Automated Termination in Model Checking Modulo Theories

A. Carioni[1], S. Ghilardi[1], and S. Ranise[2]

[1] Università degli Studi di Milano, Milano, Italia
[2] FBK-Irst, Trento, Italia

Abstract. We use a declarative SMT-based approach to model-checking of infinite state systems to design a procedure for automatically establishing the termination of backward reachability by using well-quasi-orderings. Besides showing that our procedure succeeds in many instances of problems covered by general termination results, we argue that it could predict termination also on single problems outside the scope of applicability of such general results.

1 Introduction

Infinite state model checking is nowadays a mature field and many successful attempts have been made to verify disparate problems, using various kinds of methodologies. Still, termination of search (both forward and backward) is a major problem and it is sometimes difficult to predict *in advance* whether a given problem will be solved in a finite amount of time or some source of divergence is hidden somewhere and appropriate techniques (such as acceleration or abstraction) should be employed in order to make the search terminating. In the literature, several results for the termination of backward reachability are known covering entire classes of infinite state systems [13,11,5,2,6]. In most cases, following the seminal work in [1], these results are based on the use of well-quasi-orders on configurations, which are (finite) symbolic representations of infinite sets of backward reachable states. Thus, their applicability crucially rely on the human ability to reformulate a given specification so that it fits in one of the classes of systems for which termination is guaranteed.

In this paper, we propose an *automated* technique capable (when successful) of predicting termination from the static analysis of a given verification problem that is amenable to backward reachability. We develop our ideas in the model checking modulo theories framework [14,18] where array-based (guarded assignment) transition systems are used to symbolically specify a wide range of systems by using certain classes of first-order formulae and background theories. There are two main ingredients. The former is that of a *wqo-theory* W, which is the declarative counterpart of a well-quasi-order on configurations used in the main arguments for termination in [1]. The second ingredient is the standard notion in first-order logic textbooks (see, e.g., [12]) of *syntactic interpretation*. Then, we cook these two ingredients together to design a method for establishing the termination of backward reachability on a given verification problem by checking for the *existence* of a syntactic translation from W, satisfying certain conditions. Such conditions refer to a search space *restricted to formulae describing small models* and the possibility of such restriction is indeed the essential content of our main result (Theorem 4.5 below). Interestingly, the conditions of Theorem 4.5 can all be checked trough proof obligations that can be efficiently discharged by using SMT solving techniques. We shall turn to an informal discussion on our main issue at the beginning of Section 4.

For space constraints, we shall supply here only high level explanations and formal statements of our results: the interested reader is referred to the online available extended version [8] for worked out examples and full proofs.

2 Preliminaries

We assume the usual syntactic (e.g., signature, variable, term, ground term, atom, literal, formula, and sentence) and semantic (e.g., structure, sub-structure, embedding, assignment, truth, and validity) notions of many-sorted first-order logic (see, e.g., [12,10]). The equality symbol $=$ is included in all signatures considered below. We use $L, M, \dots$ for literals and $\phi, \psi, \dots$ for formulae. A signature is *relational* iff it does not contain function symbols and it is *quasi-relational* iff the only function symbols it contains are constants. A formula is *open* (or *quantifier-free*) iff it does not contain quantifiers; it is *universal* (resp. *existential*) iff it is obtained from an open formula by prefixing it a finite sequence of universal (resp. existential) quantifiers. If $\phi(\underline{x})$ is a formula with free variables included in the tuple $\underline{x} = x_1, \dots, x_n$ and $\underline{a} = a_1, \dots, a_n$ is a (sort-conforming) tuple of elements of the support $|\mathcal{M}|$ of a structure $\mathcal{M}$, we write $\mathcal{M} \models \phi(\underline{a})$ to denote that $\phi(\underline{x})$ is valid in $\mathcal{M}$ under the assignment $\{x_1 \mapsto a_1, \dots, x_n \mapsto a_n\}$.
SMT. Following [22], a *theory* T is a pair $(\Sigma, \mathcal{C})$, where Σ is a signature and $\mathcal{C}$ is a class of Σ-structures; the structures in $\mathcal{C}$ are the *models* of T. Below, let $T = (\Sigma, \mathcal{C})$. A Σ-formula ϕ is *T-satisfiable* if there exists a Σ-structure $\mathcal{M}$ in $\mathcal{C}$ such that ϕ is true in $\mathcal{M}$ under a suitable assignment to the free variables of ϕ (in symbols, $\mathcal{M} \models \phi$); it is *T-valid* (in symbols, $T \models \varphi$) if its negation is T-unsatisfiable. Two formulae φ_1 and φ_2 are *T-equivalent* if $\varphi_1 \leftrightarrow \varphi_2$ is T-valid. The *quantifier-free satisfiability modulo the theory T ($SMT(T)$) problem* amounts to establishing the T-satisfiability of quantifier-free Σ-formulae. A theory is said to be *syntactically specified* if we are given a set of Σ-sentences (called the *axioms* of T): in this case, the class $\mathcal{C}$ of the models of T is formed by the Σ-structures in which all the axioms of T are true. A theory T is *universal* iff it is syntactically specified and its axioms are universal sentences.
Diagrams. Given a (finite, in our applications) Σ-structure $\mathcal{M}$, take a free variable x_a for every a in the support of $\mathcal{M}$ and call $\underline{x}$ the set of all x_a (varying a). The *Σ-diagram $\delta_{\mathcal{M}}$ of $\mathcal{M}$* [10] is the set of all Σ-literals $L(\underline{x})$ such that $\mathcal{M}, \mathtt{a} \models L$, where $\mathtt{a}$ is the assignment mapping x_a to a. By abuse of notation, we shall confuse the variable x_a with the element a. Intuitively, we can view $\delta_{\mathcal{M}}$ as a sort of 'multiplication table' of the structure $\mathcal{M}$. Notice that the diagram

of a finite structure is also finite and can be seen as the formula obtained by the conjunction of its literals.

Interpretations. Informally, an interpretation $(-)^*$ of a Σ-theory T into a Σ'-theory T' is a mapping from the expressions of T to the expressions of T' which preserves the validity of sentences. We will consider a special class of interpretations, generalizations of our definition exist—see, e.g., [12]—but we do not need them here. Formally, $(-)^*$ is a mapping associating (i) a sort S^* of Σ' with each sort S of Σ, (ii) a Σ'-formula $R^*(x_1, \ldots, x_n)$ with each predicate symbol R of Σ in such a way that the variables $x_1, \ldots, x_n$ occurring free in R^* match the translations of the arity sorts of R (implicitly we assume that identity of sort S is translated into identity of sort S^*), and (iii) a Σ'-term $f^*(x_1, \ldots, x_n)$ with each function symbol f of Σ (with the same condition on $x_1, \ldots, x_n$ as for predicate symbols). Then, $(-)^*$ can be extended inductively to formulae in the obvious way: $(R(t_1, \ldots, t_n))^* = R^*(t_1^*/x_1, \ldots, t_n^*/x_n)$ for atomic formulae and $(A \wedge B)^* = A^* \wedge B^*, (\neg A)^* = \neg A^*, (\forall x A)^* = \forall x A^*$ for non-atomic formulae. The last requirement that $(-)^*$ is supposed to satisfy is the following: (iv) for every sentence ϕ in the signature of T, if $T \models \phi$ then $T' \models \phi^*$. Notice that, if T is specified syntactically, it is sufficient to check (iv) only for the axioms of T. In this paper, we shall limit ourselves to *quantifier-free* translations, i.e. to translations in which the formulae R^* mentioned in (ii) above are quantifier-free.

3 Array-based systems and backward reachability

Array-based transition systems [14,19,16,18] have been proved useful for the verification of several classes of infinite state systems, such as broadcast protocols, lossy channel systems, timed networks, parametric and distributed systems. The Model Checker Modulo Theories (MCMT) tool [20] implements symbolic backward reachability for array-based systems (its executable, several benchmark problems, the documentation, and related papers can be downloaded at `http://homes.dsi.unimi.it/~ghilardi/mcmt`). The state variables of array-based systems are arrays "connecting" the theories T_I and T_E: the former describes the topology and the latter the data structures of the system. We fix for the whole paper the following conventions:

- **(T1)** $T_I = (\Sigma_I, \mathcal{C}_I)$ is a relational mono-sorted theory whose unique sort is named `INDEX`), its $SMT(T_I)$-problem is decidable, and $\mathcal{C}_I$ is closed under substructures (meaning that $\mathcal{C}_I$ contains every substructure of any $\mathcal{M} \in \mathcal{C}_I$);
- **(T2)** T_E is a multi-sorted theory whose sorts are $\mathtt{ELEM}_1, ..., \mathtt{ELEM}_S$ for some $S \geq 1$ and its $SMT(T_E)$-problem is decidable;
- **(T3)** A_I^E is a compound theory, intended to describe 'arrays with indexes in T_I and elements in T_E.' Formally, the signature of A_I^E contains the sort symbols of $\Sigma_I \cup \Sigma_E$, together with a new sort symbol $\mathtt{ARRAY}_\ell$ for each sort $\mathtt{ELEM}_\ell$ of Σ_E; it contains also all the function and predicate symbols in $\Sigma_I \cup \Sigma_E$ together with a new function symbol $\_[\_]_\ell \ : \ \mathtt{ARRAY}_\ell, \mathtt{INDEX} \longrightarrow \mathtt{ELEM}_\ell$ for each sort $\mathtt{ELEM}_\ell$ of Σ_E. The models $\mathcal{M}$ of A_I^E are the structures whose Σ_I- and Σ_E-reducts are models of T_I and T_E, respectively; the sort $\mathtt{ARRAY}_\ell$ is

interpreted as the set of total functions $\mathtt{INDEX}^{\mathcal{M}} \longrightarrow \mathtt{ELEM}_{\ell}^{\mathcal{M}}$ and the symbol $\_[\_]_{\ell}$ as function application.

So, $a[i]_{\ell}$ denotes the element of sort $\mathtt{ELEM}_{\ell}$ stored in the array a of sort $\mathtt{ARRAY}_{\ell}$ at index i; the subscript ℓ is dropped whenever it is clear from the context.

Definition 3.1. *An* array-based (transition) system (for (T_I, T_E)) *is a triple* $\mathcal{S} = (a, I, \tau)$ *where* (i) $a = a_1, \ldots, a_s$ *is a tuple of free constants of array sorts (these are to be thought as* state variables *storing data of sorts* $\mathtt{ELEM}_1, \ldots, \mathtt{ELEM}_s$*, respectively);* (ii) $I(a)$ *is the* initial *formula;* (iii) $\tau(a, a')$ *is the* transition *formula, where* a' *contains the renamed copies of the variables in* a*. The formula* I *is assumed to be a* $\forall^I$*-formula and the formula* τ *is a disjunction* $\bigvee_{h=1}^{r} \tau_h$ *of guarded assignments in functional form.*

To give the definition of a $\forall^I$-formula and of a guarded assignment in functional form, we preliminarily introduce the following notational convention and definitions. Below, d, e range over variables of a sort $\mathtt{ELEM}_{\ell}$ of Σ_E, $i, j, k, z, \ldots$ over variables of sort $\mathtt{INDEX}$. An underlined variable name abbreviates a tuple of variables of unspecified (but finite) length and, if $\underline{i} := i_1, \ldots, i_n$, the notation $a[\underline{i}]$ abbreviates the $s * n$-tuple of terms $a_1[i_1], \ldots, a_s[i_1], \ldots, a_1[i_n], \ldots, a_s[i_n]$. Possibly sub/super-scripted expressions of the form $\phi(\underline{i}, \underline{e}), \psi(\underline{i}, \underline{e})$ denote *quantifier-free* $(\Sigma_I \cup \Sigma_E)$*-formulae in which at most the variables* $\underline{i} \cup \underline{e}$ *occur.* Also, $\phi(\underline{i}, \underline{t}/\underline{e})$ (or simply $\phi(\underline{i}, \underline{t})$) abbreviates the substitution of the Σ-terms $\underline{t}$ for the variables $\underline{e}$. Thus, for instance, $\phi(\underline{i}, a[\underline{i}])$ denotes the formula obtained by replacing $\underline{e}$ with $a[\underline{i}]$ in the quantifier-free formula $\phi(\underline{i}, \underline{e})$.

Given a theory T (in our case, T will be A_I^E), a *T-partition* is a finite set $C_1(\underline{x}), \ldots, C_n(\underline{x})$ of quantifier-free formulae (with free variables contained in the tuple $\underline{x}$) such that $T \models \forall \underline{x} \bigvee_{i=1}^{n} C_i(\underline{x})$ and $T \models \bigwedge_{i \neq j} \forall \underline{x} \neg (C_i(\underline{x}) \wedge C_j(\underline{x}))$. The formulae $C_1, \ldots, C_k$ are called the *components* of the T-partition. A *case-definable extension* $T' = (\Sigma', \mathcal{C}')$ of a theory $T = (\Sigma, \mathcal{C})$ is obtained from T by applying (finitely many times) the following procedure: (i) take a T-partition $C_1(\underline{x}), \ldots, C_n(\underline{x})$ together with Σ-terms $t_1(\underline{x}), \ldots, t_n(\underline{x})$; (ii) let Σ' be $\Sigma \cup \{F\}$, where F is a "fresh" function symbol (i.e. $F \notin \Sigma$) whose arity is equal to the length of $\underline{x}$; (iii) take as $\mathcal{C}'$ the class of Σ'-structures $\mathcal{M}$ whose Σ-reduct is a model of T and such that $\mathcal{M} \models \bigwedge_{i=1}^{n} \forall \underline{x}\, (C_i(\underline{x}) \to F(\underline{x}) = t_i(\underline{x}))$. Thus a case-definable extension T' of a theory T contains finitely many additional function symbols, called case-defined functions. By abuse of notation, below, we shall identify T with its case-definable extensions T'.

A formula $\forall \underline{i}.\phi(\underline{i}, a[\underline{i}])$ is a *$\forall^I$-formula*, one of the form $\exists \underline{i}.\phi(\underline{i}, a[\underline{i}])$ is an *$\exists^I$-formula*, and a sentence $\exists a\ \exists \underline{i}\ \forall \underline{j}\ \psi(\underline{i}, \underline{j}, a[\underline{i}], a[\underline{j}])$ is an *$\exists^{A,I}\forall^I$-sentence.* A *guarded assignment in functional form* is a formula of the form

$$\exists e \exists \underline{k}\, (\phi_L(e, \underline{k}, a[\underline{k}]) \wedge a' = \lambda j.F(e, \underline{k}, a[\underline{k}], j, a[j])) \qquad (1)$$

where: (i) $F = F_1, \ldots, F_s$ is a tuple of case-defined functions; (ii) the existentially quantified data variable e ranges over a sort $\mathtt{ELEM}_{\ell}$ such that T_E admits quantifier elimination with respect to quantified variables of sort $\mathtt{ELEM}_{\ell}$.

Given an array-based system $\mathcal{S} = (a, I, \tau)$ and an $\exists^I$-formula $U(a)$ describing a set of *unsafe* states (also called *error* states), the symbolic backward reachability procedure iteratively computes the set of backward reachable states $BR(a)$ as follows. (Below, we give a very high-level description of the symbolic backward reachability procedure implemented in MCMT. For a description of the techniques and heuristics used to make the procedure effective in practice, the reader is pointed to [19,17,16,18]. In particular, [18] reports an extensive experimental evaluation of the tool.) Preliminarily, define (for $n \geq 0$) the *n-pre-image* of a formula $K(a)$ as $Pre^0(\tau, K) := K$ and $Pre^{n+1}(\tau, K) := Pre(\tau, Pre^n(\tau, K))$, where $Pre(\tau, K) := \exists a'.(\tau(a, a') \wedge K(a'))$. Intuitively, $Pre^n(\tau, U)$ describes the set of backward reachable states in $n \geq 0$ steps. At the n-th iteration, the *backward reachability procedure* computes the formula $BR^n(\tau, U) := \bigvee_{i=0}^{n} Pre^i(\tau, U)$ representing the set of states which are backward reachable from the states in U with at most n steps. While computing $BR^n(\tau, U)$, the procedure also checks whether the system is unsafe by establishing if the formula $I \wedge Pre^n(\tau, U)$ is A_I^E-satisfiable (*safety test*) or whether a fix-point has been reached by checking if $(BR^n(\tau, U) \rightarrow BR^{n-1}(\tau, U))$ is A_I^E-valid or, equivalently, if the formula $BR^n(\tau, U) \wedge \neg BR^{n-1}(\tau, U)$ is A_I^E-unsatisfiable (*fix-point test*).

To mechanize this procedure, it is mandatory to identify a class of formulae for representing sets of backward reachable states which is closed under preimage computation and such that the safety and fix-point checks are decidable. Using $\exists^I$-formulae for representing unsafe states, this is indeed the case as stated in the following theorem, a corollary of more general results in [14,18].

Theorem 3.2. *Let $\mathcal{S} = (a, I, \tau)$ be an array-based system; we have that*

(RF1) *if K is an $\exists^I$-formula, then $Pre(\tau, K)$ is equivalent to an (effectively computable) $\exists^I$-formula;*

(RF2) *if the set U of unsafe states is represented by an $\exists^I$-formula, then the A_I^E-satisfiability checks for safety and fix-point of the backward reachability procedure are effective.*

As shown in [14,19,18,9,7], this theorem allows the automated verification of reachability properties for several classes of systems (e.g., parameterised systems, timed networks, or fault-tolerant algorithms). For several of these problems, a (declarative reformulation of) the *approximate model technique* (see, e.g., [4]) is required as explained in [15]. **(RF2)** is a special case of the decidability of the A_I^E-satisfiability problem for $\exists^{A,I}\forall^I$-sentences in [18]. Assumptions (**T1**) on T_I are essential for **(RF2)** because, if they are dropped, undecidability of $\exists^{A,I}\forall^I$-sentences arises [18]. The proof of **(RF2)** in [18] consists of a decision procedure integrating quantifier-free SMT solving and quantifier instantiation. Powerful heuristics [17] are also crucial for implementation.

An important refinement of the backward reachability procedure above is to exploit invariants whenever they are available. An invariant $J(a)$ for the array-based system $\mathcal{S} = (a, I, \tau)$ is a $\forall^I$-formula such that (a) $A_I^E \models I(a) \rightarrow J(a)$ and (b) $A_I^E \models Pre(\tau, J) \rightarrow J(a)$. The requirement that $J(a)$ is a $\forall^I$-formula allows us to reduce conditions (a) and (b) to the A_I^E-satisfiability of $\exists^{A,I}\forall^I$-sentences, which is decidable because of (**T1**). Techniques for invariant synthesis are discussed in [18] and are also implemented in MCMT. Whenever an invariant J is known, we can replace A_I^E with $A_I^E \cup \{J\}$ in our satisfiability tests (e.g. for fix-point in the backward reachability procedure). The presence of invariants in such tests is often crucial either to greatly speed up the performances of MCMT or to obtain termination (see again [18] for details).

3.1 Closure under pre-image computation

The proof of **(RF1)** (Theorem 3.2) in, e.g., [14,18] consists of simple logical manipulations (this is a distinguishing feature of our approach). Here, we briefly discuss a variant of such proofs, whose details are needed to state the main result of this paper (see Theorem 4.5 below).

Let $\mathcal{S} = (a, I, \tau)$ be an array-based system, where τ is a finite disjunction of formulae $\tau_1, ..., \tau_r$ of the form (1). We consider only $Pre(\tau_h, K)$ since $Pre(\tau, K)$ is easly seen to be equivalent to the disjunction of $Pre(\tau_1, K), ..., Pre(\tau_r, K)$. Let us now focus on the definition of $F = F_1, ..., F_s$ in (1). Without loss of generality (since partitions admit common refinements), we assume that the A_I^E-partition $\{C_1(e, \underline{k}, a[\underline{k}], j, a[j]), \dots, C_m(e, \underline{k}, a[\underline{k}], j, a[j])\}$ is the same for each F_l ($l \in \{1, ..., s\}$). Thus, each case-defined function F_l can be written as

$$F_l(j, a[j]) := \texttt{case of}\{C_1(j, a[j]) : t_{l1}(j, a[j]);\ \cdots\ C_m(j, a[j]) : t_{lm}(j, a[j])\}, \quad (2)$$

for $l = 1, \dots, s$. (According to the definition of case-definable extension, the logical reading of the `case of` construct is the conjunction of the formulae $\bigwedge_{z=1}^m \forall \underline{j}.(C_z(j, a[j]) \to F_l(j, a[j]) = t_{lz})$ for each $l = 1, ..., s$.) Notice that $F_l, C_1, \dots, C_m, t_{l1}, \dots, t_{lm}$ depend not only on $j, a[j]$ but also on $e, \underline{k}, a[\underline{k}]$; to simplify notation, we omit these dependences in the rest of this paper. If $K(a)$ is the $\exists^I$-formula $\exists \underline{i}\, \psi(\underline{i}, a[\underline{i}])$, then $Pre(\tau_h, K)$ is logically equivalent to the formula obtained from

$$\phi_L \wedge \psi(\underline{i}, F[\underline{i}]) \quad (3)$$

by prefixing it with the existential quantifiers $\exists \underline{i}\, \exists \underline{k}\, \exists e$; here the notation $F[\underline{i}]$ abbreviates the $(n * s)$-tuple of terms $F_l(i_z, a[i_z])$, varying $l = 1, \dots, s$ and $z = 1, \dots, n$ when $\underline{i} = i_1, \dots, i_n$. We can further manipulate the formula (3) in order to eliminate the defined symbols $F_1, ..., F_s$. To do this, we consider the functions $f : \underline{i} \to \{1, \dots, m\}$ and rewrite the formula (3) as the disjunction (varying f) of the formulae

$$\begin{aligned} \tau_h[\psi, f] := \quad & \phi_L \wedge C_{f(i_1)}(i_1, a[i_1]) \wedge \cdots \wedge C_{f(i_n)}(i_n, a[i_n]) \wedge \\ & \wedge \psi(\underline{i}, t_{1f(i_1)}(i_1, a[i_1]), \dots, t_{sf(i_n)}(i_n, a[i_n])) \ . \end{aligned}$$

At this point, it is clear that $Pre(\tau_h, K)$ is equivalent to $\bigvee_f \exists \underline{i}\, \exists \underline{k}\, \exists e\, \tau_h[\psi, f]$, where the functions f indexing the disjunction will be called *case-marking functions* and their purpose is to mark each index in $\underline{i}$ with the case that formally applies to it. This concludes the proof of (**RF1**).

4 Wqo-theories, QE-degrees, and Termination

The assumptions of Theorem 3.2 guarantee that the backward reachability procedure described in the previous section can be mechanized but they are not sufficient to guarantee termination. This is because the symbolic representation of pre-images (in our case, $\exists^I$-formulae) may not be expressive enough to represent a fix-point of the set of backward reachable states. Termination can be achieved only under additional assumptions. The classical (non declarative) method for obtaining termination (see, e.g., [1,5,6]) consists in endowing the states of the system with a preorder relation $\preceq$ and in assuming that (1) pre-images are monotonic w.r.t. $\preceq$ and (2) $\preceq$ is a well-quasi-ordering (wqo). Now, (1) implies that the pre-image of an upward-closed (w.r.t. $\preceq$) set is still upward-closed and (2) implies that every upward-closed set can be characterized by a finite set of minimal (w.r.t. $\preceq$) elements. Thus, starting from an upward-closed set U of states, the iterative computation of the backward reachable configurations from U necessarily terminates because the fixpoint is upward closed and hence its minimal elements are rechable in finitely many steps. Obviously, this requires that relevant upward-closed sets can be effectively represented and manipulated. Our goal here is to recast this argument in our declarative framework underlying the backward reachability procedure of Section 3. Roughly, our plan is as follows: without loss of generality (see [18], Section 4), states of the system can be identified with the values assigned to the array constants a in a model $\mathcal{M}$ having finitely many generators of sort `INDEX`. Since we represent backward reachable states by $\exists^I$-formulae, we replace $\preceq$ by the embeddability relation between such finitely generated models. The fact that existential formulae are preserved by super-structures guarantees that they describe upward closed sets of states. However, it would be too strong to require embeddability among finitely generated models to be a wqo: instead, we make an abstraction of the system, through a wqo-theory W and a syntactic translation into A_I^E. This will replace embeddability between finitely generated models of A_I^E by *embeddability between the abstract states*, i.e. between the finitely generated models of W. The definition of a wqo theory just says that embeddability between such abstract states is a wqo. The key step of our plan consists of checking whether every pre-image P_i computed by the backward reachability procedure is a translation of an existential formula β_i in W for $i \geq 0$ where $P_0 := U$. The sequence $\beta_0, \beta_1, ...$ is finite because it describes increasingly larger upsets of a wqo, hence the sequence $P_0, P_1, ...$ must be finite too. In fact, if there exists $b \geq 1$ such that β_b is a fix-point, i.e. $W \models \beta_b \to \beta_{b-1}$, by using the syntactic translation we must also have that $A_I^E \models P_b \to P_{b-1}$ where P_b and P_{b-1} are the translations of β_b and β_{b-1}, respectively. Hence P_b is a fixed point of our backward reachability procedure. Thus we must find a condition that guarantees that the backward reachability procedure generates only $\exists^I$-formulae which are translations of existential formulae of W: to this aim, we reduce the general case to finitely many cases involving formulae of the kind $\exists e\, \exists \underline{i}\, \exists \underline{k}\, \tau_h[\psi, f]$, *where ψ is the translation of the diagram of a "small" model of W*. Finally, since the elimination of the existentially quantified data variable e is required, we need also assumptions on the quantifier elimination algorithm.

Wqo-theories. A wqo $(P, \leq)$ is a set P endowed with a binary reflexive and transitive relation $\leq$ such that for every infinite sequence $p_1, p_2, \ldots$ of elements from P there are $i < j$ such that $p_i \leq p_j$.

Definition 4.1. *A wqo-theory is a universal theory whose finitely generated models*[3] *are a well-quasi-order with respect to the embeddability relation.*

Simple examples of wqo-theories can be obtained by taking vector spaces on a fixed field, torsion-free abelian groups, etc. The reason why we get a wqo in these cases is that an embedding always exists whenever the dimension is lower. Examples of wqo-theories which are more relevant to this paper can be obtained by re-interpreting declaratively some special cases of Kruskal theorem or Higman lemma, as sketched in the following example.

Example 4.2. Consider a signature contaning one sort, finitely many 0-ary and 1-ary predicates and a single binary predicate $\leq$ (besides equality). We get a wqo theory W by Higmann lemma if we syntactically specify W through the following set of axioms

$$\forall x\,(x \leq x), \quad \forall x, y, z\,(x \leq y \wedge y \leq z \to x \leq z), \text{ and } \forall x, y\,(x \leq y \vee y \leq x),$$

(the axioms say that $\leq$ is to be interpreted as a total pre-order relation).

QE-degree. A theory T admits *quantifier elimination* (relative to a sort S) iff for every formula $\varphi(\underline{x})$ containing only quantified variables of sort S, there exists a quantifier-free formula $\varphi'(\underline{x})$ such that $T \models \forall \underline{x}(\varphi(\underline{x}) \leftrightarrow \varphi'(\underline{x}))$.

A set $\mathcal{P}$ of Σ-predicates is said to be *T-representative* for a Σ-theory T iff for every Σ-literal $L(\underline{x})$ one can compute a formula $\psi(\underline{x})$ which is a positive combination (i.e. a disjunction of conjunctions) of atoms whose root predicate symbol is in $\mathcal{P}$ such that $T \models \forall \underline{x}\,(L \leftrightarrow \psi)$. The atoms whose root symbol is a predicate in $\mathcal{P}$ are called *T-representative literals.* The set of T-representative literals is denoted by $\mathcal{L}_{\mathcal{P}}$. Notice that T-representative literals are closed under taking substitutions of terms for variables, by definition.

Definition 4.3. *Let T be a theory eliminating quantifiers for a sort S. We say that T has QE-degree N with respect to a set of representative predicates $\mathcal{P}$ iff for every finite family $\{L_i\}_{i \in I}$ of literals from $\mathcal{L}_{\mathcal{P}}$, the formula*

$$\exists x\,(\bigwedge_{i \in I} L_i) \leftrightarrow \bigwedge_{I_0 \subseteq_N I} \exists x\,(\bigwedge_{i \in I_0} L_i) \tag{4}$$

is T-valid (here the variable x is of sort S and $I_0 \subseteq_N I$ means that I_0 is a subset of I having cardinality at most N).

[3] Recall that a model $\mathcal{M}$ is finitely generated iff there is a finite subset X of the support $|\mathcal{M}|$ of $\mathcal{M}$ such that for every $c \in |\mathcal{M}|$, there are $\underline{b} \subseteq X$ and a term $t(\underline{x})$ such that $\mathcal{M} \models t(\underline{b}) = c$.

Notice that the left-to-right implication in (4) is universally valid.

Example 4.4. Real linear arithmetic has the signature $\{0, 1, -, +, =, <, \leq\}$ and the single structure $\mathbb{R}$ (endowed with the natural interpretation) as class of models. If we take the predicates in $\{<, \leq, =\}$ as representatives, an inspection of the formulae produced by the Fourier-Motzkin quantifier elimination procedure shows that this theory has QE-degree equal to 2. A similar result holds for the so-called real and integer 'difference logic.'

We remark that Definition 4.3 does not allow negative literals to be T-representatives. However, there is an obvious way to circumvent this limitation (when needed) by expanding the signature in an inessential way. For instance, if we want negated equations to be T-representative, it is sufficient to expand the signature with a new binary predicate Neq, to let the formula $\forall x \forall y(Neq(x, y) \leftrightarrow x \neq y)$ be T-valid (e.g., by adding it to the axioms of T), and then to include Neq into the set $\mathcal{P}$ of representative predicates. Because of this, we prefer to speak of 'T-representative literals' rather than of 'T-representative atoms'. Finally, notice that, in a multi-sorted context, we might be interested in quantifier elimination over just one sort (e.g., the sort representing time, in timed networks); in this case, it is convenient to take all predicates not involving such sort *and their negations* as representative, by using the above trick.

4.1 Automated termination

Let us fix from now on a wqo theory W and an array-based system $\mathcal{S} = (a, I, \tau)$ (built up over theories T_I, T_E). We make the following extra assumptions:

(E1) T_E has QE-degree N for all sorts occurring in τ as sorts of an existentially quantified data variable (this is the variable e in (1));

(E2) for each disjunct of the form (1) in τ, ϕ_L and the partition components in F are conjunctions of representative literals;

(E3) W has a finite relational signature Σ_W with unique sort S and there is a syntactic interpretation $(-)^*$ from W into A_I^E such that $S^* = \texttt{INDEX}$.

The fact that a translation map $(-)^*$ is a syntactic interpretation from W into A_I^E is decidable since the axioms for W are universal and their translations modulo A_I^E generate $\exists^{A,I}\forall^I$-sentences, whose satisfiability is decidable. As a consequence, if the signature of A_I^E is finite (which is always the case in practical examples), the syntactic interpretations $(-)^*$ can be *enumerated*. In theory, this guarantees the possibility to find the right translation (if it exists) for the termination argument of Theorem 4.5 below to work, relatively to the wqo W, the array-based system $\mathcal{S}$, and the set of unsafe states U under consideration. In practice, however, the search for the right translation could be driven by user provided hints.

We are ready to state the main results of the paper. We use $\alpha(\underline{i}), \beta(\underline{i}), \ldots$ to denote quantifier-free Σ_W-formulae in which at most the variables $\underline{i}$ occur. We say that an $\exists^I$-formula $\exists \underline{i}\ \psi(\underline{i}, a[\underline{i}])$ is a *translation* iff ψ is equivalent (modulo

A_I^E) to a formula α^* for some $\alpha(\underline{i})$. Being a translation is clearly a decidable notion: this is because the signature of W is finite and relational, so the search space for the suitable $\alpha(\underline{i})$ is finite (the required validity tests fall within the decidability result for $\exists^{A,I}\forall^I$-sentences).

Theorem 4.5. *Assume (**E1**)-(**E3**). Suppose that the $\exists^I$-formula $U(a)$ describing unsafe states is a translation and let M be the maximum arity of the predicate symbols in Σ_W. The backward reachability procedure terminates if the following conditions are satisfied by every finite model $\mathcal{M}$ of W ($\sharp\mathcal{M}$ indicates the cardinality of the support of $\mathcal{M}$):*

(i) *if $\sharp\mathcal{M} \leq M$, then the the translation $\delta^*_{\mathcal{M}}$ of the diagram of $\mathcal{M}$ is A_I^E-equivalent to a conjunction of representative literals $L_1^{\mathcal{M}} \wedge \cdots \wedge L_{k_{\mathcal{M}}}^{\mathcal{M}}$;*

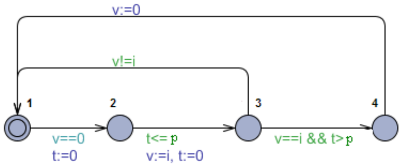
(ii) *if $\sharp\mathcal{M} \leq M$, then for every substructure $\mathcal{M}[\underline{i}_0]$ of $\mathcal{M}$, for every $r = 1, \dots, k_{\mathcal{M}}$, we have that if $L_r^{\mathcal{M}}$ is of the kind $L(\underline{i}_0, a[\underline{i}_0])$ (i.e. if it mentions at most the elements $\underline{i}_0$ of the support of $\mathcal{M}[\underline{i}_0]$), then $A_I^E \models \delta^*_{\mathcal{M}[\underline{i}_0]} \to L_r^{\mathcal{M}}$;*

(iii) *if $\sharp\mathcal{M} \leq M * N$, then the formulae obtained after the elimination of the quantifier $\exists e$ from the formulae $\exists e\, \tau_h[\delta^*_{\mathcal{M}}, f]$ (varying τ_h among the disjuncts of τ and f under the suitable case-marking functions) are all translations.*

In practice, N and M have very small values, typically $N = M = 2$. Thus, only small models must be inspected in order to apply the theorem. When there is no existentially quantified data variable in all transitions (i.e. variable e does not occur neither in the ϕ_L's nor in the F's of (1)), the proof of Theorem 4.5 shows that we can assume $N = 1$ and condition (i) is not needed. In general, condition (i) might not be made fully automated; however, if it holds, an enumerative search can always effectively checks it. In practice, the situation is simple because the following straightforward procedure succeeds and is sufficient to guarantee (i). Consider $\delta^*_{\mathcal{M}}(\underline{i})$: if we replace in it all literals by their Boolean combinations of representative literals and put the result in disjunctive normal form, we get a formula of the form $\theta_1(\underline{i}, a[\underline{i}]) \vee \cdots \vee \theta_k(\underline{i}, a[\underline{i}])$ where the θ_j's are conjunctions of representative literals. Condition (i) of Theorem 4.5 is certainly guaranteed if $k = 1$ or if all θ_j but one are A_I^E-inconsistent. Condition (ii) is technical but usually holds trivially (we can figure that only pathological examples may violate it). The significant condition to verify is just (iii); it can be effectively checked because "being a translation" is decidable.

Theorem 4.5 covers termination of the backward reachability procedure described in Section 3 for several classes of systems including broadcast protocols [13,11], lossy channels systems [5], timed networks with integer clocks [3], and timed networks with a single real clock [6].

The complexity of the procedure of Theorem 4.5 is hard to evaluate, because too many parameters contribute to it (not only M, N, but also the size of τ, of Σ_W, and of the translated atoms, the complexity of quantifier elimination, the number and the complexity of the involved SMT tests, etc.); however, it is arguable that we are well below the lower bounds known for deciding problems which are nevertheless covered by Theorem 4.5 (see e.g. [21]). Thus it might be convenient to apply our termination test before directly running backward search.



Legend: t is the local clock value, v is a shared variable, p is a positive parameter

Fig. 1. Automaton for one process of the Fischer's protocol

4.2 An application of Theorem 4.5: the Fischer protocol

The goal of the Fischer protocol is to ensure mutual exclusion in a network of processes, using a clock and a shared variable v. Each process has a local clock and a control state variable ranging over $\{1, 2, 3, 4\}$. Each process is identified by a natural number > 0 and can read/update a shared variable whose values is either 0 or the index of one of the processes. A process wishing to enter the critical section 4 starts in 1. If $v = 0$, the process goes to 2 and resets its local clock. From 2, the process can go to state 3 if the clock is $< p$ time unit (where p is a positive parameter), sets v to its own index, and again resets its clock. From 3, the process can go to 4 if the clock is $> p$ time unit and v is still equal to the index of the process performing the transition. When exiting 4, the process sets v to 0. The set of unsafe states, i.e. those states violating the mutual exclusion property, can be characterized by the presence of at least two processes entering 4 at the same time. This specification is depicted in Figure 1. Before applying Theorem 4.5, we make few observations to simplify the technical development. As implemented in MCMT, the shared variable v is modelled as a constant array which is updated uniformly so that the invariant $\forall i\,\forall j\,(v[i] = v[j])$ holds. The invariant will be taken into consideration during backward search as explained in Section 3. For simplicity, below, we do not indicate the redundant dependency on the index i and write just v instead of $v[i]$. A second remark concerns *processes identifiers (id's)*, which are integers and the sort of integers is not the same as the sort `INDEX`. Formally, we view the id's as an array $id : \texttt{INDEX} \to \mathbb{Z}$ that is never updated; we implicitly add the conjunct $\forall i\,\forall j\,(id[i] = id[j] \to i = j)$ to the initial formula I and use this too as an invariant (this invariant just says that different processes have different id's); again, for simplicity however, below we write just i for $id[i]$. Since the id's are positive, we also implicitly add the conjunct $\forall i\,(id[i] > 0)$ to the initial formula and use it as an invariant. A third remark is about the use of *additional trivial invariants* that are a substantial part of the specification of the problem and do not capture any deep insight into the system. In our case, for local clocks, stored in the real-valued array t, we use the invariant $\forall i\,(t[i] \geq 0)$ in order to specify that clock values are non-negative.

The theory A_I^E is composed of the theory of pure equality for T_I and the theory T_E is the union of three theories $T_{E_1}, T_{E_2}, T_{E_3}$, where T_{E_1} is linear real arithmetic, T_{E_2} is the theory of the single finite structure $Q = \{1, 2, 3, 4\}$ (the signature of T_{E_2} has just four constants and the identity predicate), and T_{E_3} is linear integer arithmetic. Since the specification contains the positive parameter

p, the theory T_{E_1} is extended with a further constant p constrained by the axiom $p > 0$. Quantifier elimination is assumed only for T_{E_1} and we take as representative predicates all the predicates of T_{E_2}, T_{E_3} together with their negations; we pick just $<, \leq, =$ as representative predicates from T_{E_1} (we get QE-degree 2 with this choice). The following array-based system (a, I, τ) formalizes the Fischer protocol:

- a is a 4-tuple of array variables $\langle l, t, v, id \rangle$, whose target sorts are those of T_{E_1}, T_{E_2} and T_{E_3} (twice), respectively (l is the array of locations, t is the array of clocks, v is the shared register, and id is the array of the id's);
- the formula I is $\forall i \, (l[i] = 1 \wedge t[i] = 0 \wedge v = 0)$, which constrains the initial locations to be equal to 1 and that clocks and the shared variable to 0;
- τ is the disjunction of the time elapsing transition τ_1 and the discrete transitions $\tau_2, ..., \tau_6$ listed below above (identical updates of the array id are not displayed for the sake of conciseness):

$$\tau_1 := \exists c. \big(c > 0 \wedge v' = v \wedge l' = l \wedge t' = \lambda j. (t[j] + c) \big)$$

$$\tau_2 := \exists i. \left(\begin{array}{l} l[i] = 1 \wedge v = 0 \ \wedge v' = v \wedge \\ l' = \lambda j. \ (\mathsf{if} \ j = i \ \mathsf{then} \ 2 \ \mathsf{else} \ l[j]) \ \wedge \\ t' = \lambda j. \ (\mathsf{if} \ j = i \ \mathsf{then} \ 0 \ \mathsf{else} \ t[j]) \end{array} \right)$$

$$\tau_3 := \exists i. \left(\begin{array}{l} l[i] = 2 \wedge t[i] \leq p \ \wedge v' = i \ \wedge \\ l' = \lambda j. \ (\mathsf{if} \ j = i \ \mathsf{then} \ 3 \ \mathsf{else} \ l[j]) \ \wedge \\ t' = \lambda j. \ (\mathsf{if} \ j = i \ \mathsf{then} \ 0 \ \mathsf{else} \ t[j]) \end{array} \right)$$

$$\tau_4 := \exists i. \left(\begin{array}{l} l[i] = 3 \wedge \ v \neq i \ \wedge v' = v \wedge t' = t \ \wedge \\ l' = \lambda j. \ (\mathsf{if} \ j = i \ \mathsf{then} \ 1 \ \mathsf{else} \ l[j]) \end{array} \right)$$

$$\tau_5 := \exists i. \left(\begin{array}{l} l[i] = 3 \wedge \ v = i \ \wedge t[i] > p \ \wedge v' = v \wedge t' = t \ \wedge \\ l' = \lambda j. \ (\mathsf{if} \ j = i \ \mathsf{then} \ 4 \ \mathsf{else} \ l[j]) \end{array} \right)$$

$$\tau_6 := \exists i. \left(\begin{array}{l} l[i] = 4 \ \wedge v' = 0 \ \wedge t' = t \ \wedge \\ l' = \lambda j. \ (\mathsf{if} \ j = i \ \mathsf{then} \ 1 \ \mathsf{else} \ l[j]) \end{array} \right)$$

This completes the array-based specification for the Fischer protocol. Finally, the $\exists^I$-formula U describing the set of unsafe states in which the mutual exclusion property for location 4 is violated is $\exists i_1 \, \exists i_2 \, (i_1 \neq i_2 \wedge l[i_1] = 4 \wedge l[i_2] = 4)$.

To apply Theorem 4.5, we use the theory W in Example 4.2, relatively to the set of predicates indicated below, together with their syntactic translations.

- The unary predicates $Q_1, \ldots, Q_4$ are translated as the formulae $l[i] = 1, \ldots, l[i] = 4$, respectively;
- the unary predicates $P_{=0}, P_{>0}, P_{<p}, P_{=p}, P_{>p}$ are translated as $t[i] = 0, t[i] > 0, t[i] < p, t[i] = p, t[i] > p$, respectively;
- the unary predicate F is translated as $v = i$;
- the 0-ary predicate f is translated as $v = 0$;
- the binary predicate $\leq$ is translated as $t[i_1] \leq t[i_2]$.

Clearly, the translations of the axioms of W (namely reflexivity, transitivity and linearity axioms for $\leq$) are valid modulo A_I^E. The unsafety formula is a translation because $i_1 \neq i_2 \wedge l[i_1] = 4 \wedge l[i_2] = 4$ is the translation of $i_1 \neq i_2 \wedge Q_4(i_1) \wedge Q_4(i_2)$. It remains to check conditions (i)-(ii)-(iii) of Theorem 4.5. For lack of space, we outline how to do this for (iii) by analysing the models of W having at most four elements. Instead of examining them one by one, we use a powerful heuristics. All representative literals coming from the translations of the diagrams of the four-elements models are included in the following list:

$$(L) \qquad \begin{array}{l} k \neq k', l[k] = 1, l[k] = 2, l[k] = 3, l[k] = 4, v = k, v \neq k, v = 0, v \neq 0, \\ t[k] = 0, t[k] > 0, t[k] < p, t[k] = p, t[k] > p, t[k] \leq t[k'], t[k] < t[k'], \end{array}$$

where $k, k' \in \{i_1, i_2, i_3, i_4\}$ - the literal $t[k] < t[k']$ is the translation of $k' \not\leq k$. (Notice that to get the list above, it is sufficient to consider models on a support with at most two elements, because there are no function symbols and at most binary predicates in Σ_W, so any model has a diagram which is the conjunction of the diagrams of all submodels of cardinality at most two.) If we succeed in proving that the formulae $\exists e\, \tau_1[L_1 \wedge L_2, f]$ and $\tau_h[L_1 \wedge L_2, f]$ $(2 \leq h \leq 6)$ are translations for every pair of literals L_1, L_2 coming from the above list (with possibly $L_1 \equiv L_2$), we actually proved more than what is required by condition (iii) (the limitation to at most two literals is due to the fact the QE-degree is 2).

The case of the discrete transitions $\tau_2, ..., \tau_6$ is trivial. It remains to analyze the time elapsing transition τ_1 where $\exists e\, \tau_1[L_1 \wedge L_2, f]$ does not have case-marking function and is $\exists e\, (e > 0 \wedge L_1^{+e} \wedge L_2^{+e})$ where L_i^{+e} is L_i after the substitution of the terms $t[k]$ with $t[k] + e$. The relevant cases to be analyzed are 28 and in all of them we get that $\exists e\, \tau_1[L_1 \wedge L_2, f]$ is a translation. For example, $\exists e\, (e > 0 \wedge t[k_1] + e > 0 \wedge t[k_2] + e = p)$ gives $p > t[k_2] \wedge t[k_1] + p - t[k_2] > 0$ which is equivalent to $p > t[k_2]$ (i.e. to $P_{<p}(k_2)^*$), taking into account the invariant saying that clocks are non-negative.

Thus, all conditions from Theorem 4.5 have been checked and we can predict termination of backward search for Fischer protocol. We emphasize that *the above arguments can be fully mechanized: they consist just in satisfiability checks that can be automatically generated and quickly discharged by suitable tools.*

5 Conclusions

We identified a sufficient condition for the termination of a symbolic backward reachability procedure encompassing many results from the literature in a uniform and declarative framework. We believe that the statement of Theorem 4.5 could be seen as a paradigm for a declaratively-oriented approach to termination; the statement itself needs to be further investigated and exploited in connection to more examples of wqo-theories and syntactic translations arising from encoding termination arguments based on Kruskal theorem.

An interesting direction for future work consists in applying the methods of this paper in connection to abstraction techniques: our results could be profitably employed to predict whether a proposed abstraction of a system yields a terminating search.

References

1. P. A. Abdulla, K. Cerans, B. Jonsson, and Y.-K. Tsay. General decidability theorems for infinite-state systems. In *Proc. of LICS*, pages 313–321, 1996.
2. P. A. Abdulla, G. Delzanno, N. B. Henda, and A. Rezine. Regular model checking without transducers. In *TACAS*, volume 4424 of *LNCS*, pages 721–736, 2007.
3. P. A. Abdulla, J. Deneux, and P. Mahata. Multi-clock timed networks. In *Proc. of LICS'04, the 18th IEEE Int. Symp. on Logic in Computer Science*, 2004.
4. Parosh Aziz Abdulla. Forcing monotonicity in parameterized verification: From multisets to words. In *Proc. of SOFSEM '10*, pages 1–15. Springer-Verlag, 2010.
5. Parosh Aziz Abdulla and Bengt Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91–101, 1996.
6. Parosh Aziz Abdulla and Bengt Jonsson. Model checking of systems with many identical timed processes. *Theoretical Computer Science*, pages 241–264, 2003.
7. F. Alberti, S. Ghilardi, E. Pagani, S. Ranise, and G. P. Rossi. Brief Announcement: Automated Support for the Design and Validation of Fault Tolerant Parameterized Systems—a case study. In *DISC 10*, number 6343 in LNCS, pages 392–394, 2010.
8. A. Carioni, S. Ghilardi, and S. Ranise. Automated Termination in Model Checking Modulo Theories - extended version. Available at `http://homes.dsi.unimi.it/~ghilardi/allegati/CGR_RP11_extended.pdf`.
9. A. Carioni, S. Ghilardi, and S. Ranise. MCMT in the Land of Parametrized Timed Automata. In *Proc. of VERIFY 10*, 2010.
10. Chen-Chung Chang and Jerome H. Keisler. *Model Theory*. North-Holland, Amsterdam-London, third edition, 1990.
11. G. Delzanno, J. Esparza, and A. Podelski. Constraint-based analysis of broadcast protocols. In *Proc. of CSL*, volume 1683 of *LNCS*, pages 50–66, 1999.
12. Herbert B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, New York-London, 1972.
13. J. Esparza, A. Finkel, and R. Mayr. On the verification of broadcast protocols. In *Proc. of LICS*, pages 352–359. IEEE Computer Society, 1999.
14. S. Ghilardi, E. Nicolini, S. Ranise, and D. Zucchelli. Towards SMT Model-Checking of Array-based Systems. In *Proc. of IJCAR*, LNCS, 2008.
15. S. Ghilardi and S. Ranise. A Note on the Stopping Failures Models. 2009. Unpublished Draft, available from MCMT web site.
16. S. Ghilardi and S. Ranise. Goal Directed Invariant Synthesis for Model Checking Modulo Theories. In *(TABLEAUX 09)*, LNAI, pages 173–188. Springer, 2009.
17. S. Ghilardi and S. Ranise. Model Checking Modulo Theory at work: the integration of Yices in MCMT. In *AFM 09 (co-located with CAV09)*, 2009.
18. S. Ghilardi and S. Ranise. Backward reachability of array-based systems by SMT-solving: termination and invariant synthesis. *LMCS*, 6(4), 2010.
19. S. Ghilardi, S. Ranise, and T. Valsecchi. Light-Weight SMT-based Model-Checking. In *Proc. of AVOCS 07-08*, ENTCS, 2008.
20. Silvio Ghilardi and Silvio Ranise. MCMT: A Model Checker Modulo Theories. In *Proc. of IJCAR 2010*, volume 6173 of *LNCS*, pages 22–29. Springer, 2010.
21. Schnoebelen Philippe. Verifying lossy channel systems has nonprimitive recursive complexity. *Information Processing Letters*, 83(5):251–261, 2002.
22. S. Ranise and C. Tinelli. The SMT-LIB Standard: Version 1.2. Technical report, Dep. of Comp. Science, Iowa, 2006. Available at `http://www.SMT-LIB.org/papers`.