Combining Theorem Proving and Narrowing for Rewriting-Logic Specifications

Vlad Rusu

Inria Rennes Bretagne Atlantique, France
Vlad.Rusu@inria.fr

Abstract. We present an approach for verifying dynamic systems specified in rewriting logic, a formal specification language implemented in the Maude system. Our approach is tailored for invariants, i.e., properties that hold on all states reachable from a given class of initial states. The approach consists in encoding invariance properties into inductive properties written in membership equational logic, a sublogic of rewriting logic also implemented in Maude. The invariants can then be verified using an inductive theorem prover available for membership equational logic, possibly in interaction with narrowing-based symbolic analysis tools for rewriting-logic specifications also available in the Maude environment. We show that it is possible, and useful, to automatically test invariants by symbolic analysis before interactively proving them.

1 Introduction

Rewriting logic [1], abbreviated as RL in this paper, is a formal specification language, in which a system's dynamics can be expressed by means of *rewrite rules* over a system's *state* defined in some version of equational logic. The adequacy of rewriting logic for specifying dynamic systems has been demonstrated by many practical applications, including programming language semantics [2], security protocols [3], and bioinformatics [4]. There are several systems which implement different variants of this logic, including Maude [5], Elan [6], and CAFEOBJ [7]. *Membership equational logic* [8], hereafter called MEL in this paper, is the logic implemented in Maude as RL's underlying equational logic.

The Maude system [5] consists of a language for expressing RL and MEL specifications, and a set of tools for analysing such specifications and verifying them against user-defined properties. The automatic tools provided by the Maude system include an enumerative, finite state-space searching tool and an enumerative, finite-state model checker for linear temporal logic properties [9] and also for an extension of it called linear temporal logic of rewriting [10]. A symbolic state exploration tool based on narrowing techniques has also recently been made available in Maude [11], and narrowing-based symbolic model checking for linear temporal logic has also been studied [12]. Infinite-state rewriting logic specifications can be verified in Maude with respect to temporal-logic properties, using *equations* to reduce infinite-state spaces to finite-state ones [13].

G. Fraser and A. Gargantini (Eds.): TAP 2010, LNCS 6143, pp. 135–150, 2010.

Our contribution is an approach for verifying invariants of infinite-state RL specifications. Intuitively, an invariant is a predicate that holds in all states that are reachable from a given class of initial states. Our approach consists in encoding the verification of *invariance properties* on the *reachable model* of RL theories, into the verification of MEL properties on the *initial model* of MEL theories. As a consequence, invariance properties can be proved using inductive theorem provers for MEL, such as the ITP tool [14]. Since our formalisation is consistent with that underlying Maude's narrowing-based symbolic analysis tools, our theorem-proving approach can be used in interaction with them.

Specifically, we demonstrate in this paper the usefulness of symbolic simulation in helping the interactive proofs of invariants. Such proofs are performed by induction in the theorem prover, and when the induction step fails, the user must provide the theorem prover with state predicates that (1) are invariants and (2) imply the induction step. While proving (2) is typically automatic - it amounts to proving an implication, the proof of (1) typically has to be performed, again, by induction. Then, assume the user poses a "wrong" state predicate, for which (1) is not provable. Unaware of her error, she will try to prove the invariance of her predicate, also by induction in the theorem prover. The failing induction will lead her to attempt to pose yet other additional auxiliary invariants... in a proof effort that cannot succeed. By contrast, symbolic simulation can automatically falsify invariants by symbolically exploring the system's reachable states up to a given depth, thereby preventing the user from entering dead ends in her proofs.

The rest of the paper is organised as follows. In Section 2 we provide background on MEL and on RL. In Section 3 we recall results about narrowing in the context of RL: the soundness and (under some conditions) the completeness of narrowing for solving reachability problems for rewriting-logic specifications [3]. A class of RL systems satisfying those conditions is identified, and we argue that the class is expressive enough to express many communication protocols. Hence, for systems in that class, narrowing can find all their reachable states starting from a possibly infinite set of initial states, up to a bounded depth; this property is important for our goals - testing invariants before trying to prove them.

In Section 4 we define the notion of a MEL *invariant φ of a* RL *specification $\mathcal{R}$ starting from an initial (possibly, infinite) set of states denoted by a (possibly, non-ground) term t_0*, as follows: φ is provable in the initial model of the underlying MEL theory E of $\mathcal{R}$, for all states reachable in $\mathcal{R}$ from initial states.

The core of the approach is presented in Section 5. First, we define an automatic translation that takes a RL theory $\mathcal{R}$ and a (possibly, non-ground) term t_0, and generates a MEL theory $\mathcal{M}(\mathcal{R}, t_0)$, which enriches the MEL subtheory of $\mathcal{R}$ with a new sort, called *Reachable*, and with the membership axioms that inductively define this sort. We then show (†) if $\mathcal{R}$ is topmost, then, for ground terms t, and up to equality *modulo* the equations E of $\mathcal{R}$, the statements "being of sort *Reachable* in $\mathcal{M}(\mathcal{R}, t_0)$" and "being reachable in the reachability model of $\mathcal{R}$ from ground instances of t_0" are equivalent. Next, we use the $\mathcal{M}(\mathcal{R}, t_0)$ theory to give an alternative definition of an *invariant φ of a* RL *theory $\mathcal{R}$ starting from a possibly non-ground term t_0*, as follows: $\varphi(t)$ is provable in the initial model

of the MEL subtheory of $\mathcal{R}$, for all ground terms t that have sort *Reachable* in $\mathcal{M}(\mathcal{R}, t_0)$ (again, up to equality *modulo* E). That the two given definitions of invariance are equivalent follows from (†). The advantage of the second definition is that it allows us to prove invariants of RL specifications by induction on the sort *Reachable*, using existing inductive theorem provers for MEL such as Maude's ITP. Section 6 illustrates the theorem-proving of invariants integrated with narrowing-based symbolic falsification of invariants on a Bakery mutual-exclusion algorithm. Section 7 discusses related and future work, and concludes.

2 Membership Equational Logic and Rewriting Logic

We briefly present membership equational logic and rewriting logic [1,8,11].

A *membership equational logic* (MEL) *signature* is a tuple (K, Σ, S) where K is a set of *kinds*, Σ is a $K^* \times K$ indexed family of function symbols $\Sigma = \{\Sigma_{w,k}\}_{(w,k)\in K^*\times K}$, and $S = \{S_k\}_{k\in K}$ is a pairwise disjoint K-indexed family of sets of *sorts* - where S_k is the set of sorts of kind k. A signature (K, Σ, S) is often denoted simply by Σ; then, T_Σ denotes the set of ground terms over signature Σ. Given a set $X = \{x_1 : k_1, \ldots, x_n : k_n\}$ of *kinded variables*, $T_\Sigma(X)$ denotes the set of terms with free variables in the set X. Similarly, $T_{\Sigma,k}$ and $T_{\Sigma,k}(X)$ denote, respectively, the set of ground terms of kind k and the set of terms of kind k with free variables in the set X. A MEL *atomic formula* over (K, Σ, S) is either an equality $t = t'$, where t and t' are terms in $T_{\Sigma,k}(X)$, for some kind $k \in K$, or a *membership assertion* $t : s$, where t is a term in $T_{\Sigma,k}(X)$ and s is a sort in S_k, for some kind $k \in K$. A MEL *sentence* is a Horn clause

$$(\forall X) t = t' \text{ if } C, \text{ or} \tag{1}$$

$$(\forall X) t : s \text{ if } C \tag{2}$$

where the *condition* C has the form $\bigwedge_{i\in I}(u_i = v_i) \wedge \bigwedge_{j\in J}(w_j : s_j)$, for some finite sets of indices I, J. Sentences of the form (1) are called *conditional equations*, and sentences of the form (2) are called *conditional memberships*. A sentence is *unconditional* when it does not have a condition.

A MEL *theory* is a tuple $\mathcal{M} = (\Sigma, E)$ that consists of a MEL signature Σ and a set of MEL sentences over Σ. MEL has a complete deduction system [8], in the sense that a formula φ is provable from the sentences of a theory (Σ, E), denoted as $(\Sigma, E) \vdash \varphi$ (or simply $E \vdash \varphi$), if and only φ is semantically valid, i.e., it holds in *all* the *models* of that theory. The standard model of a specification is called its *initial model* [8]. In the initial model, sorts are sets of equivalence classes of ground terms *modulo* the equations E, where two ground terms t, t' are in the same equivalence class, denoted by $t =_E t'$, iff $E \vdash (\forall \emptyset) t = t'$. We write $E \vdash_{ind} (\forall X)\varphi$ to say that the sentence φ holds in the initial model of (Σ, E).

Example 1. The specification *NAT* in Figure 1 defines natural numbers with addition. In the initial model of *NAT*, the natural number $n \geq 1$ is represented by the "sum" $1 + \cdots + 1$ of length n, and the term 0 represents the number 0.

$K_{NAT} = \{Nat?\}$ with $S_{Nat?} = \{Nat\}$.
$\Sigma_{NAT(\lambda,Nat?)} = \{0, 1\}$.
$\Sigma_{NAT(Nat?Nat?,Nat?)} = \{+\}$.
$\Sigma_{NAT\,w,Nat?} = \emptyset$, otherwise

$$E_{NAT} = \begin{cases} 0 : Nat, \\ 1 : Nat, \\ (\forall N)\ 0 + N = N \\ (\forall N, M)\ N + M = M + N \\ (\forall N, M, P)\ (N + M) + P = N + (M + P) \end{cases}$$

Fig. 1. A specification of natural numbers

Note the three equations for associativity, communitativity, and unity (ACU). These axioms can either be declared explicitly, or they can be attached to the "+" operator as so-called *equational attributes*. For our purposes we use the former solution with theorem proving, and the latter one with narrowing. The main reason is that Maude's ITP theorem prover that we use does not handle ACU operators. In contrast, *unification* (an essential part of the *narrowing-based symbolic analysis*, which we also use) is *finitary and complete* for ACU operators, provided they are not defined by other equations (or provided the remaining equations have certain technical properties) [15]. The finiteness and completeness of ACU-unification are important for the completeness of narrowing as a symbolic simulation technique for a class of rewriting-logic specifications expressive enough to encode typical communication protocols. We shall come back to this issue at the end of Section 3 after we present rewriting logic and narrowing.

A *rewriting logic* (RL) theory is a tuple $\mathcal{R} = (K, \Sigma, S, E, R)$, - often abbreviated as (Σ, E, R) - where (K, Σ, S, E) is a MEL theory and R is a set of *rewrite rules* (ρ) $(\forall X)\ l \to r$ *if* C where $l, r \in T_{\Sigma,k}(X)$ for some kind k that depends on the rule, and the condition C has the form $\bigwedge_{i\in I}(u_i = v_i) \wedge \bigwedge_{j\in J}(w_j : s_j)$ for some finite sets of indices I and J; that is, like for MEL sentences, we consider that only equations and memberships are allowed in the conditions of the rules. Note that in its most general form [16] rewriting logic also allows for *rewrites in conditions* and *frozen arguments*, which we do not consider here. Moreover, we shall only consider *topmost* theories [3]: a theory is *topmost* if there exists a certain kind $k \in K$ such that (i) for all rewrite rules of the above form, $l, r \in T_{\Sigma,k}(X)$, and (ii) no operation in Σ takes arguments of the kind k. Many authors have shown the adequacy of RL for specifying dynamic systems (including the restricted topmost theories [3] - "almost" all distributed systems can be so described). The idea is to specify the system's state *kind* by equations and membership axioms, and the system's dynamics by (topmost) rewrite rules over the kind of states.

Example 2. Mutual exclusion of two processes to a resource can be ensured by the so-called Bakery algorithm. The processes can be in modes *Sleep* (not interested in obtaining the resource), *Try* (when they are trying to obtain the resource) and *Critical* (when they have the resource). To control transitions between these modes, each process has a *ticket*, which is a natural number. The main idea is that a process gets the resource when it has the smallest ticket (because that process has been trying to get the resource for the longest time), or, alternatively, when the other process is not interested in getting the resource.

To specify the system as a RL theory $\mathcal{BAK}$ we use the specification NAT of natural numbers with addition in Figure 1, as well as a (trivial) specification of the kind *Mode?* with the only sort *Mode*, and the three constants S, T, and C of the sort *Mode*. The states of the system are built using a constructor $\langle \ldots \rangle$ that takes two modes and two natural numbers and returns a term in the sort *State*. The evolution of the system is described by the rewrite rules in Figure 2.

Here, l_1 and l_2 are variables of the sort *Mode* and t_1, t_2, x are variables of the sort *Nat*. We describe the evolution of the first process (the left-hand side column); the evolution of the second process, in the right-hand side column, is similar.

$$\begin{array}{ll}
\langle S, l_2, t_1, t_2\rangle \rightarrow \langle T, l_2, t_2+1, t_2\rangle & \langle l_1, S, t_1, t_2\rangle \rightarrow \langle l_1, T, t_1, t_1+1\rangle \\
\langle T, l_2, t_1, 0\rangle \rightarrow \langle C, l_2, t_1, 0\rangle & \langle l_1, T, 0, t_2\rangle \rightarrow \langle l_1, C, 0, t_2\rangle \\
\langle T, l_2, t_1, t_1+x+1\rangle \rightarrow \langle C, l_2, t_1, t_1+x+1\rangle & \langle l_1, T, t_2+x+1, t_2\rangle \rightarrow \langle l_1, C, t_2+x+1, t_2\rangle \\
\langle C, l_2, t_1, t_2\rangle \rightarrow \langle S, l_2, 0, t_2\rangle & \langle l_1, C, t_1, t_2\rangle \rightarrow \langle l_1, S, t_1, 0\rangle
\end{array}$$

Fig. 2. $\mathcal{BAK}$: Bakery algorithm. We only use *unconditional* rewrite rules, since conditional narrowing is outside the scope of Maude's current implementation of narrowing.

The first rule moves the process from the *Sleep* to the *Try* mode, and changes the value of its ticket t_1, by "assigning" to it the term $t_2 + 1$. Then, the first process may move to the *Critical* mode if (a) the other process has its ticket equal to 0, or (b) the first process has the smallest ticket. The latter condition is obtained by having the ticket of the second process denoted by the term t_1+x+1, for some x of the sort Nat, where t_1 is the ticket of the first process. Finally, the first process goes back to the *Sleep* mode and sets its ticket back to zero.

Assume that we want to simulate the behaviours of the $\mathcal{BAK}$ starting from a possibly infinite class of initial states. Let the initial states be denoted by the term $\langle S, S, t, t\rangle$, in which both processes are in the *Sleep* modes and have the same initial value t for their tickets, where t is a variable of the sort *Nat* - the actual initial value of the tickets is left unspecified. The desired simulation cannot be performed using Maude's enumerative state-exploration tools, because those tools require a unique initial state, i.e., a ground term. By contrast, narrowing can simulate the executions of our system, starting from a non-ground term.

Reachability. The notation $\mathcal{R} \vdash (\forall X) t_0 \rightarrow t$ expresses the fact that the term $t \in T_\Sigma(X)$ is *provable* from the term $t_0 \in T_\Sigma(X)$ in the deduction system of $\mathcal{R}$ (which amounts to applying the rewrite rules of $\mathcal{R}$ *modulo* the equations E of $\mathcal{R}$). The *reachability model* [16] of $\mathcal{R}$ is a *transition system* whose states are equivalence classes of ground terms *modulo* E. For all states $[t]_E, [t']_E$, there is a transition $[t]_E \rightarrow_{\mathcal{R}} [t']_E$ in this model iff there exists a proof $\mathcal{R} \vdash (\forall \emptyset) t \rightarrow t'$ using exactly one rewrite rule of $\mathcal{R}$. We denote by $[t]_E \rightarrow^*_{\mathcal{R}} [t']_E$ the fact that the state $[t']_E$ is reachable from the state $[t]_E$ in the reachability model of $\mathcal{R}$.

3 Narrowing, and a Class of Systems It Can Analyse

In the context of rewriting logic, narrowing is used for symbolically solving *reachability problems* [3,11] (and more generally, for symbolically model checking linear temporal-logic properties [12]). We consider reachability problems of the form *"given terms $t_0 \in T_\Sigma(X)$ and $t \in T_\Sigma$ does there exist a (ground) substitution $\sigma : X \to T_\Sigma$ such that $[t_0\sigma]_E \to^*_{\mathcal{R}} [t]_E$ holds"* for topmost theories $\mathcal{R}$, whose rewrite rules are *unconditional* and do not have supplementary variables in their right-hand sides wth respect to their left-hand sides - to simplify matters and to be consistent with the current implementation of narrowing in Maude.

Given a substitution $\sigma : X \mapsto T_\Sigma(X)$, we write $t_1 \overset{\sigma}{\leadsto}_{\mathcal{R}} t_2$ if there exists in $\mathcal{R}$ a rule $(\rho)\quad (\forall X)\ l \to r$ such that the variables occuring in t_1 and l are disjoint and such that $E \vdash t_1\sigma = l\sigma$ and $E \vdash r\sigma = t_2$. That is, it is provable in the MEL subtheory (Σ, E) of $\mathcal{R}$ that σ is a *unifier* for t_1 and the left-hand side l of the rule (ρ) and that σ *matches* the right-hand side r of the rule with the term t_2.

For $t_1, t_1', t_2, t_2' \in T_\Sigma(X)$ we write $t_1' \overset{\sigma}{\leadsto}_{\mathcal{R},E} t_2'$ if $t_1' =_E t_1 \overset{\sigma}{\leadsto}_{\mathcal{R}} t_2 =_E t_2'$. The narrowing relation $\leadsto_{\mathcal{R},E} \subseteq T_\Sigma(X) \times T_\Sigma(X)$ is defined by $t_1 \leadsto_{\mathcal{R},E} t_2$ iff $t_1 \overset{\sigma}{\leadsto}_{\mathcal{R},E} t_2$ for some substitution σ. Let $\leadsto^*_{\mathcal{R},E}$ be the reflexive transitive closure of $\leadsto_{\mathcal{R},E}$. The *soundness of narrowing for solving reachability problems* in Maude [11] says essentially that for all terms $t_0 \in T_\Sigma(X)$ and $t \in T_\Sigma$, if $t_0 \leadsto^*_{\mathcal{R},E} t$ then there exists a ground substitution $\sigma : X \mapsto T_\Sigma$ such that $[t_0\sigma]_E \to^*_{\mathcal{R}} [t]_E$.

On the other hand, *completeness* of narrowing, meaning that narrowing $t_0 \leadsto^*_{\mathcal{R},E} t$ "finds" in some sense all solutions σ such that $[t_0\sigma]_E \to^*_{\mathcal{R}} [t]_E$, does not hold in general, but only under some technical conditions [3]. The most important condition is that unification modulo E be *finitary and complete*; that is, for any equation of the form $t_1 =_E t_2$, for $t_1, t_2 \in T_\Sigma(X)$, the algorithm returns a finite set of substitutions, which are all the solutions of the equation[1].

And this is precisely the case when E consists of ACU axioms for some operations in Σ, such as those defined for the "+" operation in our specification of natural numbers (Figure 1), or the encoding of sets in MEL based on an ACU "union" operation. These observations are important because they suggest a class of systems that can be effectively symbolically simulated by narrowing, in the sense that narrowing eventually "reaches" all reachable states. The Bakery algorithm in Figure 2 is one such system, thanks to the ACU-based definition of natural numbers given in Figure 1. More generally, finite control and possibly unbounded counters, typically encountered in such protocols, can be encoded using our ACU-based encoding of natural numbers. Even more generally, *communication protocols* with unordered channels also fall in this class - by encoding unordered channels using sets constructed with an ACU definition of union.

One limitation remaining is that such communication protocols often require conditions on the counters: such as, e.g., the conditions on the *tickets* in the

[1] The other conditions are (in addition to those posed at the beginning of this section) that the theory (Σ, E) is in the *order-sorted* fragment of MEL and that that the equations be *regular and sort-preserving*: left-hand and right-hand sides have the same variables, and left-hand side does not have a greater sort than right-hand side.

Bakery algorithm. However, we can encode such *affine* conditions - comparisons of linear-arithmetic terms with constants - by unconditional rules, as follows. Assume a system endoded in RL having one conditional topmost rule of the form $\langle l \rangle \rightarrow \langle r \rangle$ *if* $\Sigma_{i=1}^{n} a_i x_i > b$ (and possibly other rules). We enrich the sort *State* with two integer components. The initial states will now have the form $\langle I, \Sigma_{i=1}^{n} a_i x_i, b \rangle$, where I is the expression denoting initial the states of the original system. Each rule except the one we are encoding, of the form $\langle l' \rangle \Rightarrow \langle r' \rangle$ *if* C', becomes $\langle l', x, y \rangle \Rightarrow \langle r', x, y \rangle$ *if* C' - that is, all the rules except the one we are encoding leaves the "new" components of the *State* unchanged. Our rule of interest becomes $\langle l, x, x+n+1 \rangle \Rightarrow \langle r, x, x+n+1 \rangle$, where the effect of checking the condition to "see" if the rule can be applied on a term is achieved by *unifying* the left-hand side of our rule: $\langle l, x, x+n+1 \rangle$ with that term. By generalising this encoding to several linear-arithmetic conditions and to several rules, we obtain an unconditional system equivalent to a conditional one.

Thus, narrowing can effectively simulate a class of systems expressive enough to encode communication protocols with finite control, counters, and channels.

4 Invariants of Rewrite Theories

We continue by discussing in this section the notion of *invariant* for a rewrite theory. In general, a predicate over the states of a dynamic system is an invariant if the predicate holds in all the states of the system that are reachable from a given class of initial states. To formalize this notion in the case of a system specified in a (topmost) RL theory $\mathcal{R}$, we have to answer the following questions:

1. how are the *states* and the *dynamics* to be specified?
2. how are the state *predicates* to be formalized?
3. when are the predicates to be considered as *holding* in a state?

For item (1) we adopt the usual RL representations: states are equivalence classes (*modulo* the equations E of $\mathcal{R}$) of ground terms of a certain kind $[State]$. There is a possibly infinite set of initial states denoted by a term t_0, possibly with variables, of the kind $[State]$; then, initial states are equivalence classes of ground instances of t_0. Regarding the dynamics of the system, it shall naturally be defined by reachability in the reachability model of $\mathcal{R}$.

With regards to item (2): state predicates shall be formalised by Horn sentences of the form $(\forall x : [State])(\forall Y)\varphi$ having a free variable x of the kind $[State]$ (and possibly other free variables in the set Y, with $x \notin Y$). Finally, for item (3), a state predicate φ shall be considered to hold in a state t when the predicate $\varphi(t/x)$, obtained from φ by substituting the variable x with the term t, holds in the initial model of the MEL subtheory of the RL theory: $E \vdash_{ind} (\forall Y)\varphi(t/x)$.

In summary, when a system is specified as a topmost RL theory $\mathcal{R}$, we formalize the intuitive notion of an invariant as a state predicate φ holding in all states are reachable from an initial state - denoted by $\langle \mathcal{R}, t_0 \rangle \vdash_{ind} \Box\varphi$ - as follows.

Definition 1. $\langle \mathcal{R}, t_0 \rangle \vdash_{ind} \Box\varphi$ *if for all ground terms* $t \in T_\Sigma$*, and all ground substitutions* $\sigma : X \mapsto T_\Sigma$*,* $[t_0\sigma]_E \rightarrow_{\mathcal{R}}^{*} [t]_E$ *implies* $E \vdash_{ind} (\forall Y)\varphi(t/x)$.

Example 3. Consider the RL specification $\mathcal{BAK}$ from Example 2. We have seen that the states of the system are quadruples consisting of two modes and two natural numbers built using the constructor $\langle \dots \rangle$ of the sort *State*. The mutual exclusion between readers and writers is encoded as the state predicate *mutex*:

$(\forall\, x : [State])(\forall\, t_1, t_2 : Nat).\ mutex(\langle C, C, t_1, t_2\rangle) = false$
$(\forall\, x : [State])(\forall\, t_1, t_2 : Nat, l_1, l_2 : Mode).\ l_1 \neq C \Rightarrow mutex(\langle l_1, l_2, t_1, t_2\rangle) = true$
$(\forall\, x : [State])(\forall\, t_1, t_2 : Nat, l_1, l_2 : Mode).\ l_2 \neq C \Rightarrow mutex(\langle l_1, l_2, t_1, t_2\rangle) = true$

The invariance of *mutex* on the $\mathcal{BAK}$ system starting from all states denoted by the term $\langle S, S, t, t\rangle$ with the variable $t : Nat$, is written $\langle \mathcal{BAK}, \langle S, S, t, t\rangle\rangle \vdash_{ind} \Box mutex$ and defined by: for all ground terms t of kind $[State]$, and for each ground term $n : Nat$, $\langle S, S, n, n\rangle \rightarrow^*_{\mathcal{BAK}} [t]_{E_{\mathcal{BAK}}}$ implies $NAT \vdash_{ind} mutex(t/x)$.

Before we proceed to verifying invariants by theorem proving, we show how invariants can be disproved using Maude's narrowing-based symbolic analysis.

Disproving invariants. Disproving an invariance statement $\langle \mathcal{R}, t_0\rangle \vdash_{ind} \Box\varphi$ amounts to finding a ground substitution σ and a sequence $[t_0\sigma]_E \rightarrow^*_{\mathcal{R}} [t]_E$ such that $E \not\models_{ind} \varphi(t/x)$. If we find a narrowing sequence $t_0 \rightsquigarrow^*_{\mathcal{R},E} t$ such that $E \not\models_{ind} \varphi(t/x)$, then, by soundness of narrowing there exists a sequence $[t_0\sigma]_E \rightarrow^*_{\mathcal{R}} [t]_E$ such that $E \not\models_{ind} \varphi(t/x)$, hence, $\langle \mathcal{R}, t_0\rangle \vdash_{ind} \Box\varphi$ is disproved.

The completeness of narrowing says that all such "disproofs" will eventually be found by narrowing. Concretely, such sequences $t_0 \rightsquigarrow^*_{\mathcal{R},E} t$ can be found by Maude's `search` command. Consider a variant $\mathcal{BAK}'$ of our running example, in which the term 1 is replaced everywhere by the term 0. The invariance statement $\langle \mathcal{BAK}', \langle S, S, t, t\rangle\rangle \vdash_{ind} \Box mutex$ can be disproved (falsified) by Maude's following command: `search`$\langle S, S, t, t\rangle \rightsquigarrow^* \langle C, C, t_1, t_2\rangle$, which immediately finds the term $\langle C, C, 0, 0\rangle$ violating the *mutex* predicate. A more involved use of the `search` command to check *auxiliary* invariants that are needed in an inductive proof of the "main" mutual-exclusion invariant of the $\mathcal{BAK}$ system is shown in Section 6.

5 Theorem Proving for Invariance Properties

The previous section discussed the falsification of invariants. In this section we propose an approach for *proving* invariants. The structure of the section is as follows. We first define an automatic translation that takes a topmost RL theory $\mathcal{R}$ and a term t_0, possibly with variables, and generates a MEL theory $\mathcal{M}(\mathcal{R}, t_0)$, which enriches $\mathcal{R}$ with a sort called *Reachable* and with memberships defining this sort. We show that for all ground terms t, the state $[t]_E$ is reachable in the initial model of $\mathcal{R}$ from an initial state $[t_0\sigma]$ if and only if the term t has the sort *Reachable* in the initial model of $\mathcal{M}(\mathcal{R}, t_0\sigma)$. Then, we prove that, for any state predicate $(\forall x : [State])(\forall Y)\varphi$, the statements $\langle \mathcal{R}, t_0\rangle \vdash_{ind} \Box\varphi$ and $\mathcal{M}(\mathcal{R}, t_0) \vdash_{ind} (\forall x : [State])(\forall Y)(x : Reachable \Rightarrow \varphi)$ are equivalent. This equivalence is the basis for proving invariants using inductive theorem provers, such as the ITP tool.

In the following definition, we "encode" reachability in a RL theory $\mathcal{R}$ (starting from a possibly non-ground term t_0) in a MEL theory $\mathcal{M}(\mathcal{R}, t_0)$ using a membership axiom for t_0 and a membership axiom $\mu(\rho)$ for each rule ρ in $\mathcal{R}$.

Definition 2. *Consider a* RL *theory* $\mathcal{R} = (K, \Sigma, S, E, R)$*, a sort* $State \in S$*, and a term* $t_0 \in T_{\Sigma,[State]}(X)$*. We denote by* $\mathcal{M}(\mathcal{R}, t_0)$ *the following* MEL *theory* $(K_{\mathcal{M}(\mathcal{R},t_0)}, \Sigma_{\mathcal{M}(\mathcal{R},t_0)}, S_{\mathcal{M}(\mathcal{R},t_0)}, E_{\mathcal{M}(\mathcal{R},t_0)})$ *constructed as follows:*

- $K_{\mathcal{M}(\mathcal{R},t_0)} = K$
- $\Sigma_{\mathcal{M}(\mathcal{R},t_0)} = \Sigma$
- $S_{\mathcal{M}(\mathcal{R},t_0)} = S \cup S'_{[State]}$ *with* $S'_{[State]} = S_{[State]} \cup \{Reachable\}$ *and* $Reachable \notin S$
- $E_{\mathcal{M}(\mathcal{R},t_0)} = E \cup \{(\forall X) t_0 : Reachable\} \cup \{\mu(\rho) | \rho \in R\}$*, with* $\mu((\forall X)\ l \rightarrow r\ if\ C))$ *being the membership* $(\forall X)\ r : Reachable\ if\ l : Reachable \wedge C$. □

Example 4. Consider the Bakery system given in Example 2. The MEL theory $\mathcal{M}(\mathcal{BAK}, \langle S, S, t, t\rangle)$ consists of the following elements:

- $K_{\mathcal{M}(\mathcal{BAK},\langle S,S,t,t\rangle)} = K_{\mathcal{BAK}}$.
- $\Sigma_{\mathcal{M}(\mathcal{BAK},\langle S,S,t,t\rangle)} = \Sigma_{\mathcal{BAK}}$.
- $S_{\mathcal{M}(\mathcal{BAK},\langle S,S,t,t\rangle)} = S_{\mathcal{BAK}} \cup S'_{[State]}$, with $S'_{[State]} = S_{\mathcal{BAK}_{[State]}} \cup \{Reachable\}$
- $E_{\mathcal{M}(\mathcal{BAK},\langle S,S,t,t\rangle)} = E_{\mathcal{BAK}} \cup E'$, where E' is the set of memberships axioms shown in Figure 3.

$\langle S, S, t, t\rangle : Reachable$

$\langle T, l_2, t_2 + 1, t_2\rangle : Reachable\ if\ \langle S, l_2, t_1, t_2\rangle : Reachable$
$\langle C, l_2, t_1, 0\rangle : Reachable\ if\ \langle T, l_2, t_1, 0\rangle : Reachable$
$\langle C, l_2, t_1, t_1 + x + 1\rangle : Reachable\ if\ \langle T, l_2, t_1, t_1 + x + 1\rangle : Reachable$
$\langle S, l_2, 0, t_2\rangle : Reachable\ if\ \langle C, l_2, t_1, t_2\rangle : Reachable$

$\langle l_1, T, t_1, t_1 + 1\rangle : Reachable\ if\ \langle l_1, S, t_1, t_2\rangle : Reachable$
$\langle l_1, C, 0, t_2\rangle : Reachable\ if\ \langle l_1, T, 0, t_2\rangle : Reachable$
$\langle l_1, C, t_2 + x + 1, t_2\rangle : Reachable\ if\ \langle l_1, T, t_2 + x + 1, t_2\rangle : Reachable$
$\langle l_1, S, t_1, 0\rangle : Reachable\ if\ \langle l_1, C, t_1, t_2\rangle : Reachable$

Fig. 3. Membership axioms for $\mathcal{M}(\mathcal{BAK}, \langle S, S, t, t\rangle)$

Lemma 1. *Consider a* RL *theory* $\mathcal{R} = (K, \Sigma, S, E, R)$*, with* $State \in S$ *and* $t \in T_{\Sigma,[State]}$*. For all* $t' \in T_{\Sigma,[State]}$*,* $[t]_E \rightarrow^*_{\mathcal{R}} [t']_E$ *iff* $\mathcal{M}(\mathcal{R}, t) \vdash t' : Reachable$.

Proof. The idea of the proof is that each transition in the reachability model of $\mathcal{R}$, generated by using a rule (ρ) of $\mathcal{R}$, can be emulated by an deduction in the proof system of $\mathcal{M}(\mathcal{R}, t)$, using the membership $\mu(\rho)$ given in Definition 2.

($\Rightarrow$) By induction on the length of the sequence $[t]_E \rightarrow^*_{\mathcal{R}} [t']_E$.

If the length is 0 then $E \vdash (\forall \emptyset) t = t'$. The membership $t : Reachable$ in $\mathcal{M}(\mathcal{R}, t)$ implies $\mathcal{M}(\mathcal{R}, t) \vdash t : Reachable$, hence, $\mathcal{M}(\mathcal{R}, t) \vdash t' : Reachable$.

Assume the statement holds for sequence of length n. Any sequence $[t]_E \rightarrow^{n+1}_{\mathcal{R}} [t']_E$ of length $n + 1$ can be decomposed into $[t]_E \rightarrow^n_{\mathcal{R}} [t'']_E \rightarrow_{\mathcal{R}} [t']_E$, such that the last step uses a rule (ρ) $(\forall X) l \rightarrow r\ if\ C$ with a ground substitution σ.

1. then, $t'' \equiv t''\sigma =_E l\sigma$ because t'' is ground (and $\equiv$ denotes syntactical equality), $t' =_E r\sigma$, and $E \vdash C\sigma$. The latter implies *a fortiori* $\mathcal{M}(\mathcal{R}, t) \vdash C\sigma$;

2. by induction, $\mathcal{M}(\mathcal{R},t) \vdash t'' : \mathit{Reachable}$, hence, $\mathcal{M}(\mathcal{R},t) \vdash l\sigma : \mathit{Reachable}$;
3. hence, using the membership $(\mu(\rho))\ r : \mathit{Reachable}\ \mathit{if}\ \ l : \mathit{Reachable} \wedge C$ from Definition 2 with σ, $\mathcal{M}(\mathcal{R},t) \vdash r\sigma : \mathit{Reachable}$, i.e., $\mathcal{M}(\mathcal{R},t) \vdash t' : \mathit{Reachable}$.

($\Leftarrow$) By induction on the length of the proof $\mathcal{M}(\mathcal{R},t) \vdash t' : \mathit{Reachable}$, where by length we here mean the number of applications of memberships of the form $(\mu(\rho))\ r : \mathit{Reachable}\ \mathit{if}\ \ l : \mathit{Reachable} \wedge C$ generated from rules $(\rho)\ (\forall X) l \to r\ \mathit{if}\ \ C$.

If the length is 0 then $t' : \mathit{Reachable}$ has been proved using the membership $t : \mathit{Reachable}$ for the initial term, hence, $t =_E t'$ and $[t]_E \to_{\mathcal{R}}^* [t']_E$ follows.

Assume the statement holds for sequence of length n. Any proof $\mathcal{M}(\mathcal{R},t) \vdash t' : \mathit{Reachable}$ of length $n+1$ can be decomposed into a proof $\mathcal{M}(\mathcal{R},t) \vdash t'' : \mathit{Reachable}$ of length n, for some $t'' \in T_\Sigma$, followed by an application of a membership $(\mu(\rho))\ r : \mathit{Reachable}\ \mathit{if}\ \ l : \mathit{Reachable} \wedge C$ with a ground substitution σ such that $t' =_E r\sigma$, $t'' =_E l\sigma$, and $\mathcal{M}(\mathcal{R},t) \vdash C\sigma$. Since the sort *Reachable* is "new" in $\mathcal{M}(\mathcal{R},t)$, it does not occur in the condition C, hence, $E \vdash C\sigma$. Since t'' is ground, $t'' \equiv t''\sigma =_E l\sigma$. Hence, the rule $(\rho)\ (\forall X) l \to r\ \mathit{if}\ \ C$ can be applied on t'' and generates the transition $[t'']_E \to_{\mathcal{R}} [t']_E$. By induction hypothesis, $[t]_E \to_{\mathcal{R}}^* [t'']_E$. The transitivity of the $\to_{\mathcal{R}}^*$ relation concludes. □

We have proved the equivalence between $[t]_E \to_{\mathcal{R}}^* [t']_E$ and $\mathcal{M}(\mathcal{R},t) \vdash t' : \mathit{Reachable}$ for ground terms t, t'. In particular, in our setting where the terms t are ground instances of the (possibly, non-ground) term t_0, we obtain the equivalence between *for all ground substitutions* σ, $[t_0\sigma] \to_{\mathcal{R}}^* [t]_E$ and *for all ground substitutions* σ, $\mathcal{M}(\mathcal{R},t_0\sigma) \vdash t : \mathit{Reachable}$. However, in order to reason by induction on *Reachable*, we need a different hypothesis, namely, $\mathcal{M}(\mathcal{R},t_0) \vdash_{ind} t : \mathit{Reachable}$. The following lemma bridges the gap between those statements.

Lemma 2. *For $t_0 \in T_\Sigma(X)$ and $t \in T_\Sigma$, $\mathcal{M}(\mathcal{R},t_0) \vdash_{ind} t : \mathit{Reachable}$ if and only if $\mathcal{M}(\mathcal{R},t_0\sigma) \vdash t : \mathit{Reachable}$ for all ground substitutions $\sigma : X \mapsto T_\Sigma$.*

Proof. Let us denote by $\mathcal{M}(\mathcal{R})$ the MEL theory obtained by removing the membership $(\forall X) t_0 : \mathit{Reachable}$ from the theory $\mathcal{M}(\mathcal{R},t_0)$ in Definition 2. Then, $\mathcal{M}(\mathcal{R},t_0) \vdash_{ind} t : \mathit{Reachable}$ iff $\mathcal{M}(\mathcal{R}) \vdash_{ind} ((\forall X) t_0 : \mathit{Reachable} \Rightarrow t : \mathit{Reachable})$. Since truth in the initial model for a statement is equivalent to deduction of all ground instances of that statement, we obtain that the last entailment is equivalent to $\mathcal{M}(\mathcal{R}) \vdash (t_0\sigma : \mathit{Reachable} \Rightarrow t : \mathit{Reachable})$ for all ground substitutions σ, itself equivalent to $\mathcal{M}(\mathcal{R},t_0\sigma) \vdash t : \mathit{Reachable}$ for all ground substitutions σ. □

Theorem 1. *Consider a* RL *theory $\mathcal{R} = (K, \Sigma, S, E, R)$, with $\mathit{State} \in S$, a term $t_0 \in T_{\Sigma,[\mathit{State}]}(X)$, and a state predicate $(\forall x : [\mathit{State}], \forall Y)\varphi$. Then $\langle \mathcal{R}, t_0 \rangle \vdash_{ind} \Box\varphi$ if and only if $\mathcal{M}(\mathcal{R},t_0) \vdash_{ind} (\forall x : [\mathit{State}])(\forall Y)(x : \mathit{Reachable} \Rightarrow \varphi)$.*

Proof. Since $x \notin Y$, $\mathcal{M}(\mathcal{R},t_0) \vdash_{ind} (\forall x : [\mathit{State}])(\forall Y)(x : \mathit{Reachable} \Rightarrow \varphi)$ is equivalent to $\mathcal{M}(\mathcal{R},t_0) \vdash_{ind} (\forall x : [\mathit{State}])(x : \mathit{Reachable} \Rightarrow (\forall Y)\varphi)$, and using the fact that truth in the initial model for a statement is equivalent to deduction of ground instances of that statement, we obtain equivalently $\forall t \in T_{\Sigma,[\mathit{State}]}.\, \mathcal{M}(\mathcal{R},t_0) \vdash_{ind} (t : \mathit{Reachable} \Rightarrow (\forall Y)\varphi(t/x))$. Then, using the

equivalence $A \vdash (B \Rightarrow C)$ iff $(A \vdash B$ implies $A \vdash C)$ we obtain equivalently

$$\forall t \in T_{\Sigma,[State]}.\ \ ([\mathcal{M}(\mathcal{R},t_0) \vdash_{ind} t : \mathit{Reachable}] \text{ implies } [\mathcal{M}(\mathcal{R},t_0) \vdash_{ind} (\forall Y)\varphi(t/x)]) \tag{3}$$

By using Lemmas 1 and 2, the left-hand side of the above implication is equivalent to *for all ground substitutions* σ, $[t_0\sigma]_E \rightarrow^*_{\mathcal{R}} [t]_E$. Then, the implication (3) is equivalent to (‡) *for all* $t \in T_{\Sigma,[State]}$ *and all ground substitutions* σ, $[t_0\sigma]_E \rightarrow^*_{\mathcal{R}} [t]_E$ *implies* $\mathcal{M}(\mathcal{R},t_0) \vdash_{ind} (\forall Y)\varphi(t/x)$. Finally, we note that truth in the initial model of $\mathcal{M}(\mathcal{R},t_0)$ and truth in the initial model of E are equivalent, for all statements that do not refer to the "new" sort *Reachable*, because, for, the truth of such statements, the memberships defining *Reachable* in $\mathcal{M}(\mathcal{R},t_0)$ are irrelevant. And φ does not refer to this sort, precisely because it is "new". Hence, the last statement (‡) in our chain of equivalences is itself equivalent to *for all* $t \in T_{\Sigma,[State]}$ *and all ground substitutions* σ, $[t_0\sigma]_E \rightarrow^*_{\mathcal{R}} [t]_E$ *implies* $E \vdash_{ind} (\forall Y)\varphi(t/x)$, which by Definition 1 is $\langle \mathcal{R},t_0\rangle \vdash_{ind} \Box\varphi$. □

6 Testing Invariants before Proving Them

The results in the previous section show that proving invariants is equivalent to proving inductive theorems in the initial model of a MEL theory, thus, invariants can be proved by induction. Consider the specification $\mathcal{BAK}$ and its *mutex* predicate. To prove the statement $\langle \mathcal{BAK}, \langle S,S,t,t\rangle\rangle \vdash_{ind} \Box \mathit{mutex}$, we prove $(\forall x)(x : \mathit{Reachable} \Longrightarrow \mathit{mutex}(x) = \mathit{true})$ in the initial model of $\mathcal{M}(\mathcal{BAK}, \langle S,S,t,t\rangle)$ using the ITP tool. We describe that proof in some detail, and show that narrowing-based symbolic simulation is really useful in preventing the user from taking a wrong direction in the proof.

The proof goes by induction on the sort *Reachable*. This generates nine subgoals: one for the membership defining the initial state, and eight for the eight other memberships defining the sort *Reachable* (all memberships shown in Fig. 3).

The subgoal for the initial states is automatically proved by the ITP. Out of the eight remaining subgoals, four are also automatically proved by the ITP. Those are the subgoals corresponding to the memberships whose left-hand sides are states where at least one process is *not* in the *Critical* mode. These are, for the first process: $\langle T,l_2,t_2+1,t_2\rangle : \mathit{Reachable}$ *if* $\langle S,l_2,t_1,t_2\rangle : \mathit{Reachable}$ and $\langle S,l_2,0,t_2\rangle : \mathit{Reachable}$ *if* $\langle C,l_2,t_1,t_2\rangle : \mathit{Reachable}$, and the symmetrical ones for the second process. In the left-hand sides of these memberships, the *mutex* predicate obviously holds, and the ITP tool "realises" this.

The remaining subgoals cannot be automatically proved by the ITP, because it needs additional information that only the user can provide.

Corresponding to the following membership of the first process:

$$\langle C,l_2,t_1,t_1+x+1\rangle : \mathit{Reachable}\ \mathit{if}\ \langle T,l_2,t_1,t_1+x+1\rangle : \mathit{Reachable}$$

the ITP presents us with essentially the following subgoal, written as a *sequent*:

$$\frac{\langle T, l_2, t_1, t_1 + x + 1\rangle{:}\mathit{Reachable} \quad \mathit{mutex}(\langle T, l_2, t_1, t_1 + x + 1\rangle)}{\mathit{mutex}(\langle C, l_2, t_1, t_1 + x + 1\rangle)}$$

That is, using the hypotheses "above" the line, one has to prove the conclusion "below" the line. In order to simplify her proof, the user performs a case splitting on the variable l_2, which generates three sub-subgoals for the given subgoal. Two of them are automatically proved by the ITP, because their conlusions are $\mathit{mutex}(C, T, t_1, t_1 + x + 1\rangle)$ and $\mathit{mutex}(C, S, t_1, t_1 + x + 1\rangle)$, which obviously hold. However, the third sub-subgoal is not proved by the ITP: it has the form

$$\frac{\langle T, C, t_1, t_1 + x + 1\rangle{:}\mathit{Reachable} \quad \mathit{mutex}(\langle T, C, t_1, t_1 + x + 1\rangle)}{\mathit{mutex}(\langle C, C, t_1, t_1 + x + 1\rangle)} \tag{4}$$

By examining the hypotheses in the subgoal (4), the user realises that the second one is trivially *true*, hence, it is useless; and that the conclusion is trivially *false*. The only remaining possibility for proving the subgoal is therefore to prove that the first hypothesis: $\langle T, C, t_1, t_1 + x + 1\rangle : \mathit{Reachable}$ does not hold. After some thinking, the user realises that indeed, states of the form $\langle T, C, t_1, t_1 + x + 1\rangle$ should not be reachable, because the very basic principle of the Bakery algorithm is that the process that is in the critical section should have the *smallest ticket*; but that is precisely *not* the case in the states of the above form.

Happy with her reasoning, the user poses the following lemma to the ITP:

$$\frac{\langle l_1, l_2, t_1, t_1 + x\rangle : \mathit{Reachable}}{l_2 \neq C} \tag{5}$$

She postpones proving (5), and confidently uses it to sucessfully prove the subgoal (4). Eventually, she completes the proof of the main invariant *mutex*, and returns to proving (5). However, no matter how hard she tries, she does not succeed... of course, because the lemma is not true! Indeed, had the user tried to falsify (5) using Maude's narrowing-based `search` command, she would have realised her error: `search` $\langle S, S, t, t\rangle \rightsquigarrow^* \langle l_1, C, x, x + y\rangle$ immediately finds the solution $x = 0, l_1 = S, y = 1+w$ for some $w : Nat$, which contradicts Lemma (5).

Fixing the error in the lemma amounts to adding the hypothesis $t_1 > 0$. The fixed lemma is indeed provable, but now, the new lemma does not solve by itself the subgoal (4), for which it was posed in the first place! To deal with this problem, the whole proof has to be re-thought, and a possible solution is to prove that states of the form $\langle T, C, t_1, t_1 + x + 1\rangle$ such that $t_1 > 0$ are not reachable. The proof eventually succeeds, but with more effort than if the error in the lemma had been found using Maude's narrowing-based `search` command.

7 Conclusion, Related Work, and Future Work

State-space exploration and model checking, both enumerative and symbolic, abstraction for reducing infinite-state systems to finite ones, and interactive theorem proving for infinite-state systems are well-known verification techniques.

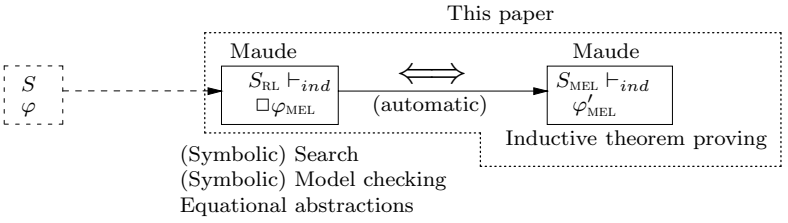

Fig. 4. Our approach in the context of Maude's verification tools

For rewriting-logic specifications, all but the last one are currently supported in the Maude environment. Our contribution adds a part currently missing. The approach is based on an automatic translation of invariance properties of a significant fragment of rewriting logic (topmost, without rules in conditions or frozen arguments) into inductive properties of membership equational logic. The proposed approach can then be used in conjunction with those other tools (Figure 4) thanks to the "semantical consistency" between the definition of invariance in reachable models of RL theories and the definition of narrowing. We illustrate on a simple Bakery algorithm the combination of theorem proving with narrowing for "testing" lemmas before proving them. The results are encouraging. We expect the benefits to be even more substantial for more complex systems and proofs. This statement has to be assessed by experiments. We have identified a class of systems that can be effectively symbolically simulated by narrowing, and can encode communication protocols; these are our natural future case studies.

Related Work. Unification and narrowing are features introduced in the latest version of Maude [11]. A tool built around Maude - the Maude NRL Analyser [17] has been used for verifying security protocols, a topic also present in [3]. We are users of (the Maude implementation of) narrowing in combination with our theorem-proving approach, and plan to use them on communication protocols.

Regarding theorem proving, our work is inspired by Bruni and Meseguer, who proposed in [16] a different encoding of RL into MEL. Their translation handles RL in its full generality, and their goal is to define the semantics and proof theory of RL in terms of those of MEL. By contrast, our encoding only captures a subset of RL, which has been shown in [3] to be expressive enough for specifying many classes of systems. But, in addition to [16] we also encode invariance properties for the given subset of RL as inductive properties in MEL, which provides us with an effective way of verifying invariants by theorem proving, possibly in interaction with Maude's other symbolic analysis tools. Moreover, our encoding is much simpler than that proposed in [16][2]. A simple encoding is essential in theorem-proving, for users to "recognise" the properties they are trying to prove.

[2] We encode reachability using only one additional *sort*, without any new operations. By contrast, [16] requires to double the number of *kinds* in the RL specification, and for each kind, there are 4 new sorts, and 4 operations defined using 7 equations each.

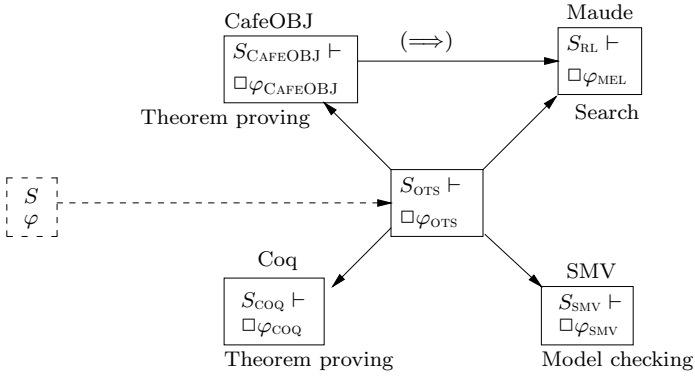

Fig. 5. Observational Transition Systems: representation in CafeOBJ and other tools

The present paper improves our own earlier, French version of this work [18]. The notions of dynamics and invariance that we use in the present paper are standard for systems specified in RL. Hence, our theorem-proving approach for invariants, and narrowing-based symbolic simulation for invariant falsification "talk about" the same *notion* of invariant, i.e., we have a semantical consistency. By contrast, the notions of dynamics and invariance in [18] are ad-hoc: we defined there a "ground top-level rewriting" and defined the dynamics of systems specified in RL and the notion of invariance based on that notion. Hence, in [18] we used enumerative state-space exploration for invariant falsification, without certainty that the falsification procedure deals with the *same* notion of invariant as the theorem-proving approach. Moreover, enumerative exploration can only deal with systems with finitely many initial states (expressed using finitely many ground terms). By contrast, symbolic analysis can deal with systems with possibly infinitely many initial states (expressed using a non-ground term). This is also the case of our theorem-proving approach. On the other hand, we verify in [18] a more involved, n-processes version of the Bakery Algorithm, where nontrivial auxiliary invariants are required for proving the mutual-exclusion goal.

There is a huge body of work dedicated to proving and disproving invariants, and it is impossible to cite all references. We limit ourselves to the approach probably closest to ours, proposed by the CafeOBJ group from Japan's Advanced Institute of Science and Technology. Their approach consists in encoding the system under verification as an Observational Transition System (OTS), and invariants as state predicates over the states of OTSs. The OTS can be represented into several formalisms (Figure 5): CafeOBJ and Coq, for theorem proving [19,20]; Maude, for invariant falsification using enumerative techniques [21]; and SMV, for model checking [22]. Closest to our work is the theorem-proving approach in CafeOBJ. The fundamental difference between our approach and theirs lies in the fact that we remain within one single, integrated environment and formalism (that of Maude and of rewriting logic/membership equational logic), which allows us to rely on a common semantics for the various verification activities (Figure 4). By contrast, the CafeOBJ group use several tools, with different formalisms and different

underlying semantics (Figure 5). This naturally raises the question of semantical consistency. On the other hand, by not "bothering" with semantical consistency, the CafeOBJ approach can be more efficient than ours, because they can use highly-specialised tools, which are typically more efficient than Maude's (symbolic) model checker and theorem prover that we are using.

In the future we are planning to explore the integration of our theorem proving approach with Maude's symbolic model checker for temporal logic [12]. The model checker builds and analyses a symbolic graph encoding the reachable states of the system. An invariant established by theorem proving may help the symbolic model checker, by showing that certain nodes of a symbolic graph are unreachable and can be safely removed from it; thereby enabling the model checker to prove certain temporal-logic properties that could not be proved before. For example, consider a version of the Bakery algorithm containing the additional rule $\langle C, C, t_1, t_2\rangle \Rightarrow \langle C, C, t_1, t_2\rangle$, which says that if the protocol enters the critical section, it stays there forever in the same state. Assume that the symbolic graph "has" a symbolic state of the form $\langle C, C, t_1 + 1, t_2 + 1\rangle$. Then, on this graph, the temporal-logic property $\Box((t_1 > 0 \wedge t_2 > 0) \Rightarrow \Diamond(t_1 = 0 \vee t_2 = 0))$, which says that, from all reachable states state where both tickets are nonzero, a state where at least one ticket is 0 will eventually be reached, is not provable. The reason is the self-loop on $\langle C, C, t_1 + 1, t_2 + 1\rangle$. By proving mutual exclusion we can safely remove that state (and the loop responsible for the model checker's failure) from the graph, thereby possibly enabling the model checker to succeed.

References

1. Martí-Oliet, N., Meseguer., J.: Rewriting logic: roadmap and bibliography. TCS 285(2), 121–154 (2002)
2. Meseguer, J., Rosu, G.: The rewriting logic semantics project. TCS 373(3), 213–237 (2007)
3. Meseguer, J., Thati., P.: Symbolic reachability analysis using narrowing and its application to the verification of cryptographic protocols. Higher-Order and Symbolic Computation 20(1-2), 123–160 (2007)
4. Eker, S., Knapp, M., Laderoute, K., Lincoln, P., Meseguer, J., Sönmez, M.K.: Pathway logic: Symbolic analysis of biological signaling. In: Pacific Symposium on Biocomputing, pp. 400–412 (2002)
5. Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., Talcott, C. L. (eds.): All About Maude - A High-Performance Logical Framework. LNCS, vol. 4350. Springer, Heidelberg (2007)
6. Borovanský, P., Kirchner, C., Kirchner, H., Moreau, P.E., Ringeissen, C.: An overview of ELAN. Electr. Notes Theor. Comput. Sci. 15 (1998)
7. Diaconescu, R., Futatsugi, K.: Logical foundations of CafeOBJ. TCS 285(2), 289–318 (2002)
8. Meseguer, J.: Membership algebra as a logical framework for equational specification. In: Parisi-Presicce, F. (ed.) WADT 1997. LNCS, vol. 1376, pp. 18–61. Springer, Heidelberg (1998)
9. Eker, S., Meseguer, J., Sridharanarayanan, A.: The Maude LTL model checker. Electr. Notes Theor. Comput. Sci. 71 (2002)

10. Meseguer, J.: The temporal logic of rewriting: A gentle introduction. In: Degano, P., De Nicola, R., Meseguer, J. (eds.) Concurrency, Graphs and Models. LNCS, vol. 5065, pp. 354–382. Springer, Heidelberg (2008)
11. Clavel, M., Durán, F., Eker, S., Escobar, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., Talcott, C. L.: Unification and narrowing in Maude 2.4. In: Treinen, R. (ed.) RTA 2009. LNCS, vol. 5595, pp. 380–390. Springer, Heidelberg (2009)
12. Escobar, S., Meseguer, J.: Symbolic model checking of infinite-state systems using narrowing. In: Baader, F. (ed.) RTA 2007. LNCS, vol. 4533, pp. 153–168. Springer, Heidelberg (2007)
13. Meseguer, J., Palomino, M., Martí-Oliet, N.: Equational abstractions. In: Baader, F. (ed.) CADE 2003. LNCS (LNAI), vol. 2741, pp. 2–16. Springer, Heidelberg (2003)
14. Clavel, M., Palomino, M., Riesco, A.: Introducing the ITP tool: a tutorial. J. Universal Computer Science 12(11), 1618–1650 (2006)
15. Escobar, S., Meseguer, J., Sasse, R.: Variant narrowing and equational unification. Electr. Notes Theor. Comput. Sci. 238(3), 103–119 (2009)
16. Bruni, R., Meseguer., J.: Semantic foundations for generalized rewrite theories. TCS 360(1-3), 386–414 (2006)
17. Escobar, S., Meadows, C., Meseguer, J.: A rewriting-based inference system for the NRL protocol analyzer and its meta-logical properties. Theor. Comput. Sci. 367(1-2), 162–202 (2006)
18. Rusu, V., Clavel, M.: Vérification d'invariants pour des systèmes spécifiés en logique de réécriture. In: JFLA. Studia Informatica Universalis, vol. 7.2, pp. 317–350 (2009), `http://www.irisa.fr/vertecs/Equipe/Rusu/rc09.pdf`
19. Futatsugi, K.: Verifying specifications with proof scores in CafeOBJ. In: ASE, pp. 3–10. IEEE Comp. Soc., Los Alamitos (2006)
20. Ogata, K., Futatsugi, K.: State machines as inductive types. IEICE Transactions 90-A(12), 2985–2988 (2007)
21. Kong, W., Seino, T., Futatsugi, K., Ogata, K.: A lightweight integration of theorem proving and model checking for system verification. In: APSEC, pp. 59–66. IEEE Comp. Soc., Los Alamitos (2005)
22. Ogata, K., Nakano, M., Nakamura, M., Futatsugi, K.: Chocolat/SMV: a translator from CafeOBJ to SMV. In: PDCAT, pp. 416–420. IEEE Comp. Soc., Los Alamitos (2005)