Models for Logics and Conditional Constraints in Automated Proofs of Termination

Salvador Lucas[1,2] and José Meseguer[2]

[1] DSIC, Universitat Politècnica de València, Spain
[2] CS Dept. University of Illinois at Urbana-Champaign, IL, USA

Abstract. Reasoning about termination of declarative programs, which are described by means of a *computational logic*, requires the definition of appropriate abstractions as *semantic models* of the logic, and also handling the *conditional constraints* which are often obtained. The formal treatment of such constraints in automated proofs, often using *numeric* interpretations and (arithmetic) constraint solving, can greatly benefit from appropriate techniques to deal with the conditional (in)equations at stake. Existing results from linear algebra or real algebraic geometry are useful to deal with them but have received only scant attention to date. We investigate the definition and use of numeric models for logics and the resolution of linear and algebraic conditional constraints as unifying techniques for proving termination of declarative programs.

Keywords: Conditional constraints, program analysis, termination.

1 Introduction

The operational semantics of sophisticated rule-based programming languages such as CafeOBJ [7], Maude [3], or Haskell [8] is often formalized in a proof-theoretic style by means of a *computational logic*, and the corresponding language interpreters better understood as *inference* machines [15]. The notion of *operational termination* [11] was introduced to give an account of the termination behavior of programs of such languages [12]. An *interpreter* for a logic $\mathcal{L}$ (for instance the logic for *Conditional Term Rewriting Systems* (CTRSs) with inference system in Figure 1) is an *inference machine* that, given a theory $\mathcal{S}$ (e.g., the CTRS $\mathcal{R}$ in Example 1) and a goal formula φ (e.g., a one-step rewriting $s \to t$ for terms s and t) tries to incrementally build a proof tree for φ by using (instances of) the inference rules $\frac{B_1,\ldots,B_n}{A} \in \mathcal{I}(\mathcal{L})$ of the inference system $\mathcal{I}(\mathcal{L})$ of $\mathcal{L}$. Then, $\mathcal{S}$ is *operationally terminating* if for any φ the interpreter either finds a proof, or fails in all possible attempts (always in finite time). In this setting, practical methods for proving operational termination involve two main issues (see [13] and also [17] for CTRSs): (1) the *simulation* of the (one-step) rewrite relations $\to$ and $\to^*$ associated to a CTRS $\mathcal{R}$ and defined by means of the inference system in Figure 1; and (2) the use of (automatically generated) *well-founded relations* $\sqsupset$ to *abstract* rewrite computations and guarantee the absence

$$
\begin{array}{llll}
\text{(Refl)} & \dfrac{}{t \to^* t} & \text{(Cong)} & \dfrac{s_i \to t_i}{f(s_1,\ldots,s_i,\ldots,s_k) \to f(s_1,\ldots,t_i,\ldots,s_k)} \\
 & & & \text{for all } k\text{-ary symbols } f \text{ and } 1 \leq i \leq k \\
\text{(Tran)} & \dfrac{s \to u \qquad u \to^* t}{s \to^* t} & \text{(Repl)} & \dfrac{s_1 \to^* t_1 \;\ldots\; s_n \to^* t_n}{\ell \to r} \\
 & & & \text{for each rule } \ell \to r \Leftarrow s_1 \to t_1 \cdots s_n \to t_n
\end{array}
$$

Fig. 1. Inference rules for the CTRS logic (all variables are universally quantified)

of infinite ones. Here, (1) amounts at dealing with abstractions for sentences like $\forall \boldsymbol{x}(B_1 \wedge \cdots \wedge B_n \Rightarrow A)$ which *simulate* the use of the aforementioned inference rules; and (2) often involves the comparison of expressions s and t using $\sqsupset$, provided that a number of *semantic* conditions (e.g., rewriting steps $s_i \to^*_{\mathcal{R}} t_i$ for some terms s_i and t_i) hold. Abstractions can be formalized as *semantic models* $\mathcal{M} = (\mathsf{D}, \mathcal{F}_\mathsf{D}, \Pi_\mathsf{D})$ of $\mathcal{L}$ (see Section 2), where D is a *domain* and $\mathcal{F}_\mathsf{D}$ and Π_D are interpretations of the function symbols $\mathcal{F}$ and predicates Π of $\mathcal{L}$, respectively. For instance, relations $\to$ and $\to^*$ (which are predicates in the corresponding logic) are typically interpreted as *orderings* on D, often a *numeric* domain like $\mathbb{N}$ or $[0, +\infty)$. In this paper we introduce the idea of using *conditional expressions* to *restrict* such domains in logical models (Section 3). This is often useful.

Example 1. Consider the following CTRS $\mathcal{R}$:

$$
\begin{array}{rlrlrl}
\mathsf{or}(\mathsf{0},x) \to x & (1) & \mathsf{or}(\mathsf{not}(x),x) \to \mathsf{1} & (6) & \mathsf{and}(x,\mathsf{not}(x)) \to \mathsf{0} & (11) \\
\mathsf{or}(x,\mathsf{0}) \to x & (2) & \mathsf{and}(\mathsf{0},x) \to \mathsf{0} & (7) & \mathsf{and}(\mathsf{not}(x),x) \to \mathsf{0} & (12) \\
\mathsf{or}(\mathsf{1},x) \to \mathsf{1} & (3) & \mathsf{and}(x,\mathsf{0}) \to \mathsf{0} & (8) & \mathsf{not}(\mathsf{1}) \to \mathsf{0} & (13) \\
\mathsf{or}(x,\mathsf{1}) \to \mathsf{1} & (4) & \mathsf{and}(\mathsf{1},x) \to x & (9) & \mathsf{not}(\mathsf{0}) \to \mathsf{1} & (14) \\
\mathsf{or}(x,\mathsf{not}(x)) \to \mathsf{1} & (5) & \mathsf{and}(x,\mathsf{1}) \to x & (10) & &
\end{array}
$$

$$\mathsf{implies}(x,y) \to \mathsf{1} \Leftarrow \mathsf{not}(x) \to \mathsf{1} \tag{15}$$

$$\mathsf{implies}(x,y) \to \mathsf{1} \Leftarrow y \to \mathsf{1} \tag{16}$$

$$\mathsf{implies}(x,y) \to \mathsf{0} \Leftarrow x \to \mathsf{1}, y \to \mathsf{0} \tag{17}$$

$$f(x) \to f(\mathsf{0}) \Leftarrow \mathsf{implies}(\mathsf{implies}(x,\mathsf{implies}(x,\mathsf{0})),\mathsf{0}) \to \mathsf{1} \tag{18}$$

We *failed* to prove operational termination of $\mathcal{R}$ by using the ordering-based techniques introduced in [13] and also with the more advanced techniques in [14]. However, below we provide a very simple proof of operational termination based on the use (within [13]!) of a *bounded* domain $[0, 1]$ which can be easily implemented by using conditional constraints.

As an interesting specialization of this general idea, Section 4 introduces *convex matrix interpretations* as a new, twofold extension of the framework introduced by Endrullis et al. [5] for TRSs, where rather than using vectors $\boldsymbol{x}$ of natural numbers (or non-negative numbers, as in [2]), we use *convex sets* satisfying a matrix inequality $A\boldsymbol{x} \geq \boldsymbol{b}$. Section 5 discusses existing approaches to deal with the obtained numeric conditional constraints: *Farkas' Lemma* and results from *Algebraic Geometry*. Section 6 compares with related work. Section 7 concludes.

2 Models of Logics for Proofs of Termination

In this paper, $\mathcal{X}$ denotes a set of variables and $\mathcal{F}$ denotes a *signature*: a set of function symbols $\{f, g, \ldots\}$, each with a fixed *arity* given by a mapping $ar : \mathcal{F} \to \mathbb{N}$. The set of *terms* built from $\mathcal{F}$ and $\mathcal{X}$ is $\mathcal{T}(\mathcal{F}, \mathcal{X})$. A CTRS $\mathcal{R} = (\mathcal{F}, R)$ consist of a signature $\mathcal{F}$ and a set R of rules $\ell \to r \Leftarrow s_1 \to t_1, \cdots, s_n \to t_n$, where $l, r, s_1, t_1, \cdots, s_n, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. For $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, we write $s \to_{\mathcal{R}} t$ ($s \to^*_{\mathcal{R}} t$) if there is a proof for $s \to t$ ($s \to^* t$) with the inference system in Figure 1.

Given a (first order) *logic signature* $\Sigma = (\mathcal{F}, \Pi)$ where $\mathcal{F}$ is a signature of function symbols and Π is a signature of *predicate symbols*, the formulas φ of a (first order) logic $\mathcal{L}$ over Σ are built up from atoms $P(t_1, \ldots, t_k)$ with $P \in \Pi$ and $t_1, \ldots, t_k \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, logic connectives (e.g., $\wedge$, $\neg$, $\Rightarrow$) and quantifiers ($\forall, \exists$) in the usual way; $Form_\Sigma$ is the set of such formulas. A *theory* $\mathcal{S}$ of $\mathcal{L}$ is a set of formulas, $\mathcal{S} \subseteq Form_\Sigma$, and its *theorems* are the formulas $\varphi \in Form_\Sigma$ for which we can derive a proof using the inference system $\mathcal{I}(\mathcal{L})$ of $\mathcal{L}$ in the usual way (written $\mathcal{S} \vdash \varphi$). Given a logic $\mathcal{L}$ describing computations in a (declarative) programming language, programs are viewed as *theories* $\mathcal{S}$ of $\mathcal{L}$.

Example 2. In the logic of CTRSs, with binary *predicates* $\to$ and $\to^*$, the theory for a CTRS $\mathcal{R} = (\mathcal{F}, R)$ is obtained from the inference rules in Figure 1 after *specializing* them as $(Cong)_{f,i}$ for each $f \in \mathcal{F}$ and i, $1 \leq i \leq ar(f)$ and $(Repl)_\rho$ for all $\rho : \ell \to r \Leftarrow c \in R$. Then, inference rules $\frac{B_1, \ldots, B_n}{A}$ become *implications* $B_1 \wedge \cdots \wedge B_n \Rightarrow A$. For instance, for $(Tran)$, $(Cong)_{\mathsf{not}}$, $(Repl)_{(1)}$, and $(Repl)_{(15)}$:

$$\forall s, t, u\, (s \to u \wedge u \to^* t \Rightarrow s \to^* t) \tag{19}$$

$$\forall s, t\, (s \to t \Rightarrow \mathsf{not}(s) \ \to\ \mathsf{not}(t)) \tag{20}$$

$$\forall x\, (\mathsf{or}(0, x) \ \to\ x) \tag{21}$$

$$\forall x, y\, (\mathsf{not}(x) \to^* 1 \Rightarrow \mathsf{implies}(x, y) \ \to\ 1) \tag{22}$$

For analysis and verification purposes we often need to *abstract* $\mathcal{L}$ into a *numeric* setting (e.g., arithmetics, linear algebra, or algebraic geometry) where appropriate techniques are available to *prove* properties of interest. This amounts at giving a (numeric) *model* of $\mathcal{L}$ that *satisfies* $\mathcal{S}$.

An $\mathcal{F}$-algebra is a pair $\mathcal{A} = (\mathsf{D}, \mathcal{F}_\mathsf{D})$, where D is a set and $\mathcal{F}_\mathsf{D}$ is a set of mappings $f_\mathcal{A} : \mathsf{D}^k \to \mathsf{D}$ for each $f \in \mathcal{F}$ where $k = ar(f)$. A Σ*-model* is a triple $\mathcal{M} = (\mathsf{D}, \mathcal{F}_\mathsf{D}, \Pi_\mathsf{D})$ where $(\mathsf{D}, \mathcal{F}_\mathsf{D})$ is an $\mathcal{F}$-algebra, and for each k-ary $P \in \Pi$, $P_\mathcal{M} \in \Pi_\mathsf{D}$ is a k-ary relation $P_\mathcal{M} \subseteq \mathsf{D}^k$. Given a *valuation mapping* $\alpha : \mathcal{X} \to \mathsf{D}$, the evaluation mapping $[\_]^\mathcal{A}_\alpha : \mathcal{T}(\mathcal{F}, \mathcal{X}) \to \mathsf{D}$ (also $[\_]^\mathcal{M}_\alpha$ if $\mathcal{A}$ is part of $\mathcal{M}$) is the unique homomorphism extending α. Finally, $[\_]^\mathcal{M}_\alpha : Form_\Sigma \to Bool$ is given by:

1. $[P(t_1, \ldots, t_k)]^\mathcal{M}_\alpha = \mathsf{true}$ if and only if $([t_1]^\mathcal{M}_\alpha, \ldots, [t_k]^\mathcal{M}_\alpha) \in P_\mathcal{A}$;
2. $[\varphi \wedge \psi]^\mathcal{M}_\alpha = \mathsf{true}$ if and only if $[\varphi]^\mathcal{M}_\alpha = \mathsf{true}$ and $[\psi]^\mathcal{M}_\alpha = \mathsf{true}$;
3. $[\varphi \Rightarrow \psi]^\mathcal{M}_\alpha = \mathsf{true}$ if and only if $[\varphi]^\mathcal{M}_\alpha = \mathsf{false}$ or $[\psi]^\mathcal{M}_\alpha = \mathsf{true}$;
4. $[\neg\varphi]^\mathcal{M}_\alpha = \mathsf{true}$ if and only if $[\varphi]^\mathcal{M}_\alpha = \mathsf{false}$;
5. $[\exists x\, \varphi]^\mathcal{M}_\alpha = \mathsf{true}$ if and only if there is $a \in \mathsf{D}$ such that $[\varphi]^\mathcal{M}_{\alpha[x \mapsto a]} = \mathsf{true}$;
6. $[\forall x\, \varphi]^\mathcal{M}_\alpha = \mathsf{true}$ if and only if for all $a \in \mathsf{D}$, $[\varphi]^\mathcal{M}_{\alpha[x \mapsto a]} = \mathsf{true}$;

We say that $\mathcal{M}$ *satisfies* $\varphi \in Form_\Sigma$ if there is $\alpha \in \mathcal{X} \to \mathsf{D}$ such that $[\varphi]^{\mathcal{M}}_\alpha = \mathsf{true}$. If $[\varphi]^{\mathcal{M}}_\alpha = \mathsf{true}$ for *all* valuations α, we write $\mathcal{M} \models \varphi$. A closed formula, i.e., a formula whose variables are all universally or existentially quantified, is called a *sentence*. We say that $\mathcal{M}$ is *a model of a set of sentences* $\mathcal{S} \subseteq Form_\Sigma$ (written $\mathcal{M} \models \mathcal{S}$) if for all $\varphi \in \mathcal{S}$, $\mathcal{M} \models \varphi$. And, given a sentence φ, we write $\mathcal{S} \models \varphi$ if and only if for *all models* $\mathcal{M}$ of $\mathcal{S}$, $\mathcal{M} \models \varphi$. *Sound* logics guarantee that every provable sentence φ is true in *every model* of $\mathcal{S}$, i.e., $\mathcal{S} \vdash \varphi$ implies $\mathcal{S} \models \varphi$.

In practice, $\mathcal{F}$-algebras $\mathcal{A}$ can be obtained if we first consider a *new* set of terms $\mathcal{T}(\mathcal{G}, \mathcal{X})$ where the new symbols $g \in \mathcal{G}$ have 'intended' (often arithmetic) interpretations over an (arithmetic) domain D as mappings g from D into D. The use of the *same name* for the syntactic and semantic objects stresses that they have an *intended* meaning. We associate an expression $e_f \in \mathcal{T}(\mathcal{G}, \{x_1, \ldots, x_k\})$ to each k-ary symbol $f \in \mathcal{F}$, where $x_1, \ldots, x_k \in \mathcal{X}$ are different variables: we write $[f](x_1, \ldots, x_k) = e_f$; and homomorphically extend it to $[\_] : \mathcal{T}(\mathcal{F}, \mathcal{X}) \to \mathcal{T}(\mathcal{G}, \mathcal{X})$. Then, for all $a_1, \ldots, a_k \in \mathsf{D}$, we let $f_{\mathcal{A}}(a_1, \ldots, a_k) = [e_f]_{\alpha_{\boldsymbol{a}}}$, for $\alpha_{\boldsymbol{a}}$ given by $\alpha_{\boldsymbol{a}}(x_i) = a_i$ for all $1 \le i \le k$.

Example 3. For $\mathcal{R}$ in Example 1, $\mathcal{F} = \{\mathsf{0}, \mathsf{1}, \mathsf{or}, \mathsf{and}, \mathsf{not}, \mathsf{implies}, \mathsf{f}\}$, where $ar(\mathsf{0}) = ar(\mathsf{1}) = 0$, $ar(\mathsf{f}) = 1$, and $ar(\mathsf{or}) = ar(\mathsf{and}) = ar(\mathsf{implies}) = 2$. Let $\mathcal{G} = \{0, 1, max, min, \_ - \_\}$ with $ar(0) = ar(1) = 0$ and $ar(max) = ar(min) = ar(\_ - \_) = 2$. We define an $\mathcal{F}$-algebra over the reals $\mathbb{R}$ as follows:

$$
\begin{array}{llll}
[\mathsf{0}] = 0 & [\mathsf{and}](x,y) = min(x,y) & [\mathsf{or}](x,y) = max(x,y) & [\mathsf{f}](x) = 0 \\
[\mathsf{1}] = 1 & [\mathsf{not}](x) = 1 - x & [\mathsf{implies}](x,y) = max(1-x,y) &
\end{array}
$$

We define a model $\mathcal{M} = (\mathsf{D}, \mathcal{F}_{\mathsf{D}}, \Pi_{\mathsf{D}})$ if each $P \in \Pi$ is interpreted as a predicate $P_{\mathcal{M}} \in \Pi_{\mathsf{D}}$, and each $\varphi \in Form_\Sigma$ as a formula $\varphi_{\mathcal{M}}$, where $\varphi_{\mathcal{M}} = P_{\mathcal{M}}([t_1], \ldots, [t_k])$ if $\varphi = P(t_1, \ldots, t_k)$; $\varphi_{\mathcal{M}} = \varphi_{\mathcal{M}} \oplus \psi_{\mathcal{A}}$ if $\varphi = \chi \oplus \psi$ for $\oplus \in \{\wedge, \Rightarrow\}$ and $\varphi_{\mathcal{M}} = \Box\chi_{\mathcal{M}}$ if $\varphi = \Box\chi$ for $\Box \in \{\neg, \forall, \exists\}$. The goal is *proving* that $\mathcal{M} \models \mathcal{S}$ holds.

Example 4. We can interpret both $\to$ and $\to^*$ as '$=$' (intended to be the equality among real numbers). Then, the sentences in Example 2 become

$$\forall s, t, u \in \mathbb{R}\, (s = u \wedge u = t \Rightarrow s = t) \tag{23}$$

$$\forall s, t \in \mathbb{R}\, (s = t \Rightarrow 1 - s = 1 - t) \tag{24}$$

$$\forall x \in \mathbb{R}\, (max(0, x) = x) \tag{25}$$

$$\forall x, y \in \mathbb{R}\, (1 - x = 1 \Rightarrow max(1 - x, y) = 1) \tag{26}$$

Unfortunately, (25) and (26) do *not* hold in the intended model due to the (big) algebraic domain $\mathbb{R}$. For instance, $max(0, -1) = 0 \neq -1$, i.e., (25) is *not* true.

Example 4 shows that the appropriate definition of the *domain* of a model is crucial to satisfy a set of formulas. The next section investigates this problem.

3 Domains for algebras and models revisited

In proofs of termination, domains D for numeric $\mathcal{F}$-algebras $\mathcal{A}$ usually are infinite (subsets of) n-dimensional open intervals which are bounded from below: $\mathbb{N}^n$ or

$[0,+\infty)^n$ for some $n \geq 1$. Furthermore, considered orderings often make the corresponding ordered sets *total* (like $[0,+\infty)$ ordered by $\geq_{\mathbb{R}}$), or nontotal but with subsets $B \subseteq \mathsf{D}$ bounded by some value $x_B \in \mathsf{D}$ (like $[0,+\infty)^n$ ordered by the pointwise extension of the usual ordering $\geq_{\mathbb{R}}$ over the reals, which is a complete lattice). More general domains can be often useful, though.

Example 5. (Continuing Example 4) Although (23) and (24) always hold (under the *intended* interpretation of '=' as the equality), *satisfiability* of other sentences may depend on the considered *domain* of values: if $\mathsf{D} = [0,1]$, then (25) and (26) hold; if $\mathsf{D} = \mathbb{N}$, then only (25) holds. The use of $\mathsf{D} = [0,1]$ can be made *explicit* in (25) and (26) by adding further constraints:

$$\forall x \in \mathbb{R}\ (\ x \geq 0 \wedge 1 \geq x \Rightarrow max(0,x) = x) \quad (27)$$

$$\forall x,y \in \mathbb{R}\ (\ x \geq 0 \wedge 1 \geq x \wedge y \geq 0 \wedge 1 \geq y \wedge 1-x = 1 \Rightarrow max(1-x,y) = 1) \quad (28)$$

Thus, we need to deal with *conditional constraints* for using such more general domains. Also to handle *max* expressions [6, 16].

Example 6. We can *expand* the definition of *max* in (27) and (28) into:

$$\forall x \in \mathbb{R}\ (\ x \geq 0 \wedge 1 \geq x \wedge 0 \geq x \Rightarrow 0 = x) \quad (29)$$

$$\forall x \in \mathbb{R}\ (\ x \geq 0 \wedge 1 \geq x \wedge x \geq 0 \Rightarrow x = x) \quad (30)$$

$$\forall x,y \in \mathbb{R}\ (\ x \geq 0 \wedge 1 \geq x \wedge y \geq 0 \wedge 1 \geq y \wedge 1-x = 1 \wedge 1-x \geq y \Rightarrow 1-x = 1) \quad (31)$$

$$\forall x,y \in \mathbb{R}\ (\ x \geq 0 \wedge 1 \geq x \wedge y \geq 0 \wedge 1 \geq y \wedge 1-x = 1 \wedge y > 1-x \Rightarrow y = 1) \quad (32)$$

where (30) clearly holds true and we do not longer care about it.

3.1 Conditional domains for term algebras and models

Given a set D and a predicate χ over D, we let $\mathsf{D}_\chi = \{x \in \mathsf{D} \mid \chi(x)\}$ be the *restriction* of D by χ. An $\mathcal{F}$-algebra $\mathcal{A} = (\mathsf{D}, \mathcal{F}_{\mathsf{D}})$ yields a *restricted* $\mathcal{F}$-algebra $\mathcal{A}_\chi = (\mathsf{D}_\chi, \mathcal{F}_{\mathsf{D}_\chi})$, where for each $f \in \mathcal{F}$, $f_{\mathcal{A}_\chi}$ is the restriction of $f_{\mathcal{A}}$ to D_χ^k, if for all k-ary symbols $f \in \mathcal{F}$, this *algebraicity* or *closedness* condition holds:

$$\forall x_1,\ldots,x_k \left(\left(\bigwedge_{i \leq i \leq k} \chi(x_i) \right) \Rightarrow \chi(f_{\mathcal{A}}(x_1,\ldots,x_k)) \right) \quad (33)$$

guaranteeing that if $f_{\mathcal{A}}$ is given inputs in D_χ, the outcome belongs to D_χ as well.

Remark 1. Algebraicity is a standard requirement for algebraic interpretations. Most times, however, the imposition of simple requirements on the shape of the numeric expressions e_f used to define $f_{\mathcal{A}}$ (see Section 2) makes this task easy and often avoids any checking. A well-known example is taking $\mathcal{D} = [0,+\infty)$ and requiring e_f to be a *polynomial* whose coefficients are all *non-negative*.

The relations $P_{\mathcal{M}} \subseteq \mathsf{D}^k$ interpreting k-ary predicates $P \in \Pi$ can be restricted to $P_{\mathcal{M}_\chi} = P_{\mathcal{M}} \cap \mathsf{D}_\chi^k$ to yield a new interpretation of P in $\mathcal{M}_\chi = (\mathsf{D}_\chi, \mathcal{F}_{\mathsf{D}_\chi}, \Pi_{\mathsf{D}_\chi})$. For practical purposes, in this paper we only consider simple restrictions of $\mathcal{F}$-algebras and models, where D is obtained as the solution of *linear* constraints.

Definition 1 (Convex polytopic domain). *Given a matrix* $A \in \mathbb{R}^{m\times n}$*, and* $\boldsymbol{b} \in \mathbb{R}^m$*, the set of solutions of the inequality* $Ax \geq \boldsymbol{b}$ *is a* convex polytope $D(A,\boldsymbol{b}) = \{\boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} \geq \boldsymbol{b}\}$*. We call* $D(A,\boldsymbol{b})$ *a* convex polytopic domain.

Example 7. For $A = (-1,1)^T$ and $\boldsymbol{b} = (-1,0)$, we have $D(A,\boldsymbol{b}) = [0,1]$. If $A = (1)$ and $\boldsymbol{b} = (0)$, then $D(A,\boldsymbol{b}) = [0,+\infty)$.

Example 8. Continuing Example 3, we obtain an $\mathcal{F}$-algebra $\mathcal{A}_{[0,1]} = ([0,1],\mathcal{F}_{[0,1]})$ as the restriction to $[0,1]$ of the $\mathcal{F}$-algebra over $\mathbb{R}$ defined there. The constraints (25) and (26) are written in the restricted model as follows

$$\forall x \in [0,1]\,(max(0,x) = x) \tag{34}$$

$$\forall x,y \in [0,1]\,(1-x = 1 \Rightarrow max(1-x,y) = 1) \tag{35}$$

After encoding memberships like $x \in [0,1]$ as inequalities $x \geq 0 \wedge 1 \geq x$ and expanding the definition of max, we obtain (29) – (32).

In sharp contrast with Example 4, restricting the model at hand to $[0,1]$ leads to a model for $\mathcal{R}$ in Example 1 which is useful to prove its operational termination.

Example 9. According to [13], $\mathcal{R}$ in Example 1 is operationally terminating if there is a relation $\gtrsim$ on terms such that $\to^* \subseteq \gtrsim$, and a well-founded ordering $\sqsupset$ satisfying $\gtrsim \circ \sqsupset \subseteq \sqsupset$ such that, for all substitutions σ, if $\sigma(\mathsf{implies}(\mathsf{implies}(x,\mathsf{implies}(x,\mathsf{0})),\mathsf{0})) \to^*_{\mathcal{R}} \sigma(1)$ holds, then $\sigma(\mathsf{F}(x)) \sqsupset \sigma(\mathsf{F}(\mathsf{0}))$ for the rule (*dependency pair*[3]) $\mathsf{F}(x) \to \mathsf{F}(\mathsf{0}) \Leftarrow \mathsf{implies}(\mathsf{implies}(x,\mathsf{implies}(x,\mathsf{0})),\mathsf{0}) \to 1$ (where F is a fresh symbol). Let $\mathcal{M} = ([0,1],\mathcal{F}'_{[0,1]},\Pi_{[0,1]})$ where $\mathcal{F}' = \mathcal{F} \cup \{F\}$, $\mathcal{F}'_{[0,1]}$ is $\mathcal{F}_{[0,1]}$ as in Example 8 extended with $[\mathsf{F}](x) = x$, and $\Pi_{[0,1]}$ given by $\to_{[0,1]} = \to^*_{[0,1]} = (=_{[0,1]})$ (i.e., the equality on $[0,1]$). $\mathcal{M}$ is a model of $\mathcal{R}$; by soundness, if $s \to^* t$ holds for $s,t \in \mathcal{T}(\mathcal{F},\mathcal{X})$, we have $[s] =_{[0,1]} [t]$. Let $\gtrsim$ be as follows: for all $s,t \in \mathcal{T}(\mathcal{F},\mathcal{X})$, $s \gtrsim t$ holds if and only if $[s] =_{[0,1]} [t]$. Then, $\to^* \subseteq \gtrsim$, as desired.

Now, consider the ordering $>_1$ over $\mathbb{R}$ given by $x >_1 y$ if and only if $x - y \geq 1$; it is a well-founded relation on $[0,1]$ (see [10]). We let $\sqsupset$ be the (well-founded) relation on $\mathcal{T}(\mathcal{F},\mathcal{X})$ induced by $>_1$ as before. Again, for all substitutions σ, if $\sigma(\mathsf{implies}(\mathsf{implies}(x,\mathsf{implies}(x,\mathsf{0})),\mathsf{0})) \to^*_{\mathcal{R}} \sigma(1)$ holds, then, by soundness,

$$[\sigma(\mathsf{implies}(\mathsf{implies}(x,\mathsf{implies}(x,\mathsf{0})),\mathsf{0}))] =_{[0,1]} [\sigma(1)] \tag{36}$$

holds as well. We also have

$$\forall x \in [0,1]([\mathsf{implies}(\mathsf{implies}(x,\mathsf{implies}(x,\mathsf{0})),\mathsf{0})] =_{[0,1]} [1] \Rightarrow [\mathsf{F}(x)] >_1 [\mathsf{F}(\mathsf{0})]) \tag{37}$$

because, for all $x \geq 0$,

$$\begin{aligned}[\mathsf{implies}(\mathsf{implies}(x,\mathsf{implies}(x,\mathsf{0})),\mathsf{0})] &= max(1-max(1-x,max(1-x,0)),0)\\ &= max(1-max(1-x,1-x),0)\\ &= max(1-(1-x),0)\\ &= max(x,0)\\ &= x\end{aligned}$$

[3] For the purpose of this paper, the procedure to obtain this new rule is not relevant. The interested reader can find the details in [13].

and hence $[\mathsf{implies}(\mathsf{implies}(x, \mathsf{implies}(x, 0)), 0)] =_{[0,1]} [1]$ holds only if $x = 1 = [1]$. Combining (36) and (37), we conclude that, for all substitutions σ, if $\sigma(\mathsf{implies}(\mathsf{implies}(x, \mathsf{implies}(x, 0)), 0)) \to^*_{\mathcal{R}} \sigma(1)$ holds, then $[\mathsf{F}(x)]^{\mathcal{M}} = 1 >_1 0 = [\mathsf{F}(0)]^{\mathcal{M}}$, as desired. This proves operational termination of $\mathcal{R}$ in Example 1.

In the following section we discuss an interesting application of convex polytopic domains to improve the well-known matrix interpretations [5, 2].

4 Convex matrix interpretations

A *convex matrix intepretation* for a k-ary symbol f is a linear expression $F_1\boldsymbol{x}_1 + \cdots + F_k\boldsymbol{x}_k + F_0$, where $F_1, \ldots, F_k \in \mathbb{R}^{n\times n}$ are (square) matrices, $F_0 \in \mathbb{R}^n$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \in \mathbb{R}^n$, which is closed on $D(A, b)$, i.e., that satisfies

$$\forall \boldsymbol{x}_1, \ldots \boldsymbol{x}_k \in \mathbb{R}^n \left(\bigwedge_{i=1}^{k} A\boldsymbol{x}_i \geq b \Rightarrow A(F_1\boldsymbol{x}_1 + \cdots + F_k\boldsymbol{x}_k + F_0) \geq b \right) \tag{38}$$

An $\mathcal{F}$-algebra $\mathcal{A} = (\mathsf{D}, \mathcal{F}_{\mathsf{D}})$ is obtained if $\mathsf{D} = D(A, b)$, and each k-ary symbol $f \in \mathcal{F}$ is given $f_{\mathcal{A}}(x_1, \ldots, x_k) = F_1x_1 + \cdots + F_kx_k + F_0$ that satisfies (38). The following ordering $\geq$ is considered: $\boldsymbol{x} = (x_1, \ldots, x_n) \geq (y_1, \ldots, y_n) = \boldsymbol{y}$ if $x_i \geq y_i$ for all $1 \leq i \leq n$. Given $\delta > 0$, the (strict) ordering $>_\delta$ is also used: $\boldsymbol{x} = (x_1, \ldots, x_n) >_\delta (y_1, \ldots, y_n) = \boldsymbol{y}$ if $x_1 - y_1 \geq \delta$ and $(x_2, \ldots, x_n) \geq (y_2, \ldots, y_n)$.

Remark 2. Convex matrix interpretations include the usual matrix interpretations in [5, 2] if $A = I_{n\times n}$ and $\boldsymbol{b} = \boldsymbol{0} \in \mathbb{R}^n$.

In contrast to $(\mathbb{N}, \geq)$ and $([0, +\infty), \geq)$, that are *total* orders, and also to $(\mathbb{N}^n, \geq)$ and $([0, +\infty)^n, \geq)$, that are not total, but are complete lattices, $(D(A, b), \geq)$ does *not* need to be total or a complete lattice. This has some interesting advantages.

Example 10. Consider the CTRS $\mathcal{R}$ [17, Example 7.2.45]:

$$a \to a \Leftarrow b \to x, c \to x \tag{39}$$

$$b \to d \Leftarrow d \to x, e \to x \tag{40}$$

$$c \to d \Leftarrow d \to x, e \to x \tag{41}$$

According to [13], $\mathcal{R}$ is operationally terminating if there is a relation $\gtrsim$ such that $\to^* \subseteq \gtrsim$, and $\sqsupset$ is a well-founded ordering such that $\gtrsim \circ \sqsupset \subseteq \sqsupset$ and for the *dependency pair* $a^\sharp \to a^\sharp \Leftarrow b \to x, c \to x$ (for $a^\sharp$ a new symbol), we have that, for all substitutions σ, if $b \to^* \sigma(x)$ and $c \to^* \sigma(x)$, then $a^\sharp \sqsupset a^\sharp$. With $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\boldsymbol{b} = (1, 0, 0)^T$, together with:

$$[a] = [a^\sharp] = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad [b] = [d] = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad [c] = [e] = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

we have $[a], [a^\sharp], [b], [c], [d], [e] \in D(A, \boldsymbol{b})$, as required by (38). It can be proved that $(D(A, \boldsymbol{b}), \mathcal{F}_{D(A,\boldsymbol{b})}, \Pi_{D(A,\boldsymbol{b})})$, where $\to, \to^* \in \Pi$ are both interpreted (in

$\mathit{\Pi}_{D(A,\boldsymbol{b})}$) as $\geq$ is a *model* of $\mathcal{R}$. For $s,t \in \mathcal{T}(\mathcal{F},\mathcal{X})$, we let $s \gtrsim t$ if and only if $[s] \geq [t]$. Thus, $\to^* \subseteq \gtrsim$ holds. The ordering $>_1$ on $D(A,b)$ is *well-founded* because $[0,+\infty)$ is bounded from below (see [10]). Thus, for $s,t \in \mathcal{T}(\mathcal{F},\mathcal{X})$, we define $s \sqsupset t$ if and only if $[s] >_1 [t]$. Now, since $\to^* \subseteq \gtrsim$, we only have to prove that $[b] \geq [x] \wedge [c] \geq [x] \Rightarrow [a^\sharp] >_1 [a^\sharp]$, i.e.,

$$\forall x_1,x_2 \in \mathbb{R} \left(\begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \wedge \begin{bmatrix} 1 \\ 0 \end{bmatrix} \geq \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \wedge \begin{bmatrix} 0 \\ 1 \end{bmatrix} \geq \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix} >_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \quad (42)$$

which can be written as a universally quantified conjunction of two formulas:

$$x_1 + x_2 \geq 1 \wedge x_1 \geq 0 \wedge x_2 \geq 0 \wedge 1 \geq x_1 \wedge 0 \geq x_2 \wedge 0 \geq x_1 \wedge 1 \geq x_2 \Rightarrow 1 >_1 1 \quad (43)$$

$$x_1 + x_2 \geq 1 \wedge x_1 \geq 0 \wedge x_2 \geq 0 \wedge 1 \geq x_1 \wedge 0 \geq x_2 \wedge 0 \geq x_1 \wedge 1 \geq x_2 \Rightarrow 0 \geq 0 \quad (44)$$

The crucial point is that the conditional part of the implications does not hold because no $\boldsymbol{x} \in D(A,b)$ satisfies $(1,0)^T \geq \boldsymbol{x}$ and $(0,1)^T \geq \boldsymbol{x}$ (see Example 11).

The following sections discuss existing mathematical techniques that can be used to automatically deal with the conditional constraints obtained so far.

5 Conditional polynomial constraints

In this section, we explore well-known results from linear algebra [20] and algebraic geometry [18] to deal with conditional polynomial constraints.

5.1 Conditional constraints with linear polynomials

Farkas' Lemma provides a *(universal) quantifier elimination* result for linear (conditional) sentences (cf. [20]).

Theorem 1 (Affine form of Farkas' Lemma). *Let $A\boldsymbol{x} \geq \boldsymbol{b}$ be a linear system of k inequalities and n unknowns over the real numbers with non-empty solution set S and let $\mathbf{c} \in \mathbb{R}^n$ and $\beta \in \mathbb{R}$. Then, the following statements are equivalent:*

1. $\mathbf{c}^T\boldsymbol{x} \geq \beta$ *for all* $\boldsymbol{x} \in S$,
2. $\exists \boldsymbol{\lambda} \in \mathbb{R}^k_0$ *such that* $\mathbf{c} = A^T\boldsymbol{\lambda}$ *and* $\boldsymbol{\lambda}^T\mathbf{b} \geq \beta$.

By condition (1) in Theorem 1 proving $\forall \boldsymbol{x}\,(A\boldsymbol{x} \geq \boldsymbol{b} \Rightarrow c^T\boldsymbol{x} \geq \beta)$ can be recast as the *constraint solving problem* of finding a nonnegative vector $\boldsymbol{\lambda}$ such that $\boldsymbol{c}$ is a linear nonnegative combination of the *rows* of A and β is smaller than the corresponding linear combination of the components of $\boldsymbol{b}$. Note that if $A\boldsymbol{x} \geq \boldsymbol{b}$ has no solution, i.e., S in Theorem 1 is *empty*, the conditional sentence trivially holds. Thus, we do not need to check S for emptiness when using Farkas' result.

Example 11. Sentences (43) and (44) can be proved using Theorem 1. This proves operational termination of $\mathcal{R}$ in Example 10.

Example 12. After encoding the equality as the conjunction of $\geq$ and $\leq$, we transform sentences (29), (31) and (32) into:

$$\forall x \in \mathbb{R}\,(\, x \geq 0 \wedge 1 \geq x \wedge 0 \geq x \Rightarrow 0 \geq x) \tag{45}$$
$$\forall x \in \mathbb{R}\,(\, x \geq 0 \wedge 1 \geq x \wedge 0 \geq x \Rightarrow x \geq 0) \tag{46}$$
$$\forall x, y \in \mathbb{R}\,(\, x \geq 0 \wedge 1 \geq x \wedge y \geq 0 \wedge 1 \geq y \wedge 1 - x = 1 \wedge 1 - x \geq y \Rightarrow 1 - x \geq 1) \tag{47}$$
$$\forall x, y \in \mathbb{R}\,(\, x \geq 0 \wedge 1 \geq x \wedge y \geq 0 \wedge 1 \geq y \wedge 1 - x = 1 \wedge 1 - x \geq y \Rightarrow 1 \geq 1 - x) \tag{48}$$
$$\forall x, y \in \mathbb{R}\,(\, x \geq 0 \wedge 1 \geq x \wedge y \geq 0 \wedge 1 \geq y \wedge 1 - x = 1 \wedge y > 1 - x \Rightarrow y \geq 1) \tag{49}$$
$$\forall x, y \in \mathbb{R}\,(\, x \geq 0 \wedge 1 \geq x \wedge y \geq 0 \wedge 1 \geq y \wedge 1 - x = 1 \wedge y > 1 - x \Rightarrow 1 \geq y) \tag{50}$$

which are conditional linear sentences provable using Farkas' Lemma.

5.2 Conditional constraints with arbitrary polynomials

Given polynomials $h_1, \ldots, h_m \in \mathbb{R}[X_1, \ldots, X_n]$, the *semialgebraic set* defined by $h_1, \ldots, h_m$ is $W_\mathbb{R}(h) = W_\mathbb{R}(h_1, \ldots, h_m) = \{x \in \mathbb{R}^n \mid h_1(x) \geq 0 \wedge \cdots \wedge h_m(x) \geq 0\}$. A well-known *representation theorem* establishes that a polynomial which is positive for all tuples $(x_1, \ldots, x_n) \in W_\mathbb{R}(h)$ can be written as a linear combination of $h_1, \ldots, h_m$ with 'coefficients' s that are sums of squares of polynomials ($s \in \sum \mathbb{R}[X]^2$) [18, Theorem 5.3.8]. If we can write a polynomial f as a linear combination of $h_1, \ldots, h_m$ with 'coefficients' that are sums of squares, this provides a *certificate* of non-negativeness of f on $W_\mathbb{R}(h_1, \ldots, h_m)$: sums of squares are non-negative, all h_i are non-negative on values in $W_\mathbb{R}(h_1, \ldots, h_m)$ and the product and addition of non-negative numbers is non-negative. Explicitly:

Theorem 2. *Let $\mathbb{R}[X] := \mathbb{R}[X_1, \ldots, X_n]$, $h_1, \ldots, h_m \in \mathbb{R}[X]$, $W_\mathbb{R}(h) = W_\mathbb{R}(h_1, \ldots, h_m)$ and $S \subseteq \mathbb{R}$ such that $W_\mathbb{R}(h_1, \ldots, h_m) \subseteq S^n$. Let $s_i \in \sum \mathbb{R}[X]^2$ for all i, $0 \leq i \leq m$. If for all $x_1, \ldots, x_n \in S$, $f \geq s_0 + \sum_{i=1}^{m} s_i \cdot h_i$, then, for all $(x_1, \ldots, x_n) \in W_\mathbb{R}(h_1, \ldots, h_m)$, $f(x_1, \ldots, x_n) \geq 0$.*

Example 13. Consider the constraint $X_1 \geq X_2^2 \wedge X_2 \geq X_3^2 \Rightarrow X_1 \geq X_3^4$ from [16, page 51]. With $s_0 = (X_3^2 - X_2)^2$, $s_1 = 1$ and $s_2 = 2X_3^2$, we have:

$$X_1 - X_3^4 = (X_3^2 - X_2)^2 + (X_1 - X_2^2) + 2X_3^2 \cdot (X_2 - X_3^2)$$

witnessing that the constraint holds.

6 Related work

The material in Section 2 can be thought of as a generalization and extension of the *intepretation method* for proving termination of Term Rewriting Systems (see, e.g., [17, Section 5]). The interpretation method uses *ordered algebras* which are algebras $\mathcal{A}$ with domain D including one or more ordering relations $\succeq_\mathsf{D}$, $\succ_\mathsf{D}$, etc., satisfying a number of properties (stability, monotonicity, etc.). Such relations are used to *induce* relations $\succeq$, $\succ$ on terms which are then used to compare the left- and right-hand sides ℓ and r of rewrite rules $\ell \to r$. The targeted rules in such comparisons and the conclusions we may reach depend on the considered approach for proving termination (see [17, Sections 5.2 and 5.4], for instance). In our setting, orderings are introduced as interpretations of computational relations (e.g., $\to$ and $\to^*$), and we do not require anything special about them beyond their ability to provide a *model* of the theory at hand. For instance, where the interpretation method requires *monotonicity*, we just expect the relation to provide a model of rules (*Cong*), which encode the monotonicity of the rewrite relation. The advantage is that we do not need reformulations of the framework when other logics are considered; in contrast, the interpretation method requires explicit adaptations. For instance, in *Context-Sensitive Rewriting* [9] rewritings are propagated to selected arguments of function symbols only. Thus, (*Cong*) may have *no specialization* for *some* arguments i of some symbols f. Whereas this requires specific adaptations of the interpretation method (see, e.g., [21]), we can apply our methods without any change. Furthermore, although our practical examples involve CTRSs, our development does not really depend on that and applies to arbitrary declarative languages.

With regard to existing approaches to deal with conditional constraints in proofs of termination, the following result formalizes the transformational approach to deal with polynomial *conditional constraints* in [6, 16].

Proposition 1. [16, Proposition 3] *Let prem and conc be two polynomials with natural coefficients, where conc is not a constant. Let* $p_1, \ldots, p_{m+1}, q_1, \ldots, q_{m+1}$ *be arbitrary polynomials with natural coefficients. If*

$$conc(p_{m+1}) - conc(q_{m+1}) - prem(p_1, \ldots, p_m) + prem(q_1, \ldots, q_m) \geq 0$$

is valid over the natural numbers, then $p_1 \geq q_1 \wedge \cdots \wedge p_m \geq q_m \Rightarrow p_{m+1} \geq q_{m+1}$ *is also valid over the natural numbers.*

This result holds if *prem* and *conc* have non-negative real coefficients, and variables range over nonnegative real numbers. When linear polynomials are used this technique is subsumed by Farkas' lemma.

Proposition 2. *Let* $C \in \mathbb{R}_{\geq 0}[Y]$ *and* $P \in \mathbb{R}_{\geq 0}[Y_1, \ldots, Y_m]$ *be linear applications with* C *nonconstant, i.e.,* $C = \gamma Y$ *with* $\gamma > 0$ *and* $P = \sum_{i=1}^{m} \pi_i Y_i$. *Let* $p_i, q_i \in \mathbb{R}_{\geq 0}[X_1, \ldots, X_n]$ *be linear polynomials for all* i, $1 \leq i \leq m+1$, *i.e.,* $p_i = p_{i0} + \sum_{j=1}^{n} p_{ij} X_j$ *and* $q_i = q_{i0} + \sum_{j=1}^{n} q_{ij} X_j$. *Let* $A = (p_{ij} - q_{ij})_{m,n}$, $\boldsymbol{b} = (q_{10} - p_{10}, \ldots q_{m0} - p_{m0})^T$, $\boldsymbol{c} = (p_{m+1,1} - q_{m+1,1}, \ldots, p_{m+1,n} - q_{m+1,n})^T$ *and* $\beta = q_{m+1,0} - p_{m+1,0}$. *If for all* $X_1, \ldots, X_m \geq 0$, $C(p_{m+1}) - C(q_{m+1}) - P(p_1, \ldots, p_m) + P(q_1, \ldots, q_m) \geq 0$, *then there is* $\boldsymbol{\lambda} \in \mathbb{R}_0^m$ *such that* $\boldsymbol{c} \geq A^T \boldsymbol{\lambda}$ *and* $\beta \leq \boldsymbol{\lambda}^T \boldsymbol{b}$.

Remark 3. Regarding mechanization, Nguyen et al.'s technique has a drawback with respect to those in Section 5. Given a rule $\ell \to r \Leftarrow \bigwedge_{i=1}^{n} s_i \to t_i$, Nguyen et al.'s technique requires that both $[s_i]$ and $[t_i]$ are polynomials *with non-negative coefficients only*. This is because $[s_i]$ and $[t_i]$ are handled separately by polynomials *conc* and *prem*. But in an implementation, $[s_i]$ and $[t_i]$ are *parametric polynomials* where the coefficients are *parameters* rather than numbers (see [4, 10] for instance). Thus, we need to *constrain* them to be *non-negative* in order to use the technique. In contrast, we do not restrict the coefficients of polynomials in any way. Hence, the coefficients of the parametric polynomials could be negative numbers without any problem. For instance, this is crucial to synthesize $D(A, \boldsymbol{b}) = [0, 1]$ used in the examples above, where A and $\boldsymbol{b}$ require negative numbers.

Farkas' Lemma is used in proofs of termination of *imperative* programs in [19].

7 Conclusion

We have provided a generic, logic-oriented approach to abstraction in proofs of termination of programs in declarative languages, which is based on defining appropriate *models* for logics. We have used numeric domains defined as *restrictions* of 'big' numeric sets by means of predicates that can be handled as conditional constraints. We have introduced *convex* domains and used them to extend the powerful *matrix interpretation method* for proving termination of TRSs in two directions: the use of *convex* domains and the application to other logics (e.g., CTRSs). We have shown the usefulness of these general purpose ideas by applying them to prove operational termination of CTRSs: $\mathcal{R}$ in Example 1 could *not* be handled within the recently introduced 2D DP framework for proving operational termination of CTRSs [13] or its extensions [14]; but the weakness was not in the framework itself, but in the available algebraic interpretations: we can prove $\mathcal{R}$ operationally terminating now due to the use of a convex domain like $[0, 1]$. And powerful tools like AProVE do not find a proof of operational termination of $\mathcal{R}$ in Example 10 by using transformations. In contrast, we found a simple proof with convex matrix interpretations and the techniques in [13].

We have shown that existing, powerful techniques to deal with numeric constraints provide an appropriate framework for implementing the previous techniques. We have implemented most of these techniques as part of our tool MU-TERM [1]. In particular, the use of Farkas' Lemma for dealing with linear conditional constraints obtained from linear polynomial interpretations and matrix interpretations plays a central role in the implementation of the 2D DP framework for operational termination of CTRSs [13] which is presented in [14]. In [10, Example 13], we advocate the use of *negative* coefficients in proofs of termination of *CSR* using polynomial interpretations. The implementation, though, was tricky (see [10, Sections 6.1.3 and 7]). This paper is a step forward because: (1) our treatment is valid for arbitrary polynomials. We do not need to provide special results as [10, Observation 1] to deal with polynomials of some specific form (quadratic, cubic, ...); (2) we avoid the introduction of disjunctive constraints which lead to an exponential blowup and to an expensive constraint solving process; and (3) we admit negative numbers everywhere. They are treated as any other number and there is no need to 'assert' which of the coefficients could be negative in order to handle them apart (see [10, Section 7] and [10, Example 20]). However, much work is necessary to make fully general use of these techniques in practical applications. We plan to address these issues in the near future.

References

1. B. Alarcón, R. Gutiérrez, S. Lucas, R. Navarro-Marset. Proving Termination Properties with MU-TERM. In *Proc. of AMAST'10*, LNCS 6486:201-208, 2011.
2. B. Alarcón, S. Lucas, R. Navarro-Marset. Using Matrix Interpretations over the Reals in Proofs of Termination. In *Proc. of PROLE'09.* pages 255-264, 2009.
3. M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. Talcott. All About Maude – A High-Performance Logical Framework. Lecture Notes in Computer Science 4350, 2007.
4. E. Contejean, C. Marché, A.-P. Tomás, and X. Urbain. Mechanically proving termination using polynomial interpretations. *J. of Aut. Reas.*, 34(4):325-363, 2006.
5. J. Endrullis, J. Waldmann, and H. Zantema. Matrix Interpretations for Proving Termination of Term Rewriting. *J. of Aut. Reas.* 40(2-3):195-220, 2008.
6. C. Fuhs, J. Giesl, A. Middeldorp, P. Schneider-Kamp, R. Thiemann, and H. Zankl. Maximal Termination. In *Proc. of RTA'08*, LNCS 5117:110-125, 2008.
7. K. Futatsugi and R. Diaconescu. *CafeOBJ Report.* World Scientific, AMAST Series, 1998.
8. P. Hudak, S.J. Peyton-Jones, and P. Wadler. Report on the Functional Programming Language Haskell: a non–strict, purely functional language. *Sigplan Notices*, 27(5):1-164, 1992.
9. S. Lucas. Context-sensitive computations in functional and functional logic programs. *Journal of Functional and Logic Programming*, 1998(1):1-61, January 1998.
10. S. Lucas. Polynomials over the reals in proofs of termination: from theory to practice. *RAIRO Theoretical Informatics and Applications*, 39(3):547-586, 2005.
11. S. Lucas, C. Marché, and J. Meseguer. Operational termination of conditional term rewriting systems. *Information Processing Letters*, 95:446–453, 2005.
12. S. Lucas and J. Meseguer. Proving Operational Termination Of Declarative Programs In General Logics. In *Proc. of PPDP'14*, pages 111-122, ACM Digital Library, 2014.
13. S. Lucas and J. Meseguer. 2D Dependency Pairs for Proving Operational Termination of CTRSs. In *Proc. of WRLA'14*, LNCS vol. 8663, to appear, 2014.
14. S. Lucas, J. Meseguer, and R. Gutiérrez. Extending the 2D DP Framework for CTRSs. In *Selected papers of LOPSTR'14*, LNCS *to appear*, 2015.
15. J. Meseguer. General Logics. In H.-D. Ebbinghaus et al., editors, *Logic Colloquium'87*, pages 275-329, North-Holland, 1989.
16. M.T. Nguyen, D. de Schreye, J. Giesl, and P. Schneider-Kamp. Polytool: Polynomial interpretations as a basis for termination of logic programs. *Theory and Practice of Logic Programming* 11(1):33-63, 2011.
17. E. Ohlebusch. *Advanced Topics in Term Rewriting.* Springer-Verlag, Apr. 2002.
18. A. Prestel and C.N. Delzell. Positive Polynomials. From Hilbert's 17th Problem to Real Algebra. Springer-Verlag, Berlin, 2001.
19. A. Podelski and A. Rybalchenko. A Complete Method for the Synthesis of Linear Ranking Functions, In *Proc. of VMCAI'04*, LNCS 2937:239 - 251, 2004.
20. A. Schrijver. Theory of linear and integer programming. John Wiley & sons, 1986.
21. H. Zantema. Termination of Context-Sensitive Rewriting. In *Proc. of RTA'97*, LNCS 1232:172-186, Springer-Verlag, Berlin, 1997.