Proving Program Properties
as First-Order Satisfiability

Salvador Lucas[✉]

DSIC, Universitat Politècnica de València, Valencia, Spain
slucas@dsic.upv.es
http://slucas.webs.upv.es/

Abstract. Program semantics can often be expressed as a (many-sorted) first-order theory $\mathcal{S}$, and program properties as sentences φ which are intended to hold in the *canonical model* of such a theory, which is often incomputable. Recently, we have shown that properties φ expressed as the existential closure of a boolean combination of atoms can be *disproved* by just finding a model of $\mathcal{S}$ and the *negation* $\neg\varphi$ of φ. Furthermore, this idea works quite well in practice due to the existence of powerful tools for the automatic generation of models for (many-sorted) first-order theories. In this paper we extend our previous results to *arbitrary* properties, expressed as sentences without any special restriction. Consequently, one can prove a program property φ by just *finding a model* of an appropriate theory (including $\mathcal{S}$ and possibly something else) and an appropriate first-order formula related to φ. Beyond its possible theoretical interest, we show that our results can also be of practical use in several respects.

Keywords: First-order logic · Logical models · Program analysis

1 Introduction

Given a first-order theory $\mathcal{S}$ and a sentence φ, finding a model $\mathcal{A}$ of $\mathcal{S} \cup \{\neg\varphi\}$, i.e., such that $\mathcal{A} \models \mathcal{S} \cup \{\neg\varphi\}$ holds, shows indeed that φ is *not* a logical consequence of $\mathcal{S}$: there is at least one model of $\mathcal{S}$ (e.g., $\mathcal{A}$) which does not satisfy φ (as it satisfies $\neg\varphi$). Provability of φ in $\mathcal{S}$, i.e., $\mathcal{S} \vdash \varphi$, implies (by correctness of the proof calculus) that φ is a logical consequence of $\mathcal{S}$ (written $\mathcal{S} \models \varphi$). Thus, $\mathcal{A} \models \mathcal{S} \cup \{\neg\varphi\}$ *disproves* φ regarding $\mathcal{S}$; this can be written $\neg(\mathcal{S} \vdash \varphi)$ by using some metalevel notation. In general, this does not allow us to conclude that $\neg\varphi$ is proved, i.e., $\mathcal{S} \vdash \neg\varphi$, or is a logical consequence of $\mathcal{S}$, i.e., $\mathcal{S} \models \neg\varphi$. What can be concluded about $\neg\varphi$ regarding $\mathcal{S}$ from the fact that $\mathcal{A} \models \mathcal{S} \cup \{\neg\varphi\}$ holds? Can this be advantageously used in a 'logic-based' approach to program analysis?

In [14], some answers to these questions are given: a sentence φ which is an *Existentially Closed Boolean Combination of Atoms* (ECBCA for short) does *not*

hold in the initial model $\mathcal{I}_\mathcal{S}$ of a theory $\mathcal{S}$ consisting of a set of ground atoms if we find a model $\mathcal{A}$ of $\mathcal{S} \cup \{\neg\varphi\}$ [14, Corollary 2]. This is useful in program analysis when considering programs P that are given a theory $\overline{P}$ representing its operational semantics so that the execution of P is described as a set $\mathcal{I}_P$ of (ground) atoms A which can be proved from $\overline{P}$ (i.e., $\mathcal{I}_P$ is the *initial* model of $\overline{P}$ in the usual first-order sense; in the following, we often refer to it as its *canonical* model [11, Sect. 1.5]). Actually, rather than being logical consequences of $\overline{P}$, the intended meaning of first-order sentences φ that represent properties of P is that they hold in the *initial model* of $\overline{P}$, see [4, Chap. 4], for instance.

In [14,16] we applied this approach to prove computational properties of rewriting-based systems in practice. This includes Term Rewriting Systems (TRSs [1]) and more general rewriting-based formalisms [3,9,18,19].

Example 1. Consider the following TRS $\mathcal{R}$ with the well-known rules defining the addition and product of natural numbers in Peano's notation:

$$\mathsf{add}(0, x) \to x \tag{1}$$
$$\mathsf{add}(\mathsf{s}(x), y) \to \mathsf{s}(\mathsf{add}(x, y)) \tag{2}$$

$$\mathsf{mul}(0, x) \to 0 \tag{3}$$
$$\mathsf{mul}(\mathsf{s}(x), y) \to \mathsf{add}(y, \mathsf{mul}(x, y)) \tag{4}$$

The associated theory $\overline{\mathcal{R}}$ is the following:

$$(\forall x)\, x \to^* x \qquad\qquad (\forall x)\, \mathsf{add}(0, x) \to x$$
$$(\forall x, y, z)\, x \to y \wedge y \to^* z \Rightarrow x \to^* z \qquad (\forall x, y)\, \mathsf{add}(\mathsf{s}(x), y) \to \mathsf{s}(\mathsf{add}(x, y))$$
$$(\forall x, y)\, x \to y \Rightarrow \mathsf{s}(x) \to \mathsf{s}(y) \qquad\qquad (\forall x)\, \mathsf{mul}(0, x) \to 0$$
$$(\forall x, y, z)\, x \to y \Rightarrow \mathsf{add}(x, z) \to \mathsf{add}(y, z) \quad (\forall x, y)\, \mathsf{mul}(\mathsf{s}(x), y) \to \mathsf{add}(y, \mathsf{mul}(x, y))$$
$$(\forall x, y, z)\, x \to y \Rightarrow \mathsf{add}(z, x) \to \mathsf{add}(z, y)$$
$$(\forall x, y, z)\, x \to y \Rightarrow \mathsf{mul}(x, z) \to \mathsf{mul}(y, z)$$
$$(\forall x, y, z)\, x \to y \Rightarrow \mathsf{mul}(z, x) \to \mathsf{mul}(z, y)$$

The first sentence in the first column represents *reflexivity* of many-step rewriting, with predicate symbol $\to^*$; the second sentence shows how one-step rewriting, with predicate symbol $\to$, contributes to $\to^*$. The next sentences describe the *propagation* of rewriting steps to (arguments of) symbols s, add and mul. The second column describes the rules of $\mathcal{R}$. More details can be found in [14, Sect. 4]. In the initial or *least Herbrand model* $\mathcal{I}_\mathcal{R}$ of $\overline{\mathcal{R}}$, $\to$ and $\to^*$ are interpreted as the sets $(\to)^{\mathcal{I}_\mathcal{R}}$ and $(\to^*)^{\mathcal{I}_\mathcal{R}}$ of all pairs (s, t) of ground terms s and t such that $s \to_\mathcal{R} t$ and $s \to_\mathcal{R}^* t$, respectively. Now, we can express the property *"the double of some natural number can be an odd number"* as an ECBCA:

$$(\exists x)(\exists y)(\exists z)\, \mathsf{add}(x, x) \to^* z \wedge \mathsf{s}(\mathsf{mul}(\mathsf{s}(\mathsf{s}(0)), y)) \to^* z \tag{5}$$

With the automatic model generator Mace4 [17] we find a model of $\overline{\mathcal{R}} \cup \{\neg(5)\}$ with domain $\mathcal{A} = \{0, 1\}$. Function symbols are interpreted as follows: $0^\mathcal{A} = 0$;

$s^{\mathcal{A}}(x) = 1 - x$; $\mathsf{add}^{\mathcal{A}}(x, y)$ returns 0 if $x = y$ and 1 otherwise; $\mathsf{mul}^{\mathcal{A}}(x, y)$ returns 1 if $x = y = 1$ and 0 otherwise. Predicates $\to$ and $\to^*$ are both interpreted as the Aphequality. Thus, we have *proved* that (5) does *not* hold for $\mathcal{R}$.

Our approach in [14] relies on the notion of *preservation* of a formula under homomorphisms h between interpretations. Roughly speaking, a homomorphism h *preserves* a formula φ if φ is satisfied in the target interpretation of h whenever φ is satisfied in its domain interpretation [11, Sect. 2.4]. Homomorphisms preserve ECBCA [11, Theorem 2.4.3(a)]; the results in [14] rely on this fact. In this paper we *extend* [14] to deal with more general program properties. Homomorphisms preserve other first-order sentences if further requirements are imposed: (i) *positive* sentences (where connective '¬' is absent) are preserved under *surjective* homomorphisms and (ii) *arbitrary* sentences are preserved under *embeddings* [11, Theorem 2.4.3]. In contrast to [14] (and [11]), here we focus on *many-sorted logic* [23] (see Sect. 2). This has an important advantage: since homomorphisms in many-sorted logic with set of sorts S are actually a *family h_s* of homomorphisms between components of sort s for each $s \in S$, the preservation requirements for h_s depend on the *specific quantification* of variables $x : s$ *for such a sort*. In Sect. 3 we provide a *unique* preservation theorem that subsumes the results in [14], and even improves [11]. Section 4 investigates how to guarantee surjectivity of homomorphisms. Section 5 shows several application examples taken from Table 1, which shows some properties of rewriting-based systems that could not be captured in [14] but we are able to handle now. Here, $t(\boldsymbol{x})$ is a term with variables $\boldsymbol{x}$ (or just t if it is *ground*), $\mathcal{C}$ (and $\mathcal{D}$) are the *constructor* (resp. *defined*) symbols in the TRS, and $\xrightarrow{\Lambda}$ is *topmost rewriting*. Section 6 discusses the possibility of providing more information about *disproved* properties by means of *refutation witnesses*, i.e., (counter)examples of *sentences* which are synthesized from the models that are used to disprove the property. Section 7 shows how to deal with completely general sentences by means of a simple example. Section 8 discusses some related work. Section 9 concludes.

Table 1. Some properties about rewriting-based systems

Property	φ
Ground reducible	$(\forall\boldsymbol{x})\,(\exists y)\;t(\boldsymbol{x}) \to y$
Completely defined symbol f	$(\forall\boldsymbol{x})(\exists y)\;f(x_1, \ldots, x_k) \to y$
Completely defined TRS	$(\forall\boldsymbol{x})(\exists\boldsymbol{y})\,\bigwedge_{f \in \mathcal{D}}\;f(x_1, \ldots, x_{ar(f)}) \to y_f$
Productive	$(\forall x)(\exists\boldsymbol{y})\;\bigvee_{c \in \mathcal{C}} x \to^* c(y_1, \ldots, y_k)$
Nonterminating	$(\exists x)(\forall n \in \mathbb{N})(\exists y)\;x \to^n y$
Infinitely root-reducible	$(\exists x)(\forall n \in \mathbb{N})(\exists y)\;x(\to^* \circ \xrightarrow{\Lambda})^n y$
Normalizing term	$(\exists x)\;(t \to^* x \wedge \neg(\exists y)\;x \to y)$
Normalizing TRS (WN)	$(\forall x)(\exists y)\;(x \to^* y \wedge \neg(\exists z)\;y \to z)$
Locally confluent (WCR)	$(\forall x, y, z)\;x \to y \wedge x \to z \Rightarrow (\exists u)\;x \to^* u \wedge z \to^* u$
Confluent (CR)	$(\forall x, y, z)\;x \to^* y \wedge x \to^* z \Rightarrow (\exists u)\;x \to^* u \wedge z \to^* u$

2   Many-Sorted First-Order Logic

Given a set of *sorts* S, a *(many-sorted) signature (with predicates)* $\Omega = (S, \Sigma, \Pi)$ consists of a set of sorts S, an $S^* \times S$-indexed family of sets $\Sigma = \{\Sigma_{w,s}\}_{(w,s) \in S^* \times S}$ containing *function symbols* $f \in \Sigma_{s_1 \cdots s_k, s}$, with a rank declaration $f : s_1 \cdots s_k \to s$ (constant symbols c have rank declaration $c : \lambda \to s$, where λ denotes the *empty* sequence), and an S^+-indexed family of sets $\Pi = \{\Pi_w\}_{w \in S^+}$ of ranked predicates $P : w$. Given an S-sorted set $\mathcal{X} = \{\mathcal{X}_s \mid s \in S\}$ of *mutually disjoint* sets of variables (which are also disjoint from Σ), the set $\mathcal{T}_\Sigma(\mathcal{X})_s$ of terms of sort s is the least set such that $\mathcal{X}_s \subseteq \mathcal{T}_\Sigma(\mathcal{X})_s$ and for each $f : s_1 \ldots s_k \to s$ and $t_i \in \mathcal{T}_\Sigma(\mathcal{X})_{s_i}$, $1 \le i \le k$, $f(t_1, \ldots, t_k) \in \mathcal{T}_\Sigma(\mathcal{X})_s$. If $\mathcal{X} = \emptyset$, we write $\mathcal{T}_\Sigma$ rather than $\mathcal{T}_\Sigma(\emptyset)$ for the set of *ground* terms. The set $\mathcal{T}_\Sigma(\mathcal{X})$ of *many-sorted terms* is $\mathcal{T}_\Sigma(\mathcal{X}) = \bigcup_{s \in S} \mathcal{T}_\Sigma(\mathcal{X})_s$. For $w = s_1 \cdots s_n \in S^+$, we write $\mathcal{T}_\Sigma(\mathcal{X})_w$ rather than $\mathcal{T}_\Sigma(\mathcal{X})_{s_1} \times \cdots \times \mathcal{T}_\Sigma(\mathcal{X})_{s_n}$ and even write $\boldsymbol{t} \in \mathcal{T}_\Sigma(\mathcal{X})_w$ rather than $t_i \in \mathcal{T}_\Sigma(\mathcal{X})_{s_i}$ for each $1 \le i \le n$. The formulas $\varphi \in Form_\Omega$ of a signature Ω are built up from atoms $P(\boldsymbol{t})$ with $P \in \Pi_w$ and $\boldsymbol{t} \in \mathcal{T}_\Sigma(\mathcal{X})_w$, logic connectives ($\neg$, $\wedge$, and also $\vee$, $\Rightarrow$,...) and quantifiers ($\forall$ and $\exists$) in the usual way. A closed formula, i.e., one whose variables are all universally or existentially quantified, is called a *sentence*. In the following, substitutions σ are assumed to be S-sorted mappings such that for all sorts $s \in S$, we have $\sigma(x) \in \mathcal{T}_\Sigma(\mathcal{X})_s$.

An *Ω-structure* $\mathcal{A}$ consists of (i) a family $\{\mathcal{A}_s \mid s \in S\}$ of sets called the *carriers* or *domains* together with (ii) a function $f_{w,s}^{\mathcal{A}} \in \mathcal{A}_w \to \mathcal{A}_s$ for each $f \in \Sigma_{w,s}$ ($\mathcal{A}_w$ is a one point set when $w = \lambda$ and hence $\mathcal{A}_w \to \mathcal{A}_s$ is isomorphic to $\mathcal{A}_s$), and (iii) an assignment to each $P \in \Pi_w$ of a subset $P_w^{\mathcal{A}} \subseteq \mathcal{A}_w$; if the identity predicate $\_ = \_ : ss$ is in Π_{ss}, then $(=)_{s\,s}^{\mathcal{A}} = \{(a, a) \mid a \in \mathcal{A}_s\}$, i.e., $\_ = \_ : ss$ is interpreted as the identity on $\mathcal{A}_s$.

Let $\mathcal{A}$ and $\mathcal{A}'$ be Ω-structures. An *Ω-homomorphism* $h : \mathcal{A} \to \mathcal{A}'$ is an S-sorted function $h = \{h_s : \mathcal{A}_s \to \mathcal{A}'_s \mid s \in S\}$ such that for each $f \in \Sigma_{w,s}$ and $P \in \Pi_w$ with $w = s_1, \ldots, s_k$, (i) $h_s(f_{w,s}^{\mathcal{A}}(a_1, \ldots, a_k)) = f_{w,s}^{\mathcal{A}'}(h_{s_1}(a_1), \ldots, h_{s_k}(a_k))$ and (ii) if $\boldsymbol{a} \in P_w^{\mathcal{A}}$, then $h(\boldsymbol{a}) \in P_w^{\mathcal{A}'}$. Given an S-sorted *valuation mapping* $\alpha : \mathcal{X} \to \mathcal{A}$, the evaluation mapping $[\_]_{\mathcal{A}}^\alpha : \mathcal{T}_\Sigma(\mathcal{X}) \to \mathcal{A}$ is the unique (S, Σ)-homomorphism extending α. Finally, $[\_]_{\mathcal{A}}^\alpha : Form_\Omega \to Bool$ is given by:

1. $[P(t_1, \ldots, t_n)]_{\mathcal{A}}^\alpha = true$ (with $P \in \Pi_w$) if and only if $([t_1]_{\mathcal{A}}^\alpha, \ldots, [t_n]_{\mathcal{A}}^\alpha) \in P_w^{\mathcal{A}}$;
2. $[\neg\varphi]_{\mathcal{A}}^\alpha = true$ if and only if $[\varphi]_{\mathcal{A}}^\alpha = false$;
3. $[\varphi \wedge \psi]_{\mathcal{A}}^\alpha = true$ if and only if $[\varphi]_{\mathcal{A}}^\alpha = true$ and $[\psi]_{\mathcal{A}}^\alpha = true$; and
4. $[(\forall x : s)\, \varphi]_{\mathcal{A}}^\alpha = true$ if and only if for all $a \in \mathcal{A}_s$, $[\varphi]_{\mathcal{A}}^{\alpha[x \mapsto a]} = true$.

A valuation $\alpha \in \mathcal{X} \to \mathcal{A}$ *satisfies* φ in $\mathcal{A}$ (written $\mathcal{A} \models \varphi\,[\alpha]$) if $[\varphi]_{\mathcal{A}}^\alpha = true$. We then say that φ is *satisfiable*. If $\mathcal{A} \models \varphi\,[\alpha]$ for *all* valuations α, we write $\mathcal{A} \models \varphi$ and say that $\mathcal{A}$ is a *model* of φ or that φ is *true* in $\mathcal{A}$. We say that $\mathcal{A}$ is *a model of a set of sentences* $\mathcal{S} \subseteq Form_\Omega$ (written $\mathcal{A} \models \mathcal{S}$) if for all $\varphi \in \mathcal{S}$, $\mathcal{A} \models \varphi$. Given a sentence φ, we write $\mathcal{S} \models \varphi$ iff $\mathcal{A} \models \varphi$ holds for *all models* $\mathcal{A}$ of $\mathcal{S}$.

3 Preservation of Many-Sorted First-Order Sentences

Every set $\mathcal{S}$ of ground atoms has an *initial model* $\mathcal{I}_{\mathcal{S}}$ (or just $\mathcal{I}$ if no confusion arises) which consists of the usual (many-sorted) *Herbrand Domain* of ground terms modulo the equivalence $\sim$ generated by the equations in $\mathcal{S}$. There is a unique homomorphism $h : \mathcal{I} \to \mathcal{A}$ from $\mathcal{I}$ to any model $\mathcal{A}$ of $\mathcal{S}$ [9, Sect. 3.2]. In the following, h refers to such a homomorphism. If $\mathcal{S}$ contains no equation, then $\mathcal{I}$ is the (many-sorted) *Least Herbrand Model* of $\mathcal{S}$ and $\mathcal{I}_s$ is $\mathcal{T}_{\Sigma_s}$ for each sort $s \in S$. In the following, we consider sentences in *prenex* form as follows:

$$(Q_1 x_1 : s_1) \cdots (Q_k x_k : s_k) \bigvee_{i=1}^{m} \bigwedge_{j=1}^{n_i} L_{ij} \tag{6}$$

where (i) for all $1 \leq i \leq m$ and $1 \leq j \leq n_i$, L_{ij} are *literals*, i.e., $L_{ij} = A_{ij}$ or $L_{ij} = \neg A_{ij}$ for some atom A_{ij} (in the first case, we say that L_{ij} is *positive*; otherwise, it is *negative*), (ii) $x_1, \ldots, x_k$ for some $k \geq 0$ are the variables occurring in those literals (of sorts $s_1, \ldots, s_k$, respectively), and (iii) $Q_1, \ldots, Q_k$ are universal/existential quantifiers. A sentence φ (equivalent to) (6) is said to be *positive* if all literals are.

Theorem 1. *Let Ω be a signature, $\mathcal{S}$ be a set of ground atoms, φ be a sentence (6), and $\mathcal{A}$ be a model of $\mathcal{S}$ such that (a) for all q, $1 \leq q \leq k$, if $Q_q = \forall$ then h_{s_q} is surjective[1] and (b) for all negative literals $L_{ij} = \neg P(\boldsymbol{t})$, with $P \in \Pi_w$, and substitutions σ, if $h(\sigma(\boldsymbol{t})) \in P^{\mathcal{A}}$ then $\sigma(\boldsymbol{t}) \in P^{\mathcal{I}}$. Then, $\mathcal{I}_{\mathcal{S}} \models \varphi \implies \mathcal{A} \models \varphi$.*

In order to achieve condition (b) in Theorem 1, given $P \in \Pi_w$, let $N(P) = \mathcal{I}_w - P^{\mathcal{I}}$ be the complement of the (Herbrand) interpretation of P. Let $\mathcal{N}(P) = \{\neg P(\boldsymbol{t}) \mid \boldsymbol{t} \in N(P)\}$ (cf. Reiter's *Closed World Assumption* [20]). In general, $\mathcal{N}(P)$ is infinite and incomputable. In some simple cases, though, we can provide a *finite description* of $\mathcal{N}(P)$ for the required predicates P (see Sect. 7).

Proposition 1. *Let Ω be a signature, $\mathcal{S}$ be a set of ground atoms, φ be a sentence (6), $\mathcal{A}$ be a model of $\mathcal{S}$, and $\mathcal{N} = \bigcup_{L_{ij} = \neg P(\boldsymbol{t})} \mathcal{N}(P)$ be such that $\mathcal{A} \models \mathcal{N}$. Let $L_{ij} = \neg P(\boldsymbol{t})$ be a negative literal and σ be a substitution. If $h(\sigma(\boldsymbol{t})) \in P^{\mathcal{A}}$, then $\sigma(\boldsymbol{t}) \in P^{\mathcal{I}}$.*

Consider a theory $\mathcal{S}$ and let $\mathcal{S}^{\vdash}$ be the set of ground atoms obtained as the *deductive closure* of $\mathcal{S}$, i.e., the set of *atoms* $P(t_1, \ldots, t_n)$ for each n-ary predicate symbol P and ground terms $t_1, \ldots, t_n$, such that $\mathcal{S} \vdash P(t_1, \ldots, t_n)$. The following result is the basis of the practical applications discussed in the following sections.

Corollary 1 (Semantic criterion). *Let Ω be a signature, $\mathcal{S}_0$ be a theory, $\mathcal{S} = \mathcal{S}_0^{\vdash}$, φ be a sentence (6), and $\mathcal{A}$ be a model of $\mathcal{S}_0$ such that (a) for all q, $1 \leq q \leq k$, if $Q_q = \forall$ then h_{s_q} is surjective and (b) for all negative literals $L_{ij} = \neg P(\boldsymbol{t})$, with $P \in \Pi_w$ and substitutions σ, if $h(\sigma(\boldsymbol{t})) \in P^{\mathcal{A}}$ then $\sigma(\boldsymbol{t}) \in P^{\mathcal{I}}$. If $\mathcal{A} \models \neg \varphi$, then $\mathcal{I}_{\mathcal{S}} \models \neg \varphi$.*

---

[1] A mapping $f : A \to B$ is *surjective* if for all $b \in B$ there is $a \in A$ such that $f(a) = b$.

In the following, we will not distinguish between theories $\mathcal{S}$ and their ground deductive closure $\mathcal{S}^{\vdash}$; we rather use $\mathcal{S}$ in both cases.

Remark 1 (Proofs by satisfiability). We can prove an arbitrary sentence φ valid in $\mathcal{I}_{\mathcal{S}}$ by *satisfiability* in some model $\mathcal{A}$ of $\mathcal{S}$. First define $\overline{\varphi}$ as the negation $\neg\varphi$ of φ. Then, find an appropriate structure $\mathcal{A}$ satisfying (a) and (b) (with regard to $\overline{\varphi}$) and such that $\mathcal{A} \models \mathcal{S} \cup \{\neg\overline{\varphi}\}$. By Corollary 1, $\mathcal{I} \models \neg\overline{\varphi}$ holds. Since $\neg\overline{\varphi}$ is equivalent to φ, $\mathcal{I} \models \varphi$ holds.

Models $\mathcal{A}$ to be used in Corollary 1 can be automatically generated from the theory $\mathcal{S}$ and sentence φ by using a tool like AGES [10] or Mace4. In the following section, we investigate how to ensure *surjectivity* when required in Corollary 1.

4   Surjective Homomorphisms

Given $\Omega = (S, \Sigma, \Pi)$, $s \in S$ and $T \subseteq \mathcal{T}_{\Sigma s}$, consider the following sentences:

$$(\forall x : s) \bigvee_{t \in T} x = t \tag{7}$$

$$\bigwedge_{t,u \in T, t \neq u} \neg(t = u) \tag{8}$$

In the following, we write $(7)_s$ to make sort s referred in (7) explicit. We do the same in similar formulas below.

Proposition 2. *Let Ω be a signature, $\mathcal{S}$ be a theory, $\mathcal{A}$ be a model of $\mathcal{S}$, $s \in S$, and $T \subseteq \mathcal{T}_{\Sigma s}$. (a) If $T \neq \emptyset$ and $\mathcal{A} \models (7)_s$, then h_s is surjective and $|\mathcal{A}_s| \leq |T|$. (b) If $\mathcal{A}_s \neq \emptyset$ and $\mathcal{A} \models (8)$, then $|\mathcal{A}_s| \geq |T|$.*

In view of Proposition 2(a), denote $(7)_s$ as $\mathsf{SuH}_s^T(\Omega)$ (or just SuH_s^T or SuH^T if no confusion arises). Whenever T is finite, Proposition 2(a) imposes that the interpretation domain $\mathcal{A}_s$ for sort s is finite. This is appropriate for tools like Mace4 which generate structures with finite domains only. However, the *choice* of T in Proposition 2, when used together with a theory $\mathcal{S}$ imposing further requirements on symbols, can be crucial for Corollary 1 to succeed. Restricting the attention to finite domains can also be a drawback. In the following, we investigate a different approach which avoids any choice of terms T and is valid for infinite structures as well. Consider the following sentence:

$$(\forall x : s)(\exists n : Nat)\ term_s(x, n) \tag{9}$$

where Nat is a new sort, to be interpreted as the set $\mathbb{N}$ of natural numbers, and $term_s : s\ Nat$ is a new predicate for each $s \in S$. The intended meaning of $(9)_s$ is

that, for all $x \in \mathcal{A}_s$, there is $t \in \mathcal{T}_{\Sigma s}$ of height at most n such that $x = t^{\mathcal{A}}$. We substantiate this, for each sort $s \in S$, by means of two (families of) formulas:

$$(\forall x : s)(\forall n : Nat) \; term_s(x, 0) \Rightarrow \bigvee_{c \in \Sigma_{\lambda,s}} x = c \tag{10}$$

$$(\forall x : s)(\forall n : Nat)(\exists m : Nat) \; (n > 0 \wedge term_s(x, n)) \Rightarrow$$

$$n > m \wedge \left(term_s(x, m) \vee \bigvee_{\substack{f \in \Sigma_{w,s} \\ w \in S^+}} (\exists \boldsymbol{y} : w) \left(x = f(\boldsymbol{y}) \wedge \bigwedge_{s_i \in w} terms_{s_i}(y_i, m) \right) \right) \tag{11}$$

Thus, by $(10)_s$, values x satisfying $term_s(x, 0)$ will be represented by some constant symbol c of sort s. Similarly, by $(11)_s$, values x satisfying $term_s(x, n)$ for some $n > 0$ will be represented by some ground term s of height m for some $m < n$, or by a term $t = f(t_1, \ldots, t_k)$, where f has rank $w \to s$ for some $w \in S^+$ and $t_1, \ldots, t_k$ have height m at most.

The set $K(s)$ of *s-relevant* sorts is the least set satisfying: (i) $s \in K(s)$ and (ii) if $f \in \Sigma_{s_1 \cdots s_k, s'}$ and $s' \in K(s)$, then $\{s_1, \ldots, s_n\} \subseteq K(s)$. Let $\Omega_{Nat,s} = (S_{Nat}, \Sigma_{Nat}, \Pi_{Nat,K(s)})$ be an *extension* of Ω where $S_{Nat} = S \cup \{Nat\}$, Σ_{Nat} extends Σ with a new constant $0 : \lambda \to Nat$, and $\Pi_{Nat,K(s)}$ extends Π with $> : Nat \; Nat$ and a predicate $term_{s'} : s' \; Nat$ for each $s' \in K(s)$. We let

$$\mathsf{SuH}_s = \{(9)_{s'}, (10)_{s'}, (11)_{s'} \mid s' \in K(s)\} \tag{12}$$

Proposition 3. *Let Ω be a signature, $\mathcal{S}$ be a theory, $s \in S$, and $\mathcal{A}$ be an $\Omega_{Nat,s}$-structure which is a model of $\mathcal{S}$. Assume that $\mathcal{A}_{Nat} = \mathbb{N}$, $0^{\mathcal{A}} = 0$, and $\mathsf{m} >^{\mathcal{A}} \mathsf{n} \Leftrightarrow \mathsf{m} >_{\mathbb{N}} \mathsf{n}$ for all $\mathsf{m}, \mathsf{n} \in \mathcal{A}_{Nat}$. If $\mathcal{A} \models \mathsf{SuH}_s$, then $h_{s'}$ is surjective for all $s' \in K(s)$.*

Given an extension Ω' of a signature Ω, every Ω'-structure $\mathcal{A}'$ defines an Ω-structure $\mathcal{A}$: just take $\mathcal{A}_s = \mathcal{A}'_s$ for all $s \in S$, and then $f^{\mathcal{A}}_{w,s} = f^{\mathcal{A}'}_{w,s}$ and $P^{\mathcal{A}}_w = P^{\mathcal{A}'}_w$ for all $w \in S^*$, $s \in S$, $f \in \Sigma_{w,s}$, and $P \in \Pi_w$. Thus, Proposition 3 is used to guarantee surjectivity of $h : \mathcal{T}_{\Sigma s} \to \mathcal{A}_{s'}$, rather than $h : \mathcal{T}_{\Sigma_{Nat} s'} \to \mathcal{A}_{s'}$.

5 Examples of Application with Positive Sentences

In this section we exemplify the use of Corollary 1 together with the approach in Sect. 4 to deal with *positive* sentences (6), i.e., all literals are positive.

5.1 Complete Definedness and Commutativity

Consider the following Maude specification (hopefully self-explained, but see [5]) for the arithmetic operations in Example 1 together with function `head`, which returns the head of a list of natural numbers:

```
mod ExAddMulHead is
  sorts N LN . *** Sorts for numbers and lists of numbers
  op Z : -> N .          op suc : N -> N .   ops add mul : N N -> N .
  op head : LN -> N .    op nil : -> LN .    op cons : N LN -> LN .
  vars x y : N .         var xs : LN .
  rl add(Z,x) => x .     rl add(suc(x),y) => suc(add(x,y)) .
  rl mul(Z,x) => Z .     rl mul(suc(x),y) => add(y,mul(x,y)) .
  rl head(cons(x,xs)) => x .
endm
```

(1) *Complete definedness.* We claim `head` to be completely defined as follows:

$$(\forall xs : \text{LN})(\exists x : \text{N}) \; \text{head}(xs) \to x \tag{13}$$

We *disprove* (13) by using Corollary 1. Due to the universal quantification of xs in (13), we need to ensure that $h_{\text{LN}} : \mathcal{T}_{\Sigma\text{LN}} \to \mathcal{A}_{\text{LN}}$ is surjective for any structure $\mathcal{A}$ we may use. We use Proposition 3. Since $K(\text{LN}) = \{\text{N}, \text{LN}\}$ due to `cons`, whose first argument is of sort N, SuH_{LN} consists of the following sentences:

$$(\forall x : \text{N})(\exists n : Nat) \; term_{\text{N}}(x, n) \tag{9}_{\text{N}}$$
$$(\forall x : \text{N}) \; term_{\text{N}}(x, 0) \Rightarrow x = \text{Z} \tag{10}_{\text{N}}$$
$$(\forall x : \text{N})(\forall n : Nat)(\exists m : Nat)(\exists y : \text{N})(\exists z : \text{N})(\exists ys : \text{LN}) \tag{11}_{\text{N}}$$
$$n > 0 \wedge term_{\text{N}}(x, n) \Rightarrow n > m \wedge [term_{\text{N}}(x, m) \vee$$
$$(term_{\text{N}}(y, m) \wedge term_{\text{N}}(z, m) \wedge term_{\text{LN}}(ys, m) \wedge$$
$$(x = \text{suc}(y) \vee x = \text{add}(y, z) \vee x = \text{mul}(y, z) \vee x = \text{head}(ys)))]$$
$$(\forall xs : \text{LN})(\exists n : Nat) \; term_{\text{LN}}(xs, n) \tag{9}_{\text{LN}}$$
$$(\forall xs : \text{LN}) \; term_{\text{LN}}(xs, 0) \Rightarrow xs = \text{nil} \tag{10}_{\text{LN}}$$
$$(\forall xs : \text{LN})(\forall n : Nat)(\exists m : Nat)(\exists y : \text{N})(\exists ys : \text{LN}) \tag{11}_{\text{LN}}$$
$$n > 0 \wedge term_{\text{N}}(x, n) \Rightarrow n > m \wedge [term_{\text{N}}(x, m) \vee$$
$$(term_{\text{N}}(y, m) \wedge term_{\text{LN}}(ys, m) \wedge xs = \text{cons}(y, ys))]$$

We obtain a model $\mathcal{A}$ of $\overline{\text{ExAddMulHead}} \cup \text{SuH}_{\text{LN}} \cup \{\neg(13)\}$ with AGES. Sorts are interpreted as follows: $\mathcal{A}_{\text{N}} = \mathcal{A}_{\text{LN}} = \{-1, 0\}$ and $\mathcal{A}_{Nat} = \mathbb{N}$. For function symbols:

$$\text{Z}^{\mathcal{A}} = -1 \qquad \text{nil}^{\mathcal{A}} = 0 \qquad \text{suc}^{\mathcal{A}}(x) = x \qquad \text{add}^{\mathcal{A}}(x, y) = 0$$
$$\text{mul}^{\mathcal{A}}(x, y) = 0 \quad \text{cons}^{\mathcal{A}}(x, xs) = -1 \quad \text{head}^{\mathcal{A}}(xs) = -xs - 1$$

For predicates, $x \to_{\text{N}}^{\mathcal{A}} y \Leftrightarrow x \geq y \wedge x \geq 0$, $x \to_{\text{LN}}^{\mathcal{A}} y \Leftrightarrow x = y = -1$, and both $x(\to_{\text{N}}^*)^{\mathcal{A}} y$ and $x(\to_{\text{LN}}^*)^{\mathcal{A}} y$ are *true*. We can check surjectivity of $h_s : \mathcal{T}_{\Sigma s} \to \mathcal{A}_s$ (for $s \in \{\text{N}, \text{LN}\}$). For instance, we have:

$$[\text{Z}]_{\mathcal{A}} = -1 \quad [\text{add}(\text{Z}, \text{Z})]_{\mathcal{A}} = 0 \quad \text{for sort N}$$
$$[\text{cons}(\text{Z}, \text{nil})]_{\mathcal{A}} = -1 \quad [\text{nil}]_{\mathcal{A}} = 0 \quad \text{for sort LN}$$

(2) *Commutativity.* It is well-known that both add and mul as defined by the rules of $\mathcal{R}$ in Example 1 are *commutative* on ground terms, i.e., for all ground terms s and t, $\mathsf{add}(s,t) =_{\mathcal{R}} \mathsf{add}(t,s)$ and $\mathsf{mul}(s,t) =_{\mathcal{R}} \mathsf{mul}(t,s)$, where $=_{\mathcal{R}}$ is the equational theory induced by the rules $\ell \to r$ in $\mathcal{R}$ treated as equations $\ell = r$. Actually, by using Birkhoff's theorem and the fact that $\mathcal{R}$ is *confluent*, we can rephrase commutativity of add as *joinability* as follows:

$$(\forall x)(\forall y)(\exists z) \; \mathsf{add}(x,y) \to^* z \land \mathsf{add}(y,x) \to^* z \tag{14}$$

Remark 2. Proving commutativity of add and mul when defined by $\mathcal{R}$ in Example 1 by using Corollary 1 is possible (see Remark 1) but unlikely. We should first define $\overline{\varphi}$ as $\neg(14)$, i.e., $\overline{\varphi}$ is

$$(\exists x)(\exists y)(\forall z) \; \neg(\mathsf{add}(x,y) \to^* z) \lor \neg(\mathsf{add}(y,x) \to^* z) \tag{15}$$

Since (15) contains two negative literals, Corollary 1 requires the use of $\mathcal{N}(\to^*)$.

Since head is not completely defined, add and mul are *not* commutative in ExAddMulHead. We prove this fact by disproving the sorted version of (14), i.e.,

$$(\forall x : \mathtt{N})(\forall y : \mathtt{N})(\exists z : \mathtt{N}) \; \mathsf{add}(x,y) \to^* z \land \mathsf{add}(y,x) \to^* z \tag{16}$$

Due to the universal quantification of x and y in (16), we need to ensure that $h_{\mathtt{N}} : \mathcal{T}_{\Sigma\mathtt{N}} \to \mathcal{A}_{\mathtt{N}}$ is surjective. Since $K(\mathtt{N}) = \{\mathtt{N}, \mathtt{LN}\}$ due to head, we have $\mathsf{SuH}_{\mathtt{N}} = \mathsf{SuH}_{\mathtt{LN}}$ as above. AGES obtain a model $\mathcal{A}$ of $\overline{\mathtt{ExAddMulHead}} \cup \mathsf{SuH}_{\mathtt{N}} \cup \{\neg(16)\}$ as follows: $\mathcal{A}_{\mathtt{N}} = \{0,1\}$, $\mathcal{A}_{\mathtt{LN}} = \{-1,0\}$ and $\mathcal{A}_{Nat} = \mathbb{N}$. Also,

$$\begin{array}{cccc}
\mathtt{Z}^{\mathcal{A}} = 1 & \mathsf{nil}^{\mathcal{A}} = -1 & \mathsf{suc}^{\mathcal{A}}(x) = x & \mathsf{add}^{\mathcal{A}}(x,y) = y \\
\mathsf{mul}^{\mathcal{A}}(x,y) = x & \mathsf{cons}^{\mathcal{A}}(x,xs) = x-1 & \mathsf{head}^{\mathcal{A}}(xs) = xs+1 & \\
x \to_{\mathtt{N}}^{\mathcal{A}} y \Leftrightarrow x = y & x(\to_{\mathtt{N}}^*)^{\mathcal{A}} y \Leftrightarrow x = y & x \to_{\mathtt{LN}}^{\mathcal{A}} y \Leftrightarrow x = y & x(\to_{\mathtt{LN}}^*)^{\mathcal{A}} y \Leftrightarrow true
\end{array}$$

5.2 Top-Termination

A TRS $\mathcal{R}$ is *top-terminating* if no infinitary reduction sequence performs infinitely many rewrites at topmost position Λ [7]. From a computational point of view, top-termination is important in the semantic description of lazy languages as it is an important ingredient to guarantee that every initial expression has an infinite normal form [7,8]. Accordingly, given a *dummy* sort $\mathtt{S}$, the *negation* of

$$(\exists x : \mathtt{S})(\forall n \in \mathbb{N})(\exists y : \mathtt{S}) \; x(\to^* \circ \xrightarrow{\Lambda})^n y \tag{17}$$

(which claims for the existence of a term with infinitely many rewriting steps at top) captures top-termination. We introduce a new predicate $\to_{\star,\Lambda}$ for the composition $\to^* \circ \xrightarrow{\Lambda}$ of the many-step rewriting relation $\to^*$ (defined as usual, i.e., by the whole theory $\overline{\mathcal{R}}$ associated to $\mathcal{R}$) and topmost rewriting $\xrightarrow{\Lambda}$ defined

by a theory $\mathcal{R}_\Lambda = \{(\forall \boldsymbol{x} : \mathsf{S}) \; \ell \xrightarrow{\Lambda} r \mid \ell \rightarrow r \in \mathcal{R}\}$. Sequences $s \rightarrow^n_{\star,\Lambda} t$ meaning that $s \rightarrow_{\star,\Lambda}$-reduces into t in $n + 1 \rightarrow_{\star,\Lambda}$-steps are defined as follows:

$$(\forall x, y, z : \mathsf{S}) \;\; x \rightarrow^* y \wedge y \xrightarrow{\Lambda} z \Rightarrow x \rightarrow^0_{\star,\Lambda} z \tag{18}$$

$$(\forall x, y, z : \mathsf{S})(\forall n \in \mathbb{N}) \;\; x \rightarrow^0_{\star,\Lambda} y \wedge y \rightarrow^n_{\star,\Lambda} z \Rightarrow x \rightarrow^{n+1}_{\star,\Lambda} z \tag{19}$$

Overall, the sentence φ to be disproved is:

$$(\exists x : \mathsf{S})(\forall n : Nat)(\exists y : \mathsf{S}) \; x \rightarrow^n_{\star,\Lambda} y \tag{20}$$

Remark 3. We use $\mathbb{N}$ in (17) but *Nat* in (20). Indeed, (17) is *not* a valid sentence because $\mathbb{N}$ is not first-order axiomatizable, see, e.g. [11, Sect. 2.2]. This is consistent with the well-known fact that termination (or top-termination) can*not* be encoded in first-order logic [22, Sect. 5.1.4]. We can use (20) together with Corollary 1 provided that *Nat* is interpreted as $\mathbb{N}$. This is possible with AGES.

Example 2. Consider the following (nonterminating) TRS $\mathcal{R}$ [8, Sect. 9.5]:

$$\mathsf{non} \rightarrow \mathsf{f}(\mathsf{g}, \mathsf{f}(\mathsf{non}, \mathsf{g})) \tag{21}$$

$$\mathsf{g} \rightarrow \mathsf{a} \tag{22}$$

$$\mathsf{f}(\mathsf{a}, x) \rightarrow \mathsf{a} \tag{23}$$

$$\mathsf{f}(\mathsf{b}, \mathsf{b}) \rightarrow \mathsf{b} \tag{24}$$

$$\mathsf{f}(\mathsf{b}, \mathsf{a}) \rightarrow \mathsf{b} \tag{25}$$

The associated theory $\mathcal{R}_{topT}$ is $\mathcal{R}_{topT} = \overline{\mathcal{R}} \cup \mathcal{R}_\Lambda \cup \{(18), (19)\}$, where $\mathcal{R}_\Lambda$ is

$$\mathsf{non} \xrightarrow{\Lambda} \mathsf{f}(\mathsf{g}, \mathsf{f}(\mathsf{non}, \mathsf{g})) \tag{26}$$

$$\mathsf{g} \xrightarrow{\Lambda} \mathsf{a} \tag{27}$$

$$(\forall x : \mathsf{S}) \; \mathsf{f}(\mathsf{b}, x) \xrightarrow{\Lambda} \mathsf{b} \tag{28}$$

$$\mathsf{f}(\mathsf{b}, \mathsf{b}) \xrightarrow{\Lambda} \mathsf{b} \tag{29}$$

$$\mathsf{f}(\mathsf{b}, \mathsf{a}) \xrightarrow{\Lambda} \mathsf{b} \tag{30}$$

Note that (20) only requires that the homomorphism mapping terms of sort *Nat* to $\mathbb{N}$ is surjective, which is automatically achieved by AGES. The structure $\mathcal{A}$ with $\mathcal{A}_\mathsf{S} = \{-1, 0, 1\}$, $\mathcal{A}_{Nat} = \mathbb{N}$, function symbols interpreted by: $\mathsf{a}^\mathcal{A} = 1$, $\mathsf{b}^\mathcal{A} = 1$, $\mathsf{g}^\mathcal{A} = 0$, $\mathsf{non}^\mathcal{A} = -1$, and $\mathsf{f}^\mathcal{A}(x) = 0$; and predicate symbols as follows:

$$x \rightarrow^\mathcal{A} y \Leftrightarrow y \geq x \wedge x + y \geq -1 \qquad x(\rightarrow^*)^\mathcal{A} y \Leftrightarrow y \geq x$$
$$x(\xrightarrow{\Lambda})^\mathcal{A} y \Leftrightarrow y > x \qquad\qquad\quad x(\rightarrow^n_{\star,\Lambda})^\mathcal{A} y \Leftrightarrow y > x + n$$

is a model of $\mathcal{R}_{topT} \cup \{\neg(20)\}$ and proves top-termination of $\mathcal{R}$.

6   Refutation Witnesses

In logic, a *witness* for an existentially quantified sentence $(\exists x)\varphi(x)$ is a specific value b to be substituted by x in $\varphi(x)$ so that $\varphi(b)$ is true (see, e.g., [2, p. 81]). Similarly, we can think of a value b such that $\neg\varphi(b)$ holds as a witness of $(\exists x)\neg\varphi(x)$ or as a *refutation witness* for $(\forall x)\varphi(x)$; we can also think of b as a *counterexample* to $(\forall x)\varphi(x)$ [13, p. 284]. Note, however, that witnesses that are given as values b belonging to an *interpretation domain* $\mathcal{A}$ can be meaningless for the user who is acquainted with the first-order language Ω but not so much with abstract values from $\mathcal{A}$ (which is often automatically synthesized by using some tool). Users can be happier to deal with *terms* t which are somehow connected to witnesses b by a homomorphism, so that $t^{\mathcal{A}} = b$. Corollary 1 permits a refutation of φ by finding a model $\mathcal{A}$ of $\neg\varphi$ to conclude that $\mathcal{I} \models \neg\varphi$. We want to obtain instances of φ to better understand unsatisfiability of φ. In this section we investigate this problem.

The negation $\neg(6)$ of (6), i.e., of $(Q_1 x_1 : s_1) \cdots (Q_k x_k : s_k) \bigvee_{i=1}^{m} \bigwedge_{j=1}^{n_i} L_{ij}$ is

$$(\overline{Q}_1 x_1 : s_1) \cdots (\overline{Q}_k x_k : s_k) \bigwedge_{i=1}^{m} \bigvee_{j=1}^{n_i} \neg L_{ij}(x_1, \ldots, x_k) \tag{31}$$

where $\overline{Q}_i$ is $\forall$ whenever Q_i is $\exists$ and $\overline{Q}_i$ is $\exists$ whenever Q_i is $\forall$. We assume $\eta \leq k$ universal quantifiers in (31) with indices $U = \{v_1, \ldots, v_\eta\} \subseteq \{1, \ldots, k\}$ and hence $k - \eta$ existential quantifiers with indices $E = \{\epsilon_1, \ldots, \epsilon_{k-\eta}\} = \{1, \ldots, k\} - U$. In the following $\overline{\eta}$ denotes $k - \eta$. For each $\epsilon \in E$, we let $U_\epsilon = \{v \in U \mid v < \epsilon\}$ be the (possibly empty) set of indices of universally quantified variables in (31) occurring before x_ϵ in the quantification prefix of (31). Let $\eta_\epsilon = |U_\epsilon|$. Note that $U_{\epsilon_1} \subseteq U_{\epsilon_2} \subseteq \cdots \subseteq U_{\epsilon_{\overline{\eta}}}$. Let $U_\exists$ be the set of indices of universally quantified variables occurring *before* some existentially quantified variable in the quantification prefix of (31). Note that $U_\exists$ is empty whenever $v_1 > \epsilon_{k-\eta}$ (no existential quantification after a universal quantification); otherwise, $U_\exists = \{v_1, \ldots, v_\exists\}$ for some $v_\exists \leq v_\eta$. Accordingly, $U_\forall = U - U_\exists = \{\epsilon_{\overline{\eta}}+1, \ldots, k\}$ is the set of indices of universally quantified variables occurring *after* all existentially quantified variables in the quantification prefix of (31). Note that $U_\forall$ is empty whenever $\epsilon_1 > v_\eta$ (no universal quantification after an existential quantification).

Most theorem provers transform sentences into universally quantified formulas by Skolemization (see, e.g., [12]). Thus, if $k > \eta$, i.e., (31) contains existential quantifiers, we need to introduce *Skolem function symbols* $sk_\epsilon : w_\epsilon \to s_\epsilon$ for each $\epsilon \in E$, where w_ϵ is the (possibly empty) sequence of η_ϵ sorts indexed by U_ϵ. Note that sk_ϵ is a *constant* if $\eta_\epsilon = 0$. The *Skolem normal form* of (31) is

$$(\forall x_{v_1} : s_{v_1}) \cdots (\forall x_{v_\eta} : s_{v_\eta}) \bigwedge_{i=1}^{m} \bigvee_{j=1}^{n_i} \neg L_{ij}(e_1, \ldots, e_k) \tag{32}$$

where for all $1 \leq q \leq k$, (i) $e_q \equiv x_q$ if $q \in U$ and (ii) $e_q \equiv sk_q(\boldsymbol{x}_{\eta_q})$ if $q \in E$, where $\boldsymbol{x}_{\eta_q}$ is the sequence of variables $x_{v_1}, \ldots, x_{v_{\eta_q}}$. If $E \neq \emptyset$ (i.e., (31) and

(32) differ), then (32) is a sentence of an *extended* signature $\Omega^{sk} = (S, \Sigma^{sk}, \Pi)$ where Σ^{sk} extends Σ with skolem functions. Since (32) logically implies (31) [2, Sect. 19.2], every model $\mathcal{A}$ of (33) is a model of (32) as well.

Definition 1 (Set of refutation witnesses). *Using the notation developed in the previous paragraphs, let $\mathcal{A}$ be an Ω^{sk}-structure such that h_{s_q} is surjective for all $q \in U_\exists \cup E$. The Ω^{sk}-sentence (32) is given a set of refutation witnesses Φ consisting of Ω-sentences ϕ_α for each valuation α of the variables $x_{v_1}, \ldots, x_{v_\exists}$ indexed by $U_\exists$; each ϕ_α is (nondeterministically) defined as follows:*

$$(\forall x_{\epsilon_{\overline{\eta}}+1} : s_{\epsilon_{\overline{\eta}}+1}) \cdots (\forall x_k : s_k) \bigwedge_{i=1}^{m} \bigvee_{j=1}^{n_i} \neg L_{ij}(e'_1, \ldots, e'_k) \tag{33}$$

where for all $1 \leq q \leq k$, (i) $e'_q \equiv x_q$ if $q \in U_\forall$ and (ii) $e'_q \equiv t$ if $q \in U_\exists \cup E$ and $t \in \mathcal{T}_{\Sigma s_q}$ is such that $[t]_\mathcal{A} = [e_q]_\mathcal{A}^\alpha$.

Note that, in Definition 1 we could emphfail to find the necessary terms $t \in \mathcal{T}_{\Sigma s_q}$ if h_{s_q} is *not* surjective. Note also that, whenever E is empty, Φ is a singleton consisting of (33) which coincides with (32). We have the following:

Proposition 4. *For every Ω^{sk}-structure $\mathcal{A}$, $\mathcal{A} \models$ (33) if and only if $\mathcal{A} \models \Phi$.*

Refutation witnesses are built from symbols in the original signature Ω only. We can use them as more intuitive *counterexamples* to the refuted property φ.

Proposition 5. *Let Ω be a signature, $\mathcal{S}$ be a theory, φ be a sentence (6), and $\mathcal{A}$ be a model of $\mathcal{S}$ such that for all negative literals $L_{ij} = \neg P(\boldsymbol{t})$ with $P \in \Pi_w$ and substitutions σ, if $h(\sigma(\boldsymbol{t})) \in P^\mathcal{A}$ then $\sigma(\boldsymbol{t}) \in P^\mathcal{I}$. For all $\phi \in \Phi$, $\mathcal{I} \models \phi$.*

Corollary 2. *If (6) is positive, then for all refutation witnesses $\phi \in \Phi$, $\mathcal{I} \models \phi$.*

Example 3. Consider `ExAddMulHead` in Sect. 5. The *refutation* of (13) using AGES actually proceeds by skolemization of the negation of (13), i.e., of

$$(\exists xs : \texttt{LN})(\forall x : \texttt{N}) \ \neg(\texttt{head}(xs) \to x) \tag{34}$$

With regard to (34), we have $E = \{1\}$, $U_\exists = \emptyset$ and $U_\forall = \{2\}$, where 1 and 2 refer to variables xs and x, respectively. Accordingly, $v_\exists = 0$. The only sort involved in the variables indexed by $U_\exists \cup E$ is `LN`. Since variables of sort `LN` are universally quantified in (13), the application of Corollary 1 in Sect. 5 already required surjectivity of $h_{\texttt{LN}}$. The *Skolem normal form* of (34) is:

$$(\forall x : \texttt{N}) \ \neg(\texttt{head}(\texttt{sk}_{xs}) \to x) \tag{35}$$

where $\texttt{sk}_{xs}$ is a new constant of sort `LN`. The structure $\mathcal{A}$ computed by AGES is actually a model of $\overline{\mathcal{R}} \cup \texttt{SuH}_{\texttt{LN}} \cup \{(36)\}$, for $\texttt{SuH}_{\texttt{LN}}$ in Sect. 5. For $\texttt{sk}_{xs}$, we have $\texttt{sk}_{xs}^\mathcal{A} = 0$. There is a single (empty) valuation α of variables indexed by $U_\exists$ (which is empty). Hence, $\Phi = \{\phi_\alpha\}$ is a singleton. According to Definition 1, since $[\texttt{nil}]_\mathcal{A} = 0 = [\texttt{sk}_{xs}]_\mathcal{A}$, the following sentence could be associated to the refutation witness ϕ_α: $(\forall x : \texttt{N}) \ \neg(\texttt{head}(\texttt{nil}) \to x)$.

Example 4. With regard to the computation of refutation witnesses for $\mathcal{R}$ in Example 2, we start with the negation of (17), i.e.,

$$(\forall x : \mathsf{S})(\exists n : Nat)(\forall y : \mathsf{S}) \ \neg(x(\to^* \circ \xrightarrow{\Lambda})^n y) \tag{36}$$

We have $E = \{2\}$, $U_\exists = \{1\}$ and $U_\forall = \{3\}$. The Skolem normal form of (36) is

$$(\forall x : \mathsf{S})(\forall y : \mathsf{S}) \ \neg(x(\to^* \circ \xrightarrow{\Lambda})^{\mathsf{sk}_n(x)} y) \tag{37}$$

where $\mathsf{sk}_n : \mathsf{S} \to Nat$ is a new (monadic) function symbol. Since the sorts for variables indexed by $U_\exists \cup E$ are S and Nat, we require surjectivity of h_S and h_{Nat}. This is achieved by using SuH_S and interpreting Nat as $\mathbb{N}$ as done in AGES. The structure $\mathcal{A}$ in Example 2 is a model of $\mathcal{R}_{topT} \cup \mathsf{SuH}_\mathsf{S} \cup \{(38)\}$. The interpretation obtained for sk_n is

$$\mathsf{sk}_n^{\mathcal{A}}(x) = 1 - x$$

Now we can compute refutation witnesses for (37). Since $U_\epsilon = \{1\}$ is a singleton whose index refers to a variable x of sort S and $\mathcal{A}_\mathsf{S} = \{-1, 0, 1\}$, we have to deal with three valuation functions for the only variable x to be considered:

$$\alpha_{-1}(x) = -1 \qquad \alpha_0(x) = 0 \qquad \alpha_1(x) = 1$$

We have $\Phi = \{\phi_{\alpha_{-1}}, \phi_{\alpha_0}, \phi_{\alpha_1}\}$, where $\phi_{\alpha_{-1}}$ is $(\forall y : \mathsf{S})\neg(\mathsf{non}(\to^* \circ \xrightarrow{\Lambda})^2 y)$, ϕ_{α_0} is $(\forall y : \mathsf{S})\neg(\mathsf{g}(\to^* \circ \xrightarrow{\Lambda})^1 y)$, and ϕ_{α_1} is $(\forall y : \mathsf{S})\neg(\mathsf{a}(\to^* \circ \xrightarrow{\Lambda})^0 y)$.

Note that, since $\mathsf{f}^{\mathcal{A}}(x) = 0$, we could also write ϕ_{α_0} as $(\forall y : \mathsf{S}) \ \neg(\mathsf{f}(t)(\to^* \circ \xrightarrow{\Lambda})^1 y)$ for *every ground term t*. This gives additional, complementary information.

7 Example of Application with General Sentences

Consider a well-known example of a *locally confluent* but *nonconfluent* TRS $\mathcal{R}$:

$$\mathsf{b} \to \mathsf{a} \qquad \mathsf{b} \to \mathsf{c} \qquad \mathsf{c} \to \mathsf{b} \qquad \mathsf{c} \to \mathsf{d}$$

Example 5 (Local confluence of $\mathcal{R}$). Local confluence corresponds to φ_{WCR} in Table 1. As explained in Remark 1, we start with $\overline{\varphi}_{WCR} = \neg\varphi_{WCR}$ i.e.,

$$(\exists x, y, z : \mathsf{S})(\forall u : \mathsf{S}) \ (x \to y \wedge x \to z \wedge \neg(x \to^* u)) \vee (x \to y \wedge x \to z \wedge \neg(z \to^* u)) \tag{38}$$

Due to the universal quantifier, $h_\mathsf{S} : \mathcal{T}_{\Sigma\mathsf{S}} \to \mathcal{A}_\mathsf{S}$ must be surjective. We can achieve this by adding the following sentence $\mathsf{SuH}_\mathsf{S}^T$ for $T = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\}$:

$$(\forall x : \mathsf{S}) \ x = \mathsf{a} \vee x = \mathsf{b} \vee x = \mathsf{c} \vee x = \mathsf{d} \tag{39}$$

Due to the negative literals $\neg(x \to^* u)$ and $\neg(z \to^* u)$, we consider $\mathcal{N}$, representing the forbidden many-step rewriting steps, explicitly given by:

$$\mathcal{N} = \{ \neg(\mathsf{a} \to^* \mathsf{b}), \neg(\mathsf{a} \to^* \mathsf{c}), \neg(\mathsf{a} \to^* \mathsf{d}), \neg(\mathsf{d} \to^* \mathsf{a}), \neg(\mathsf{d} \to^* \mathsf{b}), \neg(\mathsf{d} \to^* \mathsf{c}) \}$$

We apply Corollary 1 to prove that $\neg\overline{\varphi}_{WCR}$ (i.e., φ_{WCR}) holds by obtaining a model of $\overline{\mathcal{R}} \cup \mathsf{SuH}_\mathsf{s}^T \cup \mathcal{N} \cup \{\varphi_{WCR}\}$ with Mace4.[2] The structure has domain $\mathcal{A}_\mathsf{s} = \{0, 1, 2, 3\}$; constants are interpreted as follows: $\mathsf{a}^\mathcal{A} = 0$, $\mathsf{b}^\mathcal{A} = 1$, $\mathsf{c}^\mathcal{A} = 3$, and $\mathsf{d}^\mathcal{A} = 2$. With regard to predicate symbols, we have:

$$x \to^\mathcal{A} y = \{(1, 0), (1, 3), (3, 1), (3, 2)\} \qquad x(\to^*)^\mathcal{A} y = \{(1, x), (3, x) \mid x \in \mathcal{A}_\mathsf{s}\}$$

This proves $\mathcal{R}$ locally confluent.

Example 6 (Nonconfluence of $\mathcal{R}$). In order to *disprove* confluence of $\mathcal{R}$, which is represented by φ_{CR} in Table 1, we first write φ_{CR} in the form (6), i.e.,

$$(\forall x, y, z : \mathsf{S})(\exists u : \mathsf{S}) \ \neg(x \to^* y) \vee \neg(x \to^* z) \vee (y \to^* u \wedge z \to^* u) \qquad (40)$$

Due to the universal quantification and negative literals, we use $\mathsf{SuH}_\mathsf{s}^T$ and $\mathcal{N}$ as in Example 5. We obtain a model $\mathcal{A}$ of $\overline{\mathcal{R}} \cup \mathsf{SuH}_\mathsf{s}^T \cup \mathcal{N} \cup \{\neg\varphi_{CR}\}$ with Mace4. The domain is $\mathcal{A}_\mathsf{s} = \{0, 1, 2\}$ and symbols are interpreted by: $\mathsf{a}^\mathcal{A} = 0$, $\mathsf{b}^\mathcal{A} = \mathsf{c}^\mathcal{A} = 1$, $\mathsf{d}^\mathcal{A} = 2$, $x \to^\mathcal{A} y \Leftrightarrow x = 1$, and $x(\to^*)^\mathcal{A} y \Leftrightarrow x = y \vee x = 1$. This proves nonconfluence of $\mathcal{R}$. With regard to the refutation witnesses, $\neg\varphi_{CR}$ is

$$(\exists x, y, z : \mathsf{S})(\forall u : \mathsf{S}) \ x \to^* y \wedge x \to^* z \wedge \neg(y \to^* u \wedge z \to^* u) \qquad (41)$$

and its Skolem normal form is

$$(\forall u : \mathsf{S}) \ \mathsf{sk}_x \to^* \mathsf{sk}_y \wedge \mathsf{sk}_x \to^* \mathsf{sk}_z \wedge \neg(\mathsf{sk}_y \to^* u \wedge \mathsf{sk}_z \to^* u) \qquad (42)$$

Mace4 yields $\mathsf{sk}_x^\mathcal{A} = 1$, $\mathsf{sk}_y^\mathcal{A} = 0$ and $\mathsf{sk}_z^\mathcal{A} = 2$; Φ consists of a single sentence; e.g.,

$$(\forall u : \mathsf{S}) \ \mathsf{b} \to^* \mathsf{a} \wedge \mathsf{b} \to^* \mathsf{d} \wedge \neg(\mathsf{a} \to^* u \wedge \mathsf{d} \to^* u) \qquad (43)$$

but also: $(\forall u : \mathsf{S}) \ \mathsf{c} \to^* \mathsf{a} \wedge \mathsf{c} \to^* \mathsf{d} \wedge \neg(\mathsf{a} \to^* u \wedge \mathsf{d} \to^* u)$. Indeed, they represent the two possible cases of nonconfluent behavior in $\mathcal{R}$.

Example 7 (Normalizing TRS). $\mathcal{R}$ is not terminating, but we can prove it *normalizing* (i.e., every term has a normal form) by *disproving* $\overline{\varphi}_{WN}$, for φ_{WN} in Table 1. Therefore, $\overline{\varphi}_{WN}$ is $(\exists x : \mathsf{S})(\forall y : \mathsf{S})(\exists z : \mathsf{S}) \ (\neg(x \to^* y) \vee y \to z)$. We guarantee surjectivity by using $\mathsf{SuH}_\mathsf{s}^T$ in Example 5; we also use $\mathcal{N}$ in Example 5. Mace4 obtains a model $\mathcal{A}$ of $\overline{\mathcal{R}} \cup \mathsf{SuH}_\mathsf{s}^T \cup \mathcal{N} \cup \{\varphi_{WN}\}$ with $\mathcal{A}_\mathsf{s} = \{0, 1, 2\}$, $\mathsf{a}^\mathcal{A} = 0$, $\mathsf{b}^\mathcal{A} = \mathsf{c}^\mathcal{A} = 1$, $\mathsf{d}^\mathcal{A} = 2$, $x \to^\mathcal{A} y \Leftrightarrow x = 1$, and $x(\to^*)^\mathcal{A} y \Leftrightarrow x = y \vee x = 1$.

8   Related Work

In [14, Sect. 6] we already compared our approach to existing techniques and tools for the so-called First-Order Theory of Rewriting [6], which applies to restricted classes of TRSs and formulas. In [16], we show that our semantic approach is practical when applied to arbitrary (Conditional) TRSs.

---

[2] This proves $\mathcal{R}$ *ground* locally confluent, i.e., variables in φ_{WCR} refer to *ground* terms only; since $\mathcal{R}$ is a ground TRS, local confluence and ground local confluence coincide.

McCune's Prover9/Mace4 are popular automated systems for theorem proving in first-order and equational logic. Given a theory $\mathcal{S}$ and a *goal* or *statement* φ, Prover9 tries to prove that $\mathcal{S} \vdash \varphi$ holds. The generator of models Mace4 complements Prover9 as follows: "*If the statement is the denial of some conjecture, any structures found by* Mace4 *are counterexamples to the conjecture*".[3] Accordingly, the user introduces φ in the *goal* section of Mace4, but the system seeks a model of $\mathcal{S} \cup \{\neg\varphi\}$. Indeed, as discussed in Sect. 1, if $\mathcal{A} \models \mathcal{S} \cup \{\neg\varphi\}$ holds, then $\mathcal{S} \vdash \varphi$ does not hold. But, unless φ is an ECBCA, this does not necessarily mean that φ *does not hold* of a program P with $\mathcal{S} = \overline{P}$! Consider the following 'misleading' session with Mace4 that '*disproves*' commutativity of the addition.

Example 8. Consider $\mathcal{R}$ in Example 1. Mace4 obtains a model $\mathcal{A}$ of $\overline{\mathcal{R}} \cup \{\neg(14)\}$ with domain $\mathcal{A} = \{0, 1\}$, and function and predicate symbols as follows: $0^{\mathcal{A}} = 0$, $\mathsf{s}^{\mathcal{A}}(x) = x$, $\mathsf{add}^{\mathcal{A}}(x, y) = \begin{cases} 1 \text{ if } x = 0 \wedge y = 1 \\ 0 \text{ otherwise} \end{cases}$, $\mathsf{mul}^{\mathcal{A}}(x, y) = 0$, and $\to^{\mathcal{A}}$ and $(\to^*)^{\mathcal{A}}$ both interpreted as the equality. Additionally, Mace4 also displays the following: $\mathsf{c1}^{\mathcal{A}} = 0$ and $\mathsf{c2}^{\mathcal{A}} = 1$. These $\mathsf{c1}$ and $\mathsf{c2}$ are *new* Skolem symbols (but *unexpected* for most users!). In practice, Mace4 finds a model for the *Skolem normal form* of $\neg(14)$, which is

$$(\forall z) \, \neg(\mathsf{add}(\mathsf{c1}, \mathsf{c2}) \to^* z \wedge \mathsf{add}(\mathsf{c2}, \mathsf{c1}) \to^* z) \tag{44}$$

Indeed, $\mathcal{A}$ is a model of $\overline{\mathcal{R}} \cup \{(45)\}$. But we *should not* conclude (as suggested by the aforementioned sentences in Mace4 manual) that add is *not* commutative!

The problem in Example 8 is that $h : \mathcal{I}_{\overline{\mathcal{R}}} \to \mathcal{A}$ is *not surjective*. For instance, no ground term $t \in \mathcal{T}_{\Sigma}$ satisfies $t^{\mathcal{A}} = 1$; note that $\mathsf{c1}, \mathsf{c2} \notin \Sigma$. Since proving validity in $\mathcal{I}_{\mathcal{S}}$ is not the main purpose of Mace4, no warning in its documentation prevents the prospective user to give credit to the 'refutation' of (ground) commutativity for the addition computed by Mace4. We believe that our work is helpful to clarify the use of such tools, and even improve it by adding (for instance) sentences reinforcing surjectivity to avoid the problem discussed above. For instance, Mace4 obtains no model of $\overline{\mathcal{R}} \cup \mathsf{SuH}^T \cup \{(45)\}$ with, e.g., $T = \{0, \mathsf{s}(0)\}$.

Proofs by Satisfiability vs. Theorem Proving. In order to further clarify the differences between our approach and the use of first-order theorem proving tools, consider the CTRS $\mathcal{R}$ in [14, Example 1], consisting of the rules

$$\mathsf{b} \to \mathsf{a} \tag{45}$$

$$\mathsf{a} \to \mathsf{b} \Leftarrow \mathsf{c} \to \mathsf{b} \tag{46}$$

Its associated Horn theory $\overline{\mathcal{R}}$ is:

$$(\forall x)\ x \to^* x \tag{47}$$

$$(\forall x, y, z)\ x \to y \wedge y \to^* z \Rightarrow x \to^* z \tag{48}$$

$$b \to a \tag{49}$$

$$c \to^* b \Rightarrow a \to b \tag{50}$$

We consider some simple tests regarding goals $b \to a$ and $a \to b$ and their negations. We tried such four goals with the following theorem provers: Alt-Ergo,[4] Prover9/Mace4, PDL-tableau,[5] and Princess[6] (most of them with a web-interface). Besides attempting a *proof* of each goal with respect to $\overline{\mathcal{R}}$, tools Alt-Ergo, Mace4, and Princess can also generate *models* of the negation of the tested goal when the proof attempt fails. The following table summarizes the results of our test:

#	Goal		Alt-Ergo		Mace4		PDL-tableau		Princess	
	φ	$\mathcal{I}_{\mathcal{R}} \models \varphi$	$\overline{\mathcal{R}} \vdash \varphi$	$\mathcal{A} \models \neg\varphi$	$\overline{\mathcal{R}} \vdash \varphi$	$\mathcal{A} \models \neg\varphi$	$\overline{\mathcal{R}} \vdash \varphi$	$\mathcal{A} \models \neg\varphi$	$\overline{\mathcal{R}} \vdash \varphi$	$\mathcal{A} \models \neg\varphi$
1	$b \to a$	*true*	Y	N	Y	N	Y	–	Y	N
2	$\neg(b \to a)$	*false*	N	Y	N	Y	N	–	N	Y
3	$a \to b$	*false*	N	Y	N	Y	N	–	N	Y
4	$\neg(a \to b)$	*true*	N	Y	N	Y	N	–	N	Y

Goal $\neg(a \to b)$ in row 4 is *not* directly proved by any tool. Indeed, since $\neg(a \to b)$ is *not* a logical consequence of $\overline{\mathcal{R}}$ (see [14, Example 2]), $\overline{\mathcal{R}} \vdash \neg(a \to b)$ does *not* hold. Our *satisfiability* approach can be used to formally prove that $\mathcal{R}$ cannot reduce a into b, i.e., that $\mathcal{I}_{\mathcal{R}} \models \neg(a \to b)$ (or $a \not\to_{\mathcal{R}} b$) holds: from row 3 we see that $\mathcal{A} \models \neg(a \to b)$ holds for the models $\mathcal{A}$ of $\overline{\mathcal{R}}$ computed by some of the tools. By Corollary 1, the desired conclusion $a \not\to_{\mathcal{R}} b$ follows. Note also that row 4 reports on the ability of some tools to obtain models of $a \to b$. However, Corollary 1 cannot be used to conclude that $a \to_{\mathcal{R}} b$ holds (which is obviously wrong): since φ in row 4 is a *negative* literal, condition (b) in Corollary 1 must be fulfilled before being able to conclude $\mathcal{I}_{\mathcal{R}} \models a \to b$ from $\mathcal{A} \models a \to b$ for some model $\mathcal{A}$ of $\overline{\mathcal{R}}$. But this is *not* the case in our test set.

Although Remark 1 explains how an arbitrary program property φ can be proved by using Corollary 1 (see also Sect. 7), from a practical point of view we better think of our approach as *complementary* to the use of first-order proof techniques and tools. Provability of φ (i.e., $\mathcal{S} \vdash \varphi$) implies that $\mathcal{I}_{\mathcal{S}} \models \varphi$ holds. Thus, as usual, a proof of φ with respect to $\mathcal{S}$ implies that a program P with $\mathcal{S} = \overline{P}$ has property φ. However, as discussed above, showing that $\mathcal{S} \vdash \varphi$ or $\mathcal{S} \vdash \neg\varphi$ holds is often impossible. We can try to prove $\mathcal{I}_{\mathcal{S}} \models \neg\varphi$ by using Corollary 1, though. For positive sentences φ, this is often affordable.

---

9 Conclusions and Future Work

We have shown how to prove properties φ of computational systems whose semantics can be given as a first-order theory $\mathcal{S}$. Our *proofs by satisfiability* proceed (see Remark 1) by just finding a model $\mathcal{A}$ of $\mathcal{S} \cup \mathcal{Z} \cup \{\varphi\}$ where $\mathcal{Z}$ is an *auxiliary* theory representing the requirements (a) and (b) in Corollary 1 (referred to $\neg\varphi$), so that $\mathcal{A} \models \mathcal{S} \cup \mathcal{Z} \cup \{\varphi\}$ implies $\mathcal{I}_\mathcal{S} \models \varphi$. Surjectivity of the interpretation homomorphisms (requirement (a) in Corollary 1) is ensured if $\mathcal{Z}$ includes the appropriate theory SuH (see Sect. 4); and requirement (b), for dealing with negative literals, is fulfilled if $\mathcal{Z}$ includes $\mathcal{N}$ in Proposition 1. Our results properly subsume the ones in [14], which concern *existentially closed boolean combinations of atoms* only. We have also introduced the notion of *refutation witness* which is useful to obtain counterexamples by using the symbols in the first-order language rather than values of the computed model.

From a theoretical point of view, the idea of *proving program properties as satisfiability* (see Remark 1) is appealing as it emphasizes the role of *abstraction* (introduced by semantic structures) in theorem proving and logic-based program analysis. However, the requirement of surjectivity of the interpretation homomorphisms and the use of theories $\mathcal{N}$ with *negative* information about some of the predicates introduce additional difficulties in the model generation process. Investigating methods for the practical implementation of our techniques, and also finding specific areas of application where our approach can be useful (as done in [16], for instance) is an interesting subject for future work.

Also, our research suggests that further investigation on the generation of models for many-sorted theories that combines the use of finite and infinite domains is necessary. For instance, [15] explains how to generate such models by interpreting the sort, function, and predicate symbols by using linear algebra techniques. This is implemented in AGES. Domains are defined as the solutions of matrix inequalities, possibly restricted to an underlying set of values (e.g., $\mathbb{Z}$); thus, finite and infinite domains can be obtained as particular cases of the same technique. Since piecewise definitions are allowed, we could eventually provide fully detailed descriptions of functions and predicates by just adding more pieces to the interpretations. However, such a flexibility is expensive. In contrast, Mace4 is based on a different principle (similar to [12]) and it is really fast, but only finite domains can be generated. This is a problem, for instance, when using Proposition 3 to guarantee surjectivity of homomorphisms $h_s : \mathcal{T}_{\Sigma s} \to \mathcal{A}_s$. Even though $\mathcal{A}_s$ is finite, we still need to be able to interpret *Nat* as $\mathbb{N}$, which is not possible with Mace4. For this reason, the examples in Sect. 5 (where the computed structures $\mathcal{A}$ have finite domains for the 'proper' sorts N, LN, and S, and only *Nat* is interpreted as an infinite set) could not be handled with Mace4, or with similar tools that are able to deal with sorts (e.g., SEM [24] or the work in [21]) but which generate finite domains only.

References

1. Baader, F., Nipkow, T.: Term Rewriting and All That. Cambridge University Press, New York (1998)
2. Boolos, G.S., Burgess, J.P., Jeffrey, R.C.: Computability and Logic, 4th edn. Cambridge University Press, Cambridge (2002)
3. Bruni, R., Meseguer, J.: Semantic foundations for generalized rewrite theories. Theoret. Comput. Sci. **351**(1), 386–414 (2006)
4. Clark, K.L.: Predicate logic as a computational formalism. Ph.D. thesis, Research Monograph 79/59 TOC, Department of Computing, Imperial College of Science, and Technology, University of London, December 1979
5. Clavel, M., et al.: All About Maude - A High-Performance Logical Framework. LNCS, vol. 4350. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71999-1
6. Dauchet, M., Tison, S.: The theory of ground rewrite systems is decidable. In: Proceedings of LICS 1990, pp. 242–248. IEEE Press (1990)
7. Dershowitz, N., Kaplan, S., Plaisted, D.: Rewrite, rewrite, rewrite, rewrite, rewrite, . . . Theoret. Comput. Sci. **83**, 71–96 (1991)
8. Endrullis, J., Hendriks, D.: Lazy productivity via termination. Theoret. Comput. Sci. **412**, 3203–3225 (2011)
9. Goguen, J.A., Meseguer, J.: Models and equality for logical programming. In: Ehrig, H., Kowalski, R., Levi, G., Montanari, U. (eds.) TAPSOFT 1987. LNCS, vol. 250, pp. 1–22. Springer, Heidelberg (1987). https://doi.org/10.1007/BFb0014969
10. Gutiérrez, R., Lucas, S., Reinoso, P.: A tool for the automatic generation of logical models of order-sorted first-order theories. In: Proceedings of PROLE 2016, pp. 215–230 (2016). Tool available at http://zenon.dsic.upv.es/ages/
11. Hodges, W.: Model Theory. Cambridge University Press, Cambridge (1993)
12. Kim, S., Zhang, H.: ModGen: theorem proving by model generation. In: Proceedings of AAAI 1994, pp. 162–167. AAAI Press/MIT Press (1994)
13. Kleene, S.C.: Mathematical Logic. Wiley, Hoboken (1967). (Dover 2002)
14. Lucas, S.: Analysis of rewriting-based systems as first-order theories. In: Fioravanti, F., Gallagher, J.P. (eds.) LOPSTR 2017. LNCS, vol. 10855, pp. 180–197. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94460-9_11
15. Lucas, S., Gutiérrez, R.: Automatic synthesis of logical models for order-sorted first-order theories. J. Autom. Reason. **60**(4), 465–501 (2018)
16. Lucas, S., Gutiérrez, R.: Use of logical models for proving infeasibility in term rewriting. Inf. Process. Lett. **136**, 90–95 (2018)
17. McCune, W.: Prover9 and Mace4 (2005–2010). http://www.cs.unm.edu/~mccune/prover9/
18. Meseguer, J.: Twenty years of rewriting logic. J. Logic Algebraic Program. **81**, 721–781 (2012)
19. Ohlebusch, E.: Advanced Topics in Term Rewriting. Springer, New York (2002). https://doi.org/10.1007/978-1-4757-3661-8
20. Reiter, R.: On closed world data bases. In: Logic and Data Bases, pp. 119–140. Plenum Press, New York (1978)
21. Reger, G., Suda, M., Voronkov, A.: Finding finite models in multi-sorted first-order logic. In: Creignou, N., Le Berre, D. (eds.) SAT 2016. LNCS, vol. 9710, pp. 323–341. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40970-2_20

22. Shapiro, S.: Foundations Without Foundationalism: A Case for Second-Order Logic. Clarendon Press, Oxford (1991)
23. Wang, H.: Logic of many-sorted theories. J. Symbolic Logic **17**(2), 105–116 (1952)
24. Zhang, J., Zhang, H.: Generating models by SEM. In: McRobbie, M.A., Slaney, J.K. (eds.) CADE 1996. LNCS, vol. 1104, pp. 308–312. Springer, New York (1996)