing the formulas

$$C_i[r_2[t_{i,1}^*], \ldots, r_2[t_{i,n}^*]] = r_2[s_i^*] \tag{1}$$

[2] $\mathcal{R} \models_{\mathrm{ind}} \varphi$ means that for all constructor ground terms q^*, $\varphi[y^*/q^*]$ follows from $\mathcal{R}$'s equations and axioms stating that different constructor ground terms are not equal.

for all $\langle s_i^*, \{t_{i,1}^*, \ldots, t_{i,n}^*\}\rangle \in \mathcal{C}$. As these equations only contain constructor terms, their validity is decidable.

It turns out that $r_1 = r_2$ is valid iff all these equations are valid. For the "if"-direction, notice that (1) implies the induction formula

$$r_1[t_{i,1}^*] = r_2[t_{i,1}^*] \wedge \ldots \wedge r_1[t_{i,n}^*] = r_2[t_{i,n}^*] \;\Rightarrow\; r_1[s_i^*] = r_2[s_i^*].$$

Thus, the validity of $r_1 = r_2$ follows by Noetherian induction. For the "only if"-direction, note that the validity of $r_1 = r_2$ implies the validity of (1). □

Since $\mathsf{double}(\mathsf{half}(y)) = y$ is $\mathcal{C}_{\mathsf{half}}$-provable, the above decision procedure can determine its validity. It has to check the validity of the equations

$$C_1[r_2[t_1]] = r_2[s_1], \text{ i.e., } \mathsf{0} = \mathsf{0}, \tag{2}$$
$$C_2[r_2[t_2]] = r_2[s_2], \text{ i.e., } \mathsf{0} = \mathsf{s}(\mathsf{0}), \tag{3}$$
$$C_3[r_2[t_3]] = r_2[s_3], \text{ i.e., } \mathsf{s}(\mathsf{s}(x)) = \mathsf{s}(\mathsf{s}(x)). \tag{4}$$

Since these equations only contain constructor terms, their validity is decidable. (Obviously, such an equation is valid iff both terms in the equation are syntactically identical.) While (2) and (4) are valid, the second equation (3) is not valid and thus, the conjecture $\mathsf{double}(\mathsf{half}(y)) = y$ is not valid either.

Our aim is to extend the result of Thm. 2 to more general formulas (i.e., not just equations), provided that all equations in these formulas are $\mathcal{C}$-provable. For example, we would like to consider formulas like $\mathsf{double}(\mathsf{half}(y)) = y \Rightarrow \mathsf{even}(y) = \mathsf{true}$ or $\mathsf{double}(y) = y \Rightarrow y = \mathsf{0}$. Equations appearing in these formulas are neither valid nor unsatisfiable; consequently, there is a need to characterize the subset of instantiations for the variables for which these equations are true. For this extension, we need the notion of correctness predicates.

3 Correctness Predicates

We present a technique which automatically generates algorithms for so-called *correctness predicates* c_φ for equations φ. For any tuple of constructor ground terms q^*, the truth of $c_\varphi(q^*)$ implies that $\varphi[y^*/q^*]$ is valid. Our definition of correctness predicates is similar to the definitions of [6, 9], but its form is quite restricted since we are interested in ensuring that validity of correctness predicates is decidable and that exact correctness predicates can be generated which completely characterize the domain of values on which the conjecture holds.

We have seen that the proof of the conjecture $\mathsf{double}(\mathsf{half}(y)) = y$ can be attempted by induction w.r.t. the cover set $\mathcal{C}_{\mathsf{half}}$. If $y = \mathsf{0}$, the conjecture can be reduced to the equation (2) which is always true. In the case $y = \mathsf{s}(\mathsf{0})$ we obtain the equation (3) which is always false. Finally, in the step case where $y = \mathsf{s}(\mathsf{s}(x))$, we have to prove that the induction hypothesis $\mathsf{double}(\mathsf{half}(x)) = x$ implies the induction conclusion $\mathsf{double}(\mathsf{half}(\mathsf{s}(\mathsf{s}(x)))) = \mathsf{s}(\mathsf{s}(x))$. As shown in Section 2, $\mathsf{double}(\mathsf{half}(\mathsf{s}(\mathsf{s}(x))))$ evaluates to $\mathsf{s}(\mathsf{s}(\mathsf{double}(\mathsf{half}(x))))$. Due to the induction hypothesis, we can replace the subterm $\mathsf{double}(\mathsf{half}(x))$ by x. Thus,

we obtain the equation (4) (which is always true). Hence, provided that the induction hypothesis is valid, the induction conclusion would also be valid. This gives rise to the following rules for the correctness predicate $c_{\mathsf{double}(\mathsf{half}(y))=y}$:

$$c_{\mathsf{double}(\mathsf{half}(y))=y}(\mathsf{0}) \to \mathsf{true}, \tag{5}$$

$$c_{\mathsf{double}(\mathsf{half}(y))=y}(\mathsf{s}(\mathsf{0})) \to \mathsf{false}, \tag{6}$$

$$c_{\mathsf{double}(\mathsf{half}(y))=y}(\mathsf{s}(\mathsf{s}(x))) \to\ c_{\mathsf{double}(\mathsf{half}(y))=y}(x). \tag{7}$$

Thus, we have synthesized the even algorithm. Note that the rule (7) is stronger than the following rule one would have gotten from the above analysis:

$$c_{\mathsf{double}(\mathsf{half}(y))=y}(\mathsf{s}(\mathsf{s}(x))) \to\ \mathsf{true}\ \text{ if }\ c_{\mathsf{double}(\mathsf{half}(y))=y}(x).$$

Since we want to generate unconditional rewrite rules for the definition of correctness predicates and to synthesize a complete definition, we use the form (7). As a result, the correctness predicate so generated may not be exact, and hence, provides only a sufficient condition for the conjecture to be valid.

In general, to prove a $\mathcal{C}$-provable equation $r_1 = r_2$ w.r.t. a cover set $\mathcal{C}$, for each pair $\langle s_i^*, \{t_{i,1}^*, \ldots, t_{i,n_i}^*\}\rangle \in \mathcal{C}$ we must check whether the equation $C_i[r_2[t_{i,1}^*], \ldots, r_2[t_{i,n}^*]] = r_2[s_i^*]$ is valid, cf. Equation (1) in the proof of Thm. 2. In order to obtain correctness predicates as simple as the ones above, we have to demand that these equations are either valid for *all* instantiations or for *none*. This ensures that the right-hand sides of the rules for correctness predicates only have the form true, false, or recursive calls of correctness predicates.

Definition 3 (Radical equations). *Let $\mathcal{R}$ be a convergent sufficiently complete CS and let $\mathcal{C} = \{\langle s_1^*, \{t_{1,1}^*, \ldots, t_{1,n_1}^*\}\rangle, \ldots, \langle s_m^*, \{t_{m,1}^*, \ldots, t_{m,n_m}^*\}\rangle\}$ be a complete well-founded cover set. An equation $r_1 = r_2$ is* radical *under $\mathcal{C}$ iff $r_1 = r_2$ is a $\mathcal{C}$-provable equation where $r_1[s_i^*] \to_{\mathcal{R}}^* C_i[r_1[t_{i,1}^*], \ldots, r_1[t_{i,n_i}^*]]$ for a constructor term context C_i and for all $1 \leq i \leq m$ we have*

$$\mathcal{R} \models_{\mathrm{ind}} \quad C_i[r_2[t_{i,1}^*], \ldots, r_2[t_{i,n_i}^*]]] = r_2[s_i^*] \quad \textit{or}$$
$$\mathcal{R} \models_{\mathrm{ind}} \neg C_i[r_2[t_{i,1}^*], \ldots, r_2[t_{i,n_i}^*]]] = r_2[s_i^*].$$

Note that since all C_i, s_i^*, and t_i^* are constructor terms, it is decidable whether a $\mathcal{C}$-provable equation is radical. The reason is that one only has to check whether an equation between two constructor terms is valid or unsatisfiable. Obviously, such an equation is unsatisfiable iff the two terms are not unifiable. For instance, the equation $\mathsf{double}(\mathsf{half}(y)) = y$ is radical under $\mathcal{C}_{\mathsf{half}}$ since the terms in the equations (2) - (4) are either identical or not unifiable.

To ease the presentation, we will now restrict ourselves to cover sets where there is at most one induction hypothesis for every induction step case.[3] Thus,

[3] The definition of correctness predicates can be easily generalized to the case of multiple induction hypotheses. In fact, correctness predicates can be defined for arbitrary equations, i.e., they do not have to be $\mathcal{C}$-provable or radical as required in this paper. However, these requirements are necessary in order to generate exact correctness predicates c_φ for arbitrary conjectures φ, such that validity of c_φ is decidable.

we only consider cover sets with pairs $\langle s_i^*, \{t_{i,1}^*, \ldots, t_{i,n_i}^*\}\rangle$ where $0 \leq n_i \leq 1$. Then we obtain the following definition of correctness predicates.

Definition 4 (Correctness Predicate). *Let $\mathcal{R}$, $\mathcal{C}$, $r_1 = r_2$ be as in Def. 3 where $0 \leq n_i \leq 1$ for all $1 \leq i \leq m$ and let $r_1 = r_2$ be* radical *under $\mathcal{C}$. Then the correctness predicate $c_{r_1=r_2}$ under $\mathcal{C}$ is defined by the following rules:*

$$c_{r_1=r_2}(s_i^*) \to \begin{cases} \mathsf{true}, \text{ if } \mathcal{R} \models_{\text{ind}} \quad C_i = r_2[s_i^*] \text{ and } n_i = 0, & (8) \\ \mathsf{false}, \text{ if } \mathcal{R} \models_{\text{ind}} \neg C_i = r_2[s_i^*] \text{ and } n_i = 0, & (9) \end{cases}$$

$$c_{r_1=r_2}(s_i^*) \to \begin{cases} c_{r_1=r_2}(t_{i,1}^*), \text{ if } \mathcal{R} \models_{\text{ind}} \quad C_i[r_2[t_{i,1}^*]] = r_2[s_i^*] \text{ and } n_i = 1, & (10) \\ \mathsf{false}, \qquad \text{if } \mathcal{R} \models_{\text{ind}} \neg C_i[r_2[t_{i,1}^*]] = r_2[s_i^*] \text{ and } n_i = 1. & (11) \end{cases}$$

Thm. 5 proves that a correctness predicate indeed represents a sufficient, but not a necessary condition for the soundness of the corresponding equation.

Theorem 5 (Correctness predicates are sufficient, but not necessary). *Let $\mathcal{R}$, $\mathcal{C}$, $r_1 = r_2$ be as in Def. 4. Let $c_{r_1=r_2}$ be a correctness predicate for $r_1 = r_2$ under $\mathcal{C}$ and let $\mathcal{R}$ also contain the rules defining $c_{r_1=r_2}$. Then we have*

(a) $\mathcal{R} \models_{\text{ind}} c_{r_1=r_2}(y^*) = \mathsf{true} \Rightarrow r_1 = r_2$.
(b) In general, we have $\mathcal{R} \not\models_{\text{ind}} r_1 = r_2 \Rightarrow c_{r_1=r_2}(y^*) = \mathsf{true}$.

Proof.

(a) Let q^* be a tuple of constructor ground terms such that $\mathcal{R} \models_{\text{ind}} c_{r_1=r_2}(q^*) = \mathsf{true}$. We prove $\mathcal{R} \models_{\text{ind}} r_1[q^*] = r_2[q^*]$ by induction w.r.t. $<_{\mathcal{C}}$. Due to the completeness of the cover set, there exists some $\langle s^*, \{t_1^*, \ldots, t_n^*\}\rangle \in \mathcal{C}$ and some substitution σ such that $q^* = s^*\sigma$ and since $r_1 = r_2$ is $\mathcal{C}$-provable (due to its radicality), we have $\mathcal{R} \models_{\text{ind}} r_1[s^*] = C[r_1[t_1^*], \ldots, r_1[t_n^*]]$.
If $n = 0$, then we also have $\mathcal{R} \models_{\text{ind}} C = r_2[s^*]$ and thus $\mathcal{R} \models_{\text{ind}} r_1[s^*] = r_2[s^*]$. If $n = 1$, we have $\mathcal{R} \models_{\text{ind}} C[r_2[t_1^*]] = r_2[s^*]$ and $\mathcal{R} \models_{\text{ind}} c_{r_1=r_2}(t_1^*\sigma) = \mathsf{true}$. The induction hypothesis yields $\mathcal{R} \models_{\text{ind}} r_1[t_1^*\sigma] = r_2[t_1^*\sigma]$. From the validity of $r_1[s^*] = C[r_1[t_1^*]]$ and $C[r_2[t_1^*]] = r_2[s^*]$, $\mathcal{R} \models_{\text{ind}} r_1[s^*\sigma] = r_2[s^*\sigma]$.

(b) Consider the equation $\mathsf{half}(y) = \mathsf{s}(\mathsf{0})$ and induction w.r.t. the cover set $\mathcal{C}_{\mathsf{half}}$. In the base cases $y = \mathsf{0}$ and $y = \mathsf{s}(\mathsf{0})$ the resulting conjecture $\mathsf{0} = \mathsf{s}(\mathsf{0})$ is unsatisfiable and in the step case, the induction conclusion $\mathsf{half}(\mathsf{s}(\mathsf{s}(x))) = \mathsf{s}(\mathsf{0})$ can be evaluated to $\mathsf{s}(\mathsf{half}(x)) = \mathsf{s}(\mathsf{0})$. Applying the induction hypothesis $\mathsf{half}(x) = \mathsf{s}(\mathsf{0})$ yields $\mathsf{s}(\mathsf{s}(\mathsf{0})) = \mathsf{s}(\mathsf{0})$ which is unsatisfiable. So the equation $\mathsf{half}(y) = \mathsf{s}(\mathsf{0})$ is radical under $\mathcal{C}_{\mathsf{half}}$ and we obtain the rules $c_{\mathsf{half}(y)=\mathsf{s}(\mathsf{0})}(\mathsf{0}) \to \mathsf{false}$, $c_{\mathsf{half}(y)=\mathsf{s}(\mathsf{0})}(\mathsf{s}(\mathsf{0})) \to \mathsf{false}$, and $c_{\mathsf{half}(y)=\mathsf{s}(\mathsf{0})}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{false}$. So $c_{\mathsf{half}(y)=\mathsf{s}(\mathsf{0})}$ is always false, but $\mathsf{half}(y) = \mathsf{s}(\mathsf{0})$ holds for $\mathsf{s}^2(\mathsf{0})$ and $\mathsf{s}^3(\mathsf{0})$. □

In fact, a correctness predicate $c_\varphi(q^*)$ yields true iff the equation φ holds for both q^* *and* for all arguments p^* which are smaller than q^* w.r.t. the induction relation induced by the cover set. For that reason, the correctness predicate $c_{\mathsf{half}(y)=\mathsf{s}(\mathsf{0})}$ returns false for the arguments $\mathsf{s}^2(\mathsf{0})$ and $\mathsf{s}^3(\mathsf{0})$ although the conjecture is true, since it is false for the smaller arguments $\mathsf{0}$ and $\mathsf{s}(\mathsf{0})$.

4 Conjectures with Exact Correctness Predicate

In this section we characterize equations $r_1 = r_2$ where the correctness predicate $c_{r_1=r_2}$ is *exact*, i.e., for all q^*, $c_{r_1=r_2}(q^*)$ is true iff $\mathcal{R} \models_{\mathrm{ind}} r_1[q^*] = r_2[q^*]$. Exactness is ensured if in Def. 4, whenever Rule (10) is used, the induction conclusion $r_1[s_i^*] = r_2[s_i^*]$ is equivalent to $r_1[t_{i,1}^*] = r_2[t_{i,1}^*]$. As we have seen in Sect. 3, $c_{r_1=r_2}(q^*)$ only returns true if $r_1 = r_2$ is true for q^* and for all p^* smaller than q^* w.r.t. the induction relation induced by the cover set. Thus, $c_{r_1=r_2}$ is only exact if $r_1[q^*] = r_2[q^*]$ implies the validity of $r_1[p^*] = r_2[p^*]$ for all arguments $p^* <_{\mathcal{C}} q^*$. So $c_{r_1=r_2}$ only describes the exact set of instantiations where $r_1 = r_2$ is valid, if each induction conclusion implies all its induction hypotheses.

Consider again the proof of $\mathsf{double}(\mathsf{half}(y)) = y$ by induction w.r.t. $\mathcal{C}_{\mathsf{half}}$. We obtain the induction conclusion $\mathsf{double}(\mathsf{half}(\mathsf{s}(\mathsf{s}(x)))) = \mathsf{s}(\mathsf{s}(x))$ and the induction hypothesis $\mathsf{double}(\mathsf{half}(x)) = x$. Indeed, this conjecture has the desired property

$$\mathcal{R} \models_{\mathrm{ind}} \mathsf{double}(\mathsf{half}(\mathsf{s}(\mathsf{s}(x)))) = \mathsf{s}(\mathsf{s}(x)) \;\Rightarrow\; \mathsf{double}(\mathsf{half}(x)) = x. \qquad (12)$$

To see this, note that in the first base case where $y = \mathsf{0}$, the left-hand side $\mathsf{double}(\mathsf{half}(\mathsf{0}))$ evaluates to $\mathsf{0}$, which is smaller than or equal to the right-hand side $\mathsf{0}$ (if terms are compared by the subterm relation, for example). Similarly, in the second base case where $y = \mathsf{s}(\mathsf{0})$, the left-hand side evaluates to $\mathsf{0}$, which is again smaller than or equal to the right-hand side $\mathsf{s}(\mathsf{0})$. In the step case, the left hand side of the induction conclusion can be evaluated to

$$\mathsf{s}(\mathsf{s}(\underline{\mathsf{double}(\mathsf{half}(x))})) = \mathsf{s}(\mathsf{s}(\underline{x})).$$

This evaluated induction conclusion *contains* the induction hypothesis, since the underlined terms are the terms on both sides of the induction hypothesis. (This observation also forms the basis of the rippling technique [3].) Thus, when going from the induction hypothesis to the induction conclusion, both sides of the equation grow by the context $\mathsf{s}(\mathsf{s}(\Box))$. In other words, in the induction base cases the left-hand side is at most as great as the right-hand side and afterwards, the left-hand side always grows at most as much as the right-hand side. Thus, if one ever reaches an instantiation t where $\mathsf{double}(\mathsf{half}(t)) = t$ is no longer true, then the reason is that $\mathsf{double}(\mathsf{half}(t))$ is *smaller* then t. But since $\mathsf{double}(\mathsf{half}(y))$ grows at most as fast as y, afterwards there can never be a number $s >_{\mathcal{C}_{\mathsf{half}}} t$ where $\mathsf{double}(\mathsf{half}(s)) = s$ is true again. Hence, if the induction hypothesis $\mathsf{double}(\mathsf{half}(x)) = x$ is false, then the induction conclusion $\mathsf{double}(\mathsf{half}(\mathsf{s}(\mathsf{s}(x)))) = \mathsf{s}(\mathsf{s}(x))$ is false as well (or, formulated as a contraposition, we have Property (12)).

The observation above leads to a general criterion. For many $\mathcal{C}$-provable equations $r_1 = r_2$, one does not only have $r_1[s_i^*] \to_{\mathcal{R}}^* C_i[r_1[t_{i,1}^*], \ldots, r_1[t_{i,n_i}^*]]$ for all $\langle s_i^*, \{t_{i,1}^*, \ldots, t_{i,n_i}^*\}\rangle \in \mathcal{C}$, but also $r_2[s_i^*] = D_i[r_2[t_{i,1}^*], \ldots, r_2[t_{i,n_i}^*]]$ for some constructor ground contexts C_i and D_i.

In our example, r_1 is $\mathsf{double}(\mathsf{half}(y))$ and r_2 is the term y. For the first pair of the cover set $\mathcal{C}_{\mathsf{half}}$, we have $C_1 = \mathsf{0}$ and $D_1 = \mathsf{0}$ and for the second pair we have $C_2 = \mathsf{0}$ and $D_2 = \mathsf{s}(\mathsf{0})$. For the third pair, we have $r_1[s_3^*] = \mathsf{double}(\mathsf{half}(\mathsf{s}(\mathsf{s}(x))))$,

which can be evaluated to $\mathsf{s}(\mathsf{s}(\mathsf{double}(\mathsf{half}(x))))$ and as $t^*_{3,1} = x$, we obtain $C_3 = \mathsf{s}(\mathsf{s}(\Box))$. Since $r_2[s^*_3] = \mathsf{s}(\mathsf{s}(x))$, we also have $D_3 = \mathsf{s}(\mathsf{s}(\Box))$.

So r_1 grows by the context C_i and r_2 grows by the context D_i when going from the induction hypothesis $r_1[t^*_{i,1}] = r_2[t^*_{i,1}]$ to the induction conclusion $r_1[s^*_i] = r_2[s^*_i]$. Our aim is to ensure that whenever r_1 and r_2 are no longer $\mathcal{R}$-equal for some instantiation, then they will never become equal again for arguments which are greater w.r.t. the induction relation induced by the cover set. A sufficient requirement for this is that the contexts C_i added around r_1 are always at most as big as the contexts D_i added around r_2. To compare these contexts one can use an arbitrary ordering $\prec$ on constructor terms, i.e., any relation which is transitive and irreflexive. Moreover, we require $\prec$ to be monotonic (i.e., $s \prec t$ implies $f(\ldots s \ldots) \prec f(\ldots t \ldots)$ for all constructors f) and stable under substitutions (i.e., $s \prec t$ implies $s\sigma \prec t\sigma$). Then we only have to demand

$$C_i[x^*] \preceq D_i[x^*] \text{ for all } 1 \leq i \leq m.$$

As usual, "$\preceq$" denotes the union of "$\prec$" and "=" where "=" is syntactic equality.

Note that one may use any well-established technique for the generation of well-founded orderings such as the subterm ordering or the *recursive path ordering* $<_{\mathrm{rpo}}$ (cf. e.g. [5, 10]) to synthesize a suitable ordering $\prec$ satisfying the above constraints. Moreover, since $\prec$ only has to be irreflexive, but not necessarily well founded, one can also use any ordering $>$ which results from the reversal of such a well-founded ordering $<$ (e.g., the superterm ordering or $>_{\mathrm{rpo}}$).

In our example we need a well-founded monotonic stable ordering $\prec$ where

$$\begin{aligned} C_1 = \mathsf{0} &\preceq \mathsf{0} = D_1, \\ C_2 = \mathsf{0} &\preceq \mathsf{s}(\mathsf{0}) = D_2, \\ C_3[x] = \mathsf{s}(\mathsf{s}(x)) &\preceq \mathsf{s}(\mathsf{s}(x)) = D_3[x]. \end{aligned}$$

Such an ordering can easily found by standard techniques for automated termination proofs. For example, the constraints are satisfied by the subterm ordering. Thus, one can automatically determine that $\mathsf{double}(\mathsf{half}(y)) = y$ is a conjecture whose correctness predicate is exact. As $c_{\mathsf{double}(\mathsf{half}(y))=y}$ is only true for even numbers, we have shown that indeed this conjecture is false for all odd ones.

In general, if $r_1 = r_2$ is an equation and $\mathcal{C}$ is a cover set such that the above conditions are satisfied by some ordering $\prec$, then we say that $r_1 = r_2$ *maintains* $\prec$ under the cover set $\mathcal{C}$ w.r.t. the underlying CS $\mathcal{R}$. The reason is that the relation $\prec$ between r_1 and r_2 is indeed maintained when going from an induction hypothesis to an induction conclusion. By using established (and decidable classes of) well-founded orderings $\prec$ from the area of term rewrite systems one immediately obtains a syntactical sufficient condition for maintenance of orderings, which can easily be checked automatically.

Definition 6 (Maintenance of orderings). *Let $\mathcal{R}$ be a convergent sufficiently complete CS and let $\mathcal{C} = \{\langle s^*_1, \{t^*_{1,1}, \ldots, t^*_{1,n_1}\}\rangle, \ldots, \langle s^*_m, \{t^*_{m,1}, \ldots, t^*_{m,n_m}\}\rangle\}$ be a complete well-founded cover set (where $0 \leq n_i \leq 1$ for all $1 \leq i \leq m$). Let*

$r_1 = r_2$ *be* $\mathcal{C}$*-provable and let* C_i *and* D_i *be constructor ground contexts where*

$$r_1[s_i^*] \to_{\mathcal{R}}^* C_i[r_1[t_{i,1}^*], \ldots, r_1[t_{i,n_i}^*]] \quad \textit{and}$$
$$r_2[s_i^*] \;=\; D_i[r_2[t_{i,1}^*], \ldots, r_2[t_{i,n_i}^*]].$$

Let $\prec$ *be a monotonic ordering on constructor terms which is stable under substitutions. We say* $r_1 = r_2$ maintains $\prec$ *under the cover set* $\mathcal{C}$ *w.r.t.* $\mathcal{R}$ *iff* $C_i[x^*] \preceq D_i[x^*]$ *for all* $1 \leq i \leq m$.

The following lemma proves that for equations which maintain an ordering, each induction conclusion indeed implies its induction hypothesis.

Lemma 7 (Equations where the reverse induction formulas hold). *Let* $\mathcal{R}$, $\mathcal{C}$, $\prec$ *be as in Def. 6 and let* $r_1 = r_2$ *maintain* $\prec$ *under* $\mathcal{C}$ *w.r.t.* $\mathcal{R}$. *Then for all* $1 \leq i \leq m$ *with* $n_i = 1$, $\mathcal{R} \models_{\mathrm{ind}} r_1[s_i^*] = r_2[s_i^*] \;\Rightarrow\; r_1[t_{i,1}^*] = r_2[t_{i,1}^*]$.

Proof. We first show that for all constructor ground terms q^*, we have

$$r_1[q^*]{\downarrow_{\mathcal{R}}} \preceq r_2[q^*]. \tag{13}$$

The proof of (13) is done by induction w.r.t. $<_{\mathcal{C}}$. Due to the completeness of $\mathcal{C}$, there must be a pair $\langle s_i^*, \{t_{i,1}^*, \ldots, t_{i,n_i}^*\}\rangle \in \mathcal{C}$ such that $s_i^*\sigma = q^*$. If $n_i = 0$, then we have $r_1[q^*]{\downarrow_{\mathcal{R}}} = r_1[s_i^*\sigma]{\downarrow_{\mathcal{R}}} = C_i \preceq D_i = r_2[s_i^*\sigma] = r_2[q^*]$.

Otherwise, if $n_i = 1$, we have $r_1[q^*]\downarrow_{\mathcal{R}} = r_1[s_i^*\sigma]\downarrow_{\mathcal{R}} = C_i[r_1[t_{i,1}^*\sigma]\downarrow_{\mathcal{R}}] \preceq C_i[r_2[t_{i,1}^*\sigma]]$ by the induction hypothesis and monotonicity and stability of $\prec$. Furthermore, $C_i[r_2[t_{i,1}^*\sigma]] \preceq D_i[r_2[t_{i,1}^*\sigma]] = r_2[s_i^*\sigma] = r_2[q^*]$. So (13) is proved.

Now we can prove Lemma 7. Let σ substitute all variables of s_i^* by constructor ground terms such that $\mathcal{R} \models_{\mathrm{ind}} r_1[s_i^*\sigma] = r_2[s_i^*\sigma]$. We assume that $\mathcal{R} \not\models_{\mathrm{ind}} r_1[t_{i,1}^*\sigma] = r_2[t_{i,1}^*\sigma]$. By (13) we must have $r_1[t_{i,1}^*\sigma]{\downarrow_{\mathcal{R}}} \preceq r_2[t_{i,1}^*\sigma]$ and since the $\mathcal{R}$-normal forms of $r_1[t_{i,1}^*\sigma]$ and $r_2[t_{i,1}^*\sigma]$ are different by assumption this in fact implies $r_1[t_{i,1}^*\sigma]{\downarrow_{\mathcal{R}}} \prec r_2[t_{i,1}^*\sigma]$. Since $\prec$ is monotonic and stable we have

$$r_1[s_i^*\sigma]{\downarrow_{\mathcal{R}}} = C_i[r_1[t_{i,1}^*\sigma]{\downarrow_{\mathcal{R}}}] \prec C_i[r_2[t_{i,1}^*\sigma]] \preceq D_i[r_2[t_{i,1}^*\sigma]] = r_2[s_i^*\sigma].$$

But this contradicts $\mathcal{R} \models_{\mathrm{ind}} r_1[s_i^*\sigma] = r_2[s_i^*\sigma]$ by the irreflexivity of $\prec$. □

Now we prove that if $r_1 = r_2$ maintains an ordering, then $c_{r_1=r_2}$ is indeed exact.

Theorem 8 (Equations where the correctness predicate is exact). *Let* $\mathcal{R}$, $\mathcal{C}$, $\prec$ *be as in Def. 6 and let* $r_1 = r_2$ *be an equation which is radical and maintains some ordering* $\prec$ *under* $\mathcal{C}$ *w.r.t.* $\mathcal{R}$. *Moreover, let* $c_{r_1=r_2}$ *be a correctness predicate for* $r_1 = r_2$ *under* $\mathcal{C}$ *and let* $\mathcal{R}$ *also contain the rules defining* $c_{r_1=r_2}$. *Then* $\mathcal{R} \models_{\mathrm{ind}} r_1 = r_2 \;\Leftrightarrow\; c_{r_1=r_2}(y^*) = \mathsf{true}$.[4]

[4] A more general version of this theorem can be proved in which a conjecture does not have to be radical, and further, it is not necessary for the induction scheme of a cover set to have at most one induction hypothesis in every subgoal.

Proof. Due to Thm. 5 (a) we only have to prove $\mathcal{R} \models_{\text{ind}} r_1[q^*] = r_2[q^*] \Rightarrow c_{r_1=r_2}(q^*) = \mathsf{true}$ for all constructor ground term tuples q^*. Again, we use induction on $<_{\mathcal{C}}$. Let $\mathcal{R} \models_{\text{ind}} r_1[q^*] = r_2[q^*]$.

By the completeness of $\mathcal{C}$, there exists some $\langle s^*, \{t_1^*, \ldots, t_n^*\}\rangle \in \mathcal{C}$ and some substitution σ such that $q^* = s^*\sigma$. If $n = 0$, then we have the rule $c_{r_1=r_2}(s^*) \to \mathsf{true}$ since the rule $c_{r_1=r_2}(s^*) \to \mathsf{false}$ would only be generated if $\mathcal{R} \models_{\text{ind}} \neg r_1[s^*] = r_2[s^*]$. This implies $\mathcal{R} \models_{\text{ind}} c_{r_1=r_2}(q^*) = \mathsf{true}$.

Otherwise, if $n = 1$, by Lemma 7 the truth of $r_1[s_i^*\sigma] = r_2[s_i^*\sigma]$ implies $\mathcal{R} \models_{\text{ind}} r_1[t_{i,1}^*\sigma] = r_2[t_{i,1}^*\sigma]$. So $\mathcal{R} \models_{\text{ind}} c_{r_1=r_2}(t_{i,1}^*\sigma)$ by the induction hypothesis. By the rule $c_{r_1=r_2}(s^*) \to c_{r_1=r_2}(t_1^*)$, we obtain $\mathcal{R} \models_{\text{ind}} c_{r_1=r_2}(s_i^*\sigma) = \mathsf{true}$. □

Let us consider the counterexample of Thm. 5 (b) again. When trying to prove $\mathsf{half}(y) = \mathsf{s}(\mathsf{0})$, we obtain $C_1 = \mathsf{0}$, $D_1 = \mathsf{s}(\mathsf{0})$ and $C_2 = \mathsf{0}$, $D_2 = \mathsf{s}(\mathsf{0})$. In the step case, the left-hand side $\mathsf{half}(\mathsf{s}(\mathsf{s}(x)))$ evaluates to $\mathsf{s}(\mathsf{half}(x))$, i.e., we have $C_3 = \mathsf{s}(\Box)$, whereas $D_3 = \Box$. There does not exist an ordering $\prec$ such that $C_i[x^*] \preceq D_i[x^*]$ for all i, since $C_1 \preceq D_1$ would imply $\mathsf{0} \prec \mathsf{s}(\mathsf{0})$ and $C_3[\mathsf{0}] \preceq D_3[\mathsf{0}]$ would imply $\mathsf{s}(\mathsf{0}) \prec \mathsf{0}$ which contradicts the transitivity and irreflexivity of $\prec$. Thus, $\mathsf{half}(y) = \mathsf{s}(\mathsf{0})$ does not maintain any ordering under $\mathcal{C}_{\mathsf{half}}$ and indeed, its correctness predicate is not exact as shown in Thm. 5 (b).

The above analysis of exactness of correctness predicates can be useful for fixing faulty conjectures, an objective for which correctness predicates were introduced by Protzen [9]. Since an exact correctness predicate precisely characterizes all instantiations on which the faulty conjecture is true, it can be used to fix the faulty conjecture into the "strongest theorem" possible.

5 Conjectures where Inductive Validity is Decidable

Now we extend Thm. 2 from equations to arbitrary quantifier-free formulas φ. We require that all equations $r_1 = r_2$ occurring in φ are radical and maintain some ordering under the same cover set $\mathcal{C}$.[5] Then by Thm. 8 their correctness predicates $c_{r_1=r_2}$ are sound and exact. For example, $\mathsf{half}(y) = \mathsf{0}$ is radical and maintains the superterm ordering under $\mathcal{C}_{\mathsf{half}}$. We obtain the correctness predicate

$$c_{\mathsf{half}(y)=\mathsf{0}}(\mathsf{0}) \to \mathsf{true}, \quad c_{\mathsf{half}(y)=\mathsf{0}}(\mathsf{s}(\mathsf{0})) \to \mathsf{true}, \quad c_{\mathsf{half}(y)=\mathsf{0}}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{false}.$$

The last rule is due to the fact that the instantiated left-hand side $\mathsf{half}(\mathsf{s}(\mathsf{s}(x)))$ evaluates to $\mathsf{s}(\mathsf{half}(x))$ and the replacement of the subterm $\mathsf{half}(x)$ according to the induction hypothesis yields the equation $\mathsf{s}(\mathsf{0}) = \mathsf{0}$ which is unsatisfiable.

[5] Different equations in a conjecture may have to be proved using different cover sets; these cover sets can often be combined into a single cover set to generate a single induction scheme using merging and instantiation (cf. [2, 7]). Further, it is not necessary for different equations to maintain the same monotonic ordering. For instance, in the running example of this section two different orderings are used in a conjecture.

Given a correctness predicate c_φ, we can generate $c_{\neg\varphi}$ by replacing the result true by false and the result false by true whereas right-hand sides of the form $c_\varphi(t^*)$ are replaced by $c_{\neg\varphi}(t^*)$. In the above example this yields

$$c_{\neg\mathsf{half}(y)=\mathsf{0}}(\mathsf{0}) \to \mathsf{false}, \quad c_{\neg\mathsf{half}(y)=\mathsf{0}}(\mathsf{s}(\mathsf{0})) \to \mathsf{false}, \quad c_{\neg\mathsf{half}(y)=\mathsf{0}}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{true}.$$

This correctness predicate is sound and exact for the conjecture $\neg\mathsf{half}(y) = \mathsf{0}$.

As stated before, exact correctness predicates can also be generated for non-radical equations, as well as for equations whose validity is decided using induction schemes with multiple induction hypotheses. Thus, inductive validity of a much larger class of literals (equations and negated equations) can be decided using arbitrary well-founded complete cover sets without the requirement of radicality. The restrictions to radical equations and to induction schemes involving at most one induction step in every subgoal are needed only for the decidability of conjunctions and disjunctions of conjectures as discussed below.

Given c_{φ_1} and c_{φ_2}, a straightforward idea to obtain rules for $c_{\varphi_1\wedge\varphi_2}$ is as follows: If we have the rule $c_{\varphi_i}(s^*) \to \mathsf{false}$ for some $i \in \{1,2\}$, then we also obtain the rule $c_{\varphi_1\wedge\varphi_2}(s^*) \to \mathsf{false}$. If we have the rules $c_{\varphi_i}(s^*) \to \mathsf{true}$ for both $i \in \{1,2\}$, then we obtain $c_{\varphi_1\wedge\varphi_2}(s^*) \to \mathsf{true}$. Finally, if we have the rule $c_{\varphi_i}(s^*) \to c_{\varphi_i}(t^*)$ and either $c_{\varphi_j}(s^*) \to c_{\varphi_j}(t^*)$ or $c_{\varphi_j}(s^*) \to \mathsf{true}$ (for $i, j \in \{1,2\}$, $i \neq j$), then we also obtain the rule $c_{\varphi_1\wedge\varphi_2}(s^*) \to c_{\varphi_1\wedge\varphi_2}(t^*)$. But as the following example illustrates, such a simplistic construction does not work.

Recall the rules (5) - (7) for $c_{\mathsf{double}(\mathsf{half}(y))=y}$. We would obtain the following correctness predicate for the formula $\varphi : \mathsf{double}(\mathsf{half}(y)) = y \wedge \neg\mathsf{half}(y) = \mathsf{0}$.

$$c_\varphi(\mathsf{0}) \to \mathsf{false}, \quad c_\varphi(\mathsf{s}(\mathsf{0})) \to \mathsf{false}, \quad c_\varphi(\mathsf{s}(\mathsf{s}(x))) \to \ c_\varphi(x).$$

However, this correctness predicate is not exact, since it is always false, whereas φ is true for all even numbers greater than $\mathsf{0}$. Even worse, the resulting correctness predicate for the negated conjecture $\neg\varphi$ would not even be sound (since it would always be true whereas $\neg\varphi$ is false for $\mathsf{0}$ and all odd numbers).

The problem with the above construction of $c_{\varphi_1\wedge\varphi_2}$ is the case where one rule $c_{\varphi_1}(s^*) \to c_{\varphi_1}(t^*)$ leads to a recursive call, but the other has the form $c_{\varphi_2}(s^*) \to \mathsf{true}$. If we use the rule $c_{\varphi_1\wedge\varphi_2}(s^*) \to c_{\varphi_1\wedge\varphi_2}(t^*)$, then we may lose the exactness of the correctness predicate, since it could be that $c_{\varphi_2}(t^*) \to^* \mathsf{false}$.

To avoid this problem, we will now construct so-called *basic* correctness predicates (denoted $b_{r_1=r_2}$) where for recursive pairs $\langle s^*, \{t^*\}\rangle \in \mathcal{C}$ we always have recursive rules $b_{r_1=r_2}(s^*) \to b_{r_1=r_2}(t^*)$, but never a rule with the result false.

Fortunately, if $r_1 = r_2$ is radical and maintains an ordering under $\mathcal{C}$, one can easily obtain a basic correctness predicate by simply extending the cover set $\mathcal{C}$ in an appropriate way. For that purpose we have to restrict ourselves to cover sets where for any two recursive pairs $\langle s_i^*, \{t_i^*\}\rangle, \langle s_j^*, \{t_j^*\}\rangle \in \mathcal{C}$ with $i \neq j$, the terms t_i^* and s_j^* do not unify (after renaming their variables). In other words, the arguments t_i^* in an induction hypothesis must not unify with the arguments s_j^* in any *other* induction conclusion. The cover set $\mathcal{C}_{\mathsf{half}} = \{\langle \mathsf{0}, \varnothing\rangle, \langle \mathsf{s}(\mathsf{0}), \varnothing\rangle, \langle \mathsf{s}(\mathsf{s}(x)), \{x\}\rangle\}$ trivially satisfies this condition, since there is only one recursive pair. The motivation for this restriction is that for all chains $q_0^* <_{\mathcal{C}} q_1^* <_{\mathcal{C}} \ldots <_{\mathcal{C}} q_n^*$, it ensures

$c_\varphi(q_n^*) = \ldots = c_\varphi(q_1^*)$. So a change in the value of c_φ can only occur in the last value q_0^*, which corresponds to a base case (i.e., we might have $c_\varphi(q_1^*) \neq c_\varphi(q_0^*)$). Our aim is to extend $\mathcal{C}$ to a cover set $\mathcal{C}'$ where q_1^* is already a base case. Then for all chains $q_1^* <_{\mathcal{C}'} \ldots <_{\mathcal{C}'} q_n^*$ we have $c_\varphi(q_n^*) = \ldots = c_\varphi(q_1^*)$ and thus, we can indeed use the rule $c_\varphi(s^{*\prime}) \to c_\varphi(t^{*\prime})$ for all recursive pairs $\langle s^{*\prime}, \{t^{*\prime}\}\rangle$ of $\mathcal{C}'$.

The idea for the extension of cover sets is simply to unify the terms t_i^* of the induction hypotheses with the (variable-renamed) terms s_j^* in the left components of all pairs from $\mathcal{C}$. Let $\mu_{i,j}$ be the respective mgu's. Then every pair $\langle s_i^*, \{t_i^*\}\rangle$ is replaced by the new non-recursive pairs $\langle s_i^*\mu_{i,j}, \varnothing\rangle$ for $j \neq i$ and the instantiated recursive pair $\langle s_i^*\mu_{i,i}, \{t_i^*\mu_{i,i}\}\rangle$. For $\mathcal{C}_{\mathsf{half}}$ we obtain

$$\mathcal{C}'_{\mathsf{half}} = \{\langle \mathsf{0}, \varnothing\rangle, \langle \mathsf{s}(\mathsf{0}), \varnothing\rangle, \langle \mathsf{s}(\mathsf{s}(\mathsf{0})), \varnothing\rangle, \langle \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0}))), \varnothing\rangle, \langle \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(x)))), \{\mathsf{s}(\mathsf{s}(x))\}\rangle\}.$$

Definition 9 (Extending cover sets). *Let* $\mathcal{C} = \{\langle s_1^*, \{t_{1,1}^*, \ldots, t_{1,n_1}^*\}\rangle, \ldots, \langle s_m^*, \{t_{m,1}^*, \ldots, t_{m,n_m}^*\}\rangle\}$ *be a cover set with* $0 \leq n_i \leq 1$*, such that if* $n_i = n_j = 1$ *and* $i \neq j$ *then there do not exist substitutions* $\mu_{i,j}$ *with* $t_{i,1}^*\mu_{i,j} = s_j^*\nu\mu_{i,j}$ *for a variable renaming* ν*. Then the* extended *cover set* $\mathcal{C}'$ *is defined as follows:*

$$\begin{aligned}\mathcal{C}'=&\{\langle s_i^*, \varnothing\rangle \mid n_i = 0\}\\ &\cup\{\langle s_i^*\mu_{i,j}, \varnothing\rangle \mid n_i = 1, n_j = 0, \mu_{i,j} = mgu(t_{i,1}^*, s_j^*\nu) \text{ for a variable renaming } \nu\}\\ &\cup\{\langle s_i^*\mu_{i,i}, \{t_{i,1}^*\mu_{i,i}\}\rangle \mid n_i = 1, \mu_{i,i} = mgu(t_{i,1}^*, s_i^*\nu) \text{ for a variable renaming } \nu\}.\end{aligned}$$

Obviously, if $\mathcal{C}$ is complete and well founded, then the extension $\mathcal{C}'$ is complete and well founded, too. Moreover, if an equation $r_1 = r_2$ is radical and maintains an ordering under $\mathcal{C}$, then it is also radical and maintains the same ordering under the extension $\mathcal{C}'$. In this case we can construct the basic correctness predicate by taking the extension $\mathcal{C}'$ and by using the results true and false in its non-recursive cases and by using the rule $b_{r_1=r_2}(s^*) \to b_{r_1=r_2}(t^*)$ for all recursive pairs $\langle s^*, \{t^*\}\rangle$. Note that only *one* such extension step for cover sets $\mathcal{C}$ is already enough: If a correctness predicate b has a non-recursive rule $b(s^*) \to \mathsf{true}$ or $b(s^*) \to \mathsf{false}$ for a recursive pair $\langle s^*, \{t^*\}\rangle \in \mathcal{C}$, then a single extension step of $\mathcal{C}$ suffices to get recursive rules $b(s^{*\prime}) \to b(t^{*\prime})$ for all recursive pairs $\langle s^{*\prime}, \{t^{*\prime}\}\rangle$ of the extended cover set $\mathcal{C}'$. In our example we obtain

$$\begin{aligned}
b_{\mathsf{half}(y)=\mathsf{0}}(\mathsf{0}) &\to \mathsf{true}, & b_{\mathsf{double}(\mathsf{half}(y))=y}(\mathsf{0}) &\to \mathsf{true},\\
b_{\mathsf{half}(y)=\mathsf{0}}(\mathsf{s}(\mathsf{0})) &\to \mathsf{true}, & b_{\mathsf{double}(\mathsf{half}(y))=y}(\mathsf{s}(\mathsf{0})) &\to \mathsf{false},\\
b_{\mathsf{half}(y)=\mathsf{0}}(\mathsf{s}^2(\mathsf{0})) &\to \mathsf{false}, & b_{\mathsf{double}(\mathsf{half}(y))=y}(\mathsf{s}^2(\mathsf{0})) &\to \mathsf{true},\\
b_{\mathsf{half}(y)=\mathsf{0}}(\mathsf{s}^3(\mathsf{0})) &\to \mathsf{false}, & b_{\mathsf{double}(\mathsf{half}(y))=y}(\mathsf{s}^3(\mathsf{0})) &\to \mathsf{false},\\
b_{\mathsf{half}(y)=\mathsf{0}}(\mathsf{s}^4(x)) &\to b_{\mathsf{half}(y)=\mathsf{0}}(\mathsf{s}^2(x)). & b_{\mathsf{double}(\mathsf{half}(y))=y}(\mathsf{s}^4(x)) &\to b_{\mathsf{double}(\mathsf{half}(y))=y}(\mathsf{s}^2(x)).
\end{aligned}$$

Now indeed basic correctness predicates for conjunctions are constructed by using the result false if one of the conjuncts yields false and true if both conjuncts yield true. If one (and therefore, both) conjuncts have a recursive call, then the basic correctness predicate for the conjunction has a recursive call, too. So if φ is again the formula $\mathsf{double}(\mathsf{half}(y)) = y \wedge \neg\mathsf{half}(y) = \mathsf{0}$, then we have

$$
\begin{aligned}
b_{\neg\mathsf{half}(y)=\mathsf{0}}(\mathsf{0}) &\to \mathsf{false},\\
b_{\neg\mathsf{half}(y)=\mathsf{0}}(\mathsf{s}(\mathsf{0})) &\to \mathsf{false},\\
b_{\neg\mathsf{half}(y)=\mathsf{0}}(\mathsf{s}^2(\mathsf{0})) &\to \mathsf{true},\\
b_{\neg\mathsf{half}(y)=\mathsf{0}}(\mathsf{s}^3(\mathsf{0})) &\to \mathsf{true},\\
b_{\neg\mathsf{half}(y)=\mathsf{0}}(\mathsf{s}^4(x)) &\to b_{\neg\mathsf{half}(y)=\mathsf{0}}(\mathsf{s}^2(x)).
\end{aligned}
\qquad
\begin{aligned}
b_{\varphi}(\mathsf{0}) &\to \mathsf{false},\\
b_{\varphi}(\mathsf{s}(\mathsf{0})) &\to \mathsf{false},\\
b_{\varphi}(\mathsf{s}^2(\mathsf{0})) &\to \mathsf{true},\\
b_{\varphi}(\mathsf{s}^3(\mathsf{0})) &\to \mathsf{false},\\
b_{\varphi}(\mathsf{s}^4(x)) &\to\ b_{\varphi}(\mathsf{s}^2(x)).
\end{aligned}
$$

Definition 10 (Basic Correctness Predicates). *Let $\mathcal{R}$ be a convergent sufficiently complete CS and let $\mathcal{C}$ be a complete well-founded cover set such that for all $\langle s^*, \{t_1^*, \ldots, t_n^*\}\rangle \in \mathcal{C}$, we have $0 \leq n \leq 1$, and for two different pairs $\langle s^*, \{t^*\}\rangle$, $\langle s^{*\prime}, \{t^{*\prime}\}\rangle \in \mathcal{C}$, there does not exist a substitution μ with $t^*\mu = s^{*\prime}\nu\mu$ for a variable renaming ν. Let φ be a quantifier-free formula such that all equations in φ are radical and maintain some ordering under $\mathcal{C}$ w.r.t. $\mathcal{R}$.*

Let $\mathcal{C}' = \{\langle s_1^*, \{t_{1,1}^*, \ldots, t_{1,n_1}^*\}\rangle, \ldots, \langle s_m^*, \{t_{m,1}^*, \ldots, t_{m,n_m}^*\}\rangle\}$ be the extension of $\mathcal{C}$ and let $r_1[s_i^*] \to_{\mathcal{R}}^* C_i[r_1[t_{i,1}^*], \ldots, r_1[t_{i,n_i}^*]]$ for a constructor ground context C_i. Then the basic correctness predicate b_φ under $\mathcal{C}$ is defined by the following rules (analogous rules are used for formulas containing $\vee$, $\Rightarrow$, $\Leftrightarrow$):

$$
b_{r_1=r_2}(s_i^*) \to \begin{cases} \mathsf{true}, & \text{if } \mathcal{R} \models_{\mathrm{ind}} C_i = r_2[s_i^*] \text{ and } n_i = 0,\\ \mathsf{false}, & \text{if } \mathcal{R} \models_{\mathrm{ind}} \neg C_i = r_2[s_i^*] \text{ and } n_i = 0,\\ b_{r_1=r_2}(t_{i,1}^*), & \text{if } n_i = 1, \end{cases}
$$

$$
b_{\neg\varphi'}(s_i^*) \to \begin{cases} \mathsf{true}, & \text{if we have the rule } b_{\varphi'}(s_i^*) \to \mathsf{false},\\ \mathsf{false}, & \text{if we have the rule } b_{\varphi'}(s_i^*) \to \mathsf{true},\\ b_{\neg\varphi'}(t_{i,1}^*), & \text{if we have the rule } b_{\varphi'}(s_i^*) \to b_{\varphi'}(t_{i,1}^*), \end{cases}
$$

$$
b_{\varphi_1\wedge\varphi_2}(s_i^*) \to \begin{cases} \mathsf{true}, & \text{if } b_{\varphi_1}(s_i^*) \to \mathsf{true} \text{ and } b_{\varphi_2}(s_i^*) \to \mathsf{true},\\ \mathsf{false}, & \text{if } b_{\varphi_1}(s_i^*) \to \mathsf{false} \text{ or } b_{\varphi_2}(s_i^*) \to \mathsf{false},\\ b_{\varphi_1\wedge\varphi_2}(t_{i,1}^*), & \text{if } b_{\varphi_1}(s_i^*) \to b_{\varphi_1}(t_{i,1}^*) \text{ and } b_{\varphi_2}(s_i^*) \to b_{\varphi_2}(t_{i,1}^*). \end{cases}
$$

Now we can present the main theorem which shows that the inductive validity of arbitrary quantifier-free conjectures is decidable, if all their equations are radical and maintain an ordering under $\mathcal{C}$. The decision procedure works by constructing the basic correctness predicate and by checking whether it always yields true. The reason for the soundness of this approach is that basic correctness predicates are indeed sound and exact.

Theorem 11 (Decidability of inductive validity for arbitrary conjectures). *Let $\mathcal{R}$, $\mathcal{C}$, φ be as in Def. 10. Then inductive validity of φ is decidable (by checking whether all non-recursive rules of b_φ have the right-hand side* true*, where b_φ is the basic correctness predicate for φ under $\mathcal{C}$).*

Proof. We have to show that b_φ is sound and exact, i.e., $\mathcal{R} \models_{\mathrm{ind}} \varphi \Leftrightarrow b_\varphi(y^*) =$ true if $\mathcal{R}$ also contains the rules defining b_φ. We use an induction w.r.t. the structure of φ. First let φ be an equation $r_1 = r_2$.

Let q^* be a tuple of constructor ground terms. We prove $\mathcal{R} \models_{\mathrm{ind}} r_1[q^*] = r_2[q^*] \Leftrightarrow b_{r_1=r_2}(q^*) =$ true by induction w.r.t. $<_{\mathcal{C}'}$. Since $\mathcal{C}$ is complete and well founded, obviously its extension $\mathcal{C}'$ is complete and well founded, too. Due

to the completeness of $\mathcal{C}'$, there exists some $\langle s^*, \{t_1^*, \ldots, t_n^*\}\rangle \in \mathcal{C}'$ and some substitution σ such that $q^* = s^*\sigma$. If $n = 0$, then the claim follows from radicality of $r_1 = r_2$ under $\mathcal{C}$ and thus, under $\mathcal{C}'$ as well.

If $n = 1$ and $\mathcal{R} \models_{\text{ind}} r_1[s^*\sigma] = r_2[s^*\sigma]$ then by Lemma 7 we also have $\mathcal{R} \models_{\text{ind}} r_1[t_1^*\sigma] = r_2[t_1^*\sigma]$ since $r_1 = r_2$ maintains an ordering under $\mathcal{C}$ and thus, under $\mathcal{C}'$ as well. The induction hypothesis yields $\mathcal{R} \models_{\text{ind}} b_{r_1=r_2}(t_1^*\sigma) = \mathsf{true}$ and thus, $\mathcal{R} \models_{\text{ind}} b_{r_1=r_2}(s^*\sigma) = \mathsf{true}$ as well.

Finally, let $n = 1$ and $\mathcal{R} \models_{\text{ind}} \neg r_1[s^*\sigma] = r_2[s^*\sigma]$. We have to show that this implies $\mathcal{R} \models_{\text{ind}} \neg r_1[t_1^*\sigma] = r_2[t_1^*\sigma]$. Then the induction hypothesis would yield $\mathcal{R} \models_{\text{ind}} b_{r_1=r_2}(t_1^*\sigma) = \mathsf{false}$ and thus, $\mathcal{R} \models_{\text{ind}} b_{r_1=r_2}(s^*\sigma) = \mathsf{false}$ as well.

Note that $s^* = s^{*\prime}\mu$ and $t_1^* = t_1^{*\prime}\mu$ for some $\langle s^{*\prime}, \{t_1^{*\prime}\}\rangle \in \mathcal{C}$ by the definition of extensions. Moreover, by the requirement that arguments $t_1^{*\prime}$ of induction hypotheses may not unify with arguments of other induction conclusions we also have that $t_1^* = t_1^{*\prime}\mu = s^{*\prime}\nu\mu$ by the definition of extensions. Since $r_1 = r_2$ maintains an ordering under $\mathcal{C}$ we have $r_1[s^{*\prime}] \to_{\mathcal{R}}^* C_i'[r_1[t_1^{*\prime}]]$ for a constructor *ground* context C_i'. As $r_1[s^*] \to_{\mathcal{R}}^* C_i[r_1[t_1^*]]$, this means that $C_i' = C_i$ or, in other words, $r_1[s^{*\prime}] \to_{\mathcal{R}}^* C_i[r_1[t_1^{*\prime}]]$. Radicality of $r_1 = r_2$ under $\mathcal{C}$ implies that $\mathcal{R} \models_{\text{ind}} C_i[r_2[t_1^{*\prime}]] = r_2[s^{*\prime}]$ or $\mathcal{R} \models_{\text{ind}} \neg C_i[r_2[t_1^{*\prime}]] = r_2[s^{*\prime}]$.

First assume $\mathcal{R} \models_{\text{ind}} C_i[r_2[t_1^{*\prime}]] = r_2[s^{*\prime}]$. This implies $\mathcal{R} \models_{\text{ind}} (C_i[r_2[t_1^{*\prime}]] = r_2[s^{*\prime}])\mu$, i.e., $\mathcal{R} \models_{\text{ind}} C_i[r_2[t_1^*]] = r_2[s^*]$. If we had $\mathcal{R} \not\models_{\text{ind}} \neg r_1[t_1^*\sigma] = r_2[t_1^*\sigma]$ (i.e., $\mathcal{R} \models_{\text{ind}} (r_1[t_1^*] = r_2[t_1^*])\sigma\tau$ for some τ), then we would also have $\mathcal{R} \models_{\text{ind}} (C_i[r_1[t_1^*]] = r_2[s^*])\sigma\tau$. Since $r_1[s^*] \to_{\mathcal{R}}^* C_i[r_1[t_1^*]]$, this implies $\mathcal{R} \models_{\text{ind}} (r_1[s^*] = r_2[s^*])\sigma\tau$ in contradiction to the prerequisite $\mathcal{R} \models_{\text{ind}} \neg r_1[s^*\sigma] = r_2[s^*\sigma]$.

Thus, $\mathcal{R} \models_{\text{ind}} \neg C_i[r_2[t_1^{*\prime}]] = r_2[s^{*\prime}]$. Again assume $\mathcal{R} \models_{\text{ind}} (r_1[t_1^*] = r_2[t_1^*])\sigma\tau$ for some τ. Since $t_1^*\sigma\tau = s^{*\prime}\nu\mu\sigma\tau$, we have $\mathcal{R} \models_{\text{ind}} (r_1[s^{*\prime}] = r_2[s^{*\prime}])\nu\mu\sigma\tau$ and since $r_1 = r_2$ maintains an ordering under $\mathcal{C}$, this implies $\mathcal{R} \models_{\text{ind}} (r_1[t_1^{*\prime}] = r_2[t_1^{*\prime}])\nu\mu\sigma\tau$ by Lemma 7. By the prerequisite, this yields $\mathcal{R} \models_{\text{ind}} (\neg C_i[r_1[t_1^{*\prime}]] = r_2[s^{*\prime}])\nu\mu\sigma\tau$. However since $r_1[s^{*\prime}] \to_{\mathcal{R}}^* C_i[r_1[t_1^{*\prime}]]$, this is equivalent to $\mathcal{R} \models_{\text{ind}} (\neg r_1[s^{*\prime}] = r_2[s^{*\prime}])\nu\mu\sigma\tau$, which contradicts the assumption (as $t_1^*\sigma\tau = s^{*\prime}\nu\mu\sigma\tau$).

For formulas which are no equations, the claim immediately follows from the (outer) induction hypothesis. □

Note that the conditions in Thm. 11 (i.e., radicality and maintenance of orderings) can be checked automatically (by using orderings from the area of term rewrite systems which are amenable to automation). The set of all conjectures φ satisfying these conditions forms a class where inductive validity is decidable. To decide inductive validity of φ one simply constructs the rules for the basic correctness predicate b_φ (which can be done automatically) and one checks whether there is no rule of the form $b_\varphi(\ldots) \to \mathsf{false}$.

So for a formula like $\mathsf{double}(y) = y \Rightarrow y = 0$, one first checks whether this formula belongs to the class where inductive validity is decidable. For that purpose, one examines whether the conjecture contains a subterm $f(y^*)$ for pairwise disjoint variables y^* and an algorithm f and then one checks whether all equations in the conjecture are radical and maintain an ordering under $\mathcal{C}_f$ (using the induction variables y^*).

In our example, the equations $\mathsf{double}(y) = y$ and $y = \mathsf{0}$ indeed are both radical and they maintain the superterm ordering under $\mathcal{C}_{\mathsf{double}}$. So inductive validity of this conjecture is decidable. The decision procedure constructs the basic correctness predicate

$$\begin{aligned} b_{\mathsf{double}(y)=y \Rightarrow y=\mathsf{0}}(\mathsf{0}) &\rightarrow \mathsf{true}, \\ b_{\mathsf{double}(y)=y \Rightarrow y=\mathsf{0}}(\mathsf{s}(\mathsf{0})) &\rightarrow \mathsf{true}, \\ b_{\mathsf{double}(y)=y \Rightarrow y=\mathsf{0}}(\mathsf{s}(\mathsf{s}(x))) &\rightarrow b_{\mathsf{double}(y)=y \Rightarrow y=\mathsf{0}}(\mathsf{s}(x)), \end{aligned}$$

and checks whether all non-recursive rules of $b_{\mathsf{double}(y)=y \Rightarrow y=\mathsf{0}}$ have true on their right-hand side, which is obviously the case. Thus, the formula is valid.

Note that in this way we can *decide* the inductive validity of conjectures which were up to now hard problems for inductive theorem provers. In fact, virtually all existing inductive provers fail in verifying $\mathsf{double}(y) = y \Rightarrow y = \mathsf{0}$.[6] The reason is that the induction conclusion $\mathsf{double}(\mathsf{s}(x)) = \mathsf{s}(x) \Rightarrow \mathsf{s}(x) = \mathsf{0}$ can be evaluated to $\neg\mathsf{s}(\mathsf{double}(x)) = x$, but there is no way to apply the induction hypothesis $\mathsf{double}(x) = x \Rightarrow x = \mathsf{0}$ and thus, the proof of the induction step case does not succeed. On the other hand, by our decision procedure, validity of such conjectures can be shown without using any inductive theorem prover at all.

6 Conclusion

We presented a class of conjectures where inductive validity is decidable (by a very simple decision procedure). This allows an integration of inductive reasoning within fully automated tools like model checkers or compilers. First, we extended the results of [8] to a larger class of equations and subsequently, we extended the approach further to arbitrary quantifier-free conjectures. The main idea is to build correctness predicates for all equations occurring in a conjecture and we gave a criterion for checking whether these correctness predicates really describe the exact set of objects where the equation is valid. We showed how to construct (basic) correctness predicates for non-atomic formulas and by checking their defining rules, the inductive validity of such formulas can easily be decided.

We have used correctness predicates $c_{r_1=r_2}$ to describe the instances where an equation $r_1 = r_2$ is valid. However, in order to *combine* the correctness predicates $c_{r_1=r_2}$ and $c_{r'_1=r'_2}$ of two different equations (e.g., when building their conjunction), we have to restrict ourselves to *basic* correctness predicates and moreover, $c_{r_1=r_2}$ and $c_{r'_1=r'_2}$ must have been built w.r.t. "compatible" cover sets. In order to avoid these difficulties, an interesting alternative approach is to represent the set of instances where equations are valid by *tree automata* [4] instead of correctness predicates. As long as these sets of instances are *regular*, this indeed results in a very elegant method for deciding inductive validity (since regular languages are effectively closed under complement and intersection and since their emptiness is decidable). However, in general there are many equations where the set of instances which makes them valid is not regular. For example,

[6] This problem was pointed out to us by U. Kühler.

the equation $\mathsf{plus}(\mathsf{minus}(x, y), \mathsf{minus}(y, x)) = \mathsf{0}$ is valid iff x and y are equal. A correctness predicate describing this set can easily be constructed automatically, whereas this set is not regular and therefore cannot be described by (ordinary) tree automata. This indicates that the use of tree automata may be too restrictive compared to the use of (basic) correctness predicates. However, we intend to study the possibilities of using automata for deciding inductive validity further in future work.

In this paper, we focused on integrating induction schemes with a decision procedure for the quantifier-free theory of free constructors to obtain an extension of the decision procedure to quantifier-free formulas whose proofs (or disproofs) may require the use of induction. Kapur and Subramaniam [8] discussed an approach for integrating induction schemes into decidable quantifier-free theories including Presburger arithmetic, and they gave a decision procedure for inductive validity of a large class of equations involving $\mathcal{T}$-based function symbols, where $\mathcal{T}$ is a decidable quantifier-free theory. In future work, we intend to generalize the techniques developed in this paper from constructor systems to $\mathcal{T}$-based systems (including Presburger arithmetic) as well.

References

1. F. Baader & T. Nipkow, *Term Rewriting and All That*, Cambridge Univ. Pr., 1998.
2. R. S. Boyer and J Moore, *A Computational Logic*, Academic Press, 1979.
3. A. Bundy, A. Stevens, F. van Harmelen, A. Ireland, & A. Smaill, Rippling: A Heuristic for Guiding Inductive Proofs, *Artificial Intelligence*, 62:185-253, 1993.
4. H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, & M. Tommasi. Tree Automata and Applications. Draft, available from `http://www.grappa.univ-lille3.fr/tata/`, 1999.
5. N. Dershowitz, Termination of Rewriting, *J. Symb. Comp.*, 3:69–116, 1987.
6. M. Franova & Y. Kodratoff, Predicate Synthesis from Formal Specifications, in *Proc. ECAI 92*, 1992.
7. D. Kapur & M. Subramaniam, New Uses of Linear Arithmetic in Automated Theorem Proving by Induction, *Journal of Automated Reasoning*, 16:39–78, 1996.
8. D. Kapur & M. Subramaniam, Extending Decision Procedures with Induction Schemes, in *Proc. CADE-17*, LNAI 1831, pages 324-345, 2000.
9. M. Protzen, Patching Faulty Conjectures, *Proc. CADE-13*, LNAI 1104, 1996.
10. J. Steinbach, Simplification orderings: History of results, *Fundamenta Informaticae*, 24:47–87, 1995.
11. H. Zhang, D. Kapur, & M. S. Krishnamoorthy, A Mechanizable Induction Principle for Equational Specifications, in *Proc. CADE-9*, LNCS 310, 1988.