Proving Safety Properties of Rewrite Theories

Technical Report
November 2010

Camilo Rocha and José Meseguer

{hrochan2,meseguer}@cs.illinois.edu
Formal Methods and Declarative Languages Laboratory
Department of Computer Science
University of Illinois at Urbana-Champaign
201 N Goodwin Ave
Urbana, IL 61801

Rewriting logic theories are a general and expressive way of specifying concurrent systems, where states are axiomatized by equations and transitions among states are axiomatized by rewrite rules. In this paper, we present a *transformational* and *reductionistic* deductive approach for verifying *safety properties* of rewrite theories. In our approach all formal temporal reasoning about concurrent transitions is ultimately reduced to purely equational inductive reasoning. Narrowing modulo axioms is extensively used to simplify the equational proof obligations to which all proofs of safety formulas are ultimately reached. This allows these generic verification methods to take advantage of the existing wealth of equational reasoning techniques and tools already available. We report on the implementation of this deductive system in the Maude Invariant Analyzer tool, which provides a substantial degree of automation and can automatically discharge many proof obligations without user intervention.

By now we all know that programming
is as hard or as easy as proving a theorem ...
We have to let the symbols do the work, for that is
the only known technique that scales up.

E.W. Dijkstra, "The next fifty years"
EWD1243.

Table of Contents

Proving Safety Properties of Rewrite Theories 1
Camilo Rocha and José Meseguer
1 Introduction 4
2 Preliminaries 5
3 Ground Stability 8
4 Ground Invariance 15
5 Strengthenings for Ground Invariance 17
6 Maude's Invariant Analyzer 22
7 Related Work and Concluding Remarks 26

List of Figures

1 Checking $\mathcal{R} = (\Sigma, E, R)$ ground p-stable (with $\Theta_{(l,r,C)}$ in rule NR1 defined as in Theorem 1). 10
2 Checking $\mathcal{R} = (\Sigma, E, R)$ ground p-invariant from I (with E^q_Π in rule C⇒ as defined in Theorem 3). 16
3 $\mathcal{R} = (\Sigma, E, R)$ and $p, q, I \in \Pi$. Sets of states closed under $\rightarrow_{\mathcal{R}}$ in $\mathcal{T}_{\mathcal{R}}$ are depicted with rectangles. $\mathcal{R}$ is ground q-stable and ground q-invariant for I; $\mathcal{R}$ is ground p-invariant for I but not necessarily ground p-stable. (a) $[\![q]\!] \subseteq [\![p]\!]$. (b) $[\![q]\!] \not\subseteq [\![p]\!]$. 20
4 Checking $\mathcal{R} = (\Sigma, E, R)$ ground p-invariant under strengthenings (with $\Theta_{(l,r,C)}$ in rule NR2 as defined in Theorem 5). 21
5 Approach for checking ground invariance and ground stability of rewrite theories. 23

1 Introduction

Safety properties of concurrent systems are among the most important properties to verify. They have received extensive attention in many different formal approaches, both algorithmic and deductive. Algorithmic approaches such as model checking are quite attractive because they are automatic. However, they cannot always be applied as a system can be infinite-state, so that no model checking algorithm which assumes a finite-state system can be used. Even if an abstraction can be found to make the system finite-state, an additional difficulty may arise: although for each initial state the set of states reachable from it is finite, the set of initial states may still be *infinite*, so that model checking verification may not be possible. For example, a mutual exclusion protocol should be verified for an arbitrary number of clients in its initial state, even if we have managed to abstract its states so that the set of states reachable from each initial state is always finite.

This paper is part of a broader effort to develop *generic* methods to reason about safety properties of concurrent systems and more generally about any property specifiable in temporal logic. It advances such an effort by developing generic *deductive* methods and tools for proving two key safety properties, namely, stability and invariance, plus their combination by means strengthening techniques. By "generic" we mean that the verification methods and their associated tools are not tied to a specific programming language. By contrast, the UNITY logic is an elegant temporal logic inference system tailored for the verification of concurrent programs in the UNITY language [2], so that nontrivial changes would be required to apply such a logic to, say, threaded Java programs. Similarly, the deductive methods for verifying safety properties developed by Manna and Pnueli in [11] are tailored to verify concurrent programs in the specific imperative language described in [11].

The advantage of generic verification methods and tools is that the costly tool development effort can be amortized across a much wider range of applications, whereas a language-specific verification tool can only be applied to systems programmed in that specific language. Of course, any such generic approach requires a *logical framework* general enough to encompass many different models and languages. In our case we use the rewriting logic framework [12], which has been shown to express very naturally many different models of concurrent computation and many concurrent languages. The generic framework and its tools can then be easily specialized to specific languages. This is exactly the approach taken in the rewriting logic semantics project [15], where the semantics of a wide variety of concurrent programming languages is defined in rewriting logic, and then Maude [4] and its LTL model checker can be used to verify programs in any of those languages.

The goal of this paper is to extend rewriting logic based generic verification methods to support the *deductive* verification of concurrent systems, beginning with safety properties. In the rewriting logic framework, a concurrent system, such as, for example, a network protocol or an entire concurrent programming language such as Java, is specified as a *rewrite theory* $\mathcal{R} = (\Sigma, E, R)$, with (Σ, E)

an equational theory specifying the system's *states* as elements of the initial algebra $\mathcal{T}_{\Sigma/E}$, and R a collection of (non-equational) rewrite rules specifying the system's *concurrent transitions*. For example, the `BAKERY` module in Example 1 of Section 3 specifies Lamport's bakery protocol with three rewrite rules.

The generic approach we present to safety property verification is both *transformational* and *reductionistic*. Safety properties are a special type of *inductive* properties. That is, they do not hold for just any model of the given rewrite theory $\mathcal{R}$, but for its *initial reachability model* $\mathcal{T}_{\mathcal{R}}$ [1]. Concretely, for $\mathcal{R} = (\Sigma, E, R)$, this means that the states of such an initial model are precisely elements of the initial algebra $\mathcal{T}_{\Sigma/E}$, and that its one-step transitions are *provable* rewrite steps between such states by means of the rules R. Therefore, given any safety property φ we are interested in the model-theoretic satisfaction relation $\mathcal{T}_{\mathcal{R}} \models \varphi$, which we approximate deductively by means of an inductive inference relation $\mathcal{R} \Vdash \varphi$ which we prove is *sound*, that is, $\mathcal{R} \Vdash \varphi$ always implies $\mathcal{T}_{\mathcal{R}} \models \varphi$.

The approach we present is *transformational*, in the sense that the rules of inference transform pairs of the form $\mathcal{R} \Vdash \varphi$ into other such pairs $\mathcal{R}' \Vdash \varphi'$. It is also *reductionistic* in the sense that: (i) all temporal logic formulas eventually disappear and are replaced by *purely equational formulas*; and (ii) the rewrite theory $\mathcal{R} = (\Sigma, E, R)$ is eventually replaced by its underlying *equational theory* (Σ, E). That is, in the end *all formal reasoning about safety properties is reduced to inductive equational reasoning about equational properties* in (Σ, E). This allows these generic safety verification methods to take advantage of the existing wealth of equational reasoning techniques and tools already available. In particular, the Invariant Analyzer tool (InvA) supporting the transformational inference system we present, takes full advantage of Maude's equational logic inductive theorem prover (ITP).

We can summarize our main contributions as follows:

- Proof of the inductive soundness of a transformational inference system to prove stability and invariance properties about the initial reachability model $\mathcal{T}_{\mathcal{R}}$ of a topmost rewrite theory $\mathcal{R}$, as well as the soundness of additional inference rules supporting the *strengthening* of invariants.
- Systematic use of *narrowing modulo axioms* with the equations defining state predicates, specialized in this paper to ground stability and invariance analysis, to greatly simplify the equational proof obligations to which all proofs of safety formulas are ultimately reduced.
- Implementation of the above inference system in the InvA Invariant Analyzer, which provides a substantial degree of automation and can automatically discharge many proof obligations without user intervention.

2 Preliminaries

We follow notation and terminology from [13] for order-sorted equational logic and from [1] for rewriting logic. An *order sorted signature* $\Sigma = (S, \leq, F)$ is assumed to have a finite poset of sorts $(S, \leq)$, and a finite set of function symbols $F = \{F_{w,s}\}_{(w,s) \in S^* \times S}$. Each connected component in the poset ordering

has a top sort; for any $s \in S$, $[s]$ denotes the top sort in its component. Furthermore, for each operator declaration $f \in F_{s_1 \ldots s_n, s}$ there is also a declaration $f \in F_{[s_1] \ldots [s_n], [s]}$.

An S-sorted family $X = \{X_s\}_{s \in S}$ of disjoint sets of variables with each X_s countably infinite is assumed. $T_\Sigma(X)_s$ is the set of terms of sort s, $T_{\Sigma,s}$ is the set of ground terms of sort s, and $\mathcal{T}_\Sigma(X)$ and $\mathcal{T}_\Sigma$ denote the corresponding term algebras; throughout this paper we assume that $T_{\Sigma,s} \neq \emptyset$ for each sort s in Σ. The set of variables of a term t is written $\mathrm{vars}(t)$ and it is extended to sets of terms in the natural way.

A *substitution* θ is a sorted mapping from a finite subset $\mathrm{dom}(\theta) \subseteq X$ to $T_\Sigma(X)$; $\mathrm{ran}(\theta)$ denotes the set of variables introduced by θ, i.e., $\mathrm{ran}(\theta) = \{\mathrm{vars}(\theta(x)) \mid x \in \mathrm{dom}(\theta)\}$. A *ground substitution* is a substitution θ such that $\mathrm{ran}(\theta) = \emptyset$. Substitutions are homomorphically extended to $\mathcal{T}_\Sigma(X)$ in the natural way. The application of a substitution θ to a term t is denoted by $t\theta$ and the composition of two substitutions θ_1 and θ_2 is denoted by $\theta_1\theta_2$.

Given an order-sorted signature Σ, an (order-sorted) equational atom over Σ is a Σ-*equality* $t = u$, with $t, u \in T_\Sigma(X)_s$ for some sort $s \in S$, and an (order-sorted) Σ-*equation* (or Σ-*equational sentence*) is a Horn clause $(\forall X)\ t = u$ **if** C, where $t = u$ is a Σ-equality and the *condition* C is a finite conjunction of Σ-equalities $\bigwedge_{i \in I} t_i = u_i$ such that $\mathrm{vars}(u) \subseteq \mathrm{vars}(t)$ and $\mathrm{vars}(\{t_i, u_i\}) \subseteq \mathrm{vars}(t)$ for each $i \in I$. An equation with empty condition is called *unconditional*; otherwise it is called *conditional*. An (order-sorted) *equational theory* is a tuple $\mathcal{E} = (\Sigma, E)$ consisting of an order-sorted signature Σ and a finite set of Σ-equations E.

An equational theory $\mathcal{E} = (\Sigma, E)$ *entails* a sentence $(\forall X)\varphi$, written $\mathcal{E} \vdash (\forall X)\varphi$, with φ a conditional equation of the form $t = u$ **if** C, if and only if $(\forall X)\varphi$ can be obtained from $\mathcal{E}$ by finite application of the deduction rules in [13], if and only if $(\forall X)\varphi$ is valid in all models of $\mathcal{E}$. An equational theory (Σ, E) induces the congruence relation $=_E$ on $T_\Sigma(X)$ defined for any $t, u \in T_\Sigma(X)$ by $t =_E u$ if and only if $\mathcal{E} \vdash (\forall Y)t = u$, with $Y = \mathrm{vars}(t) \cup \mathrm{vars}(u)$. $\mathcal{T}_{\Sigma/E}(X)$ and $\mathcal{T}_{\Sigma/E}$ denote the quotient algebras induced by $=_E$ over the algebras $\mathcal{T}_\Sigma(X)$ and $\mathcal{T}_\Sigma$, respectively. $\mathcal{T}_{\Sigma/E}$ is the *initial algebra* of (Σ, E). We write $(\Sigma, E) \Vdash (\forall X)\varphi$ to denote that sentence $(\forall X)\varphi$ is valid in $\mathcal{T}_{\Sigma/E}$, i.e., $\mathcal{T}_{\Sigma/E} \models (\forall X)\varphi$, which is equivalent to $(\Sigma, E) \vdash \varphi\theta$ for each ground substitution $\theta : \mathrm{vars}(\varphi) \longrightarrow T_\Sigma$, i.e., $(\forall X)\varphi$ is an inductive consequence of (Σ, E).

An E-*unifier* for a Σ-equality $t = u$ is a substitution θ such that $t\theta =_E u\theta$. A *complete* set of E-unifiers for a Σ-equality $t = u$, written $\mathrm{CSU}_E(t = u)$, is a set of E-unifiers for $t = u$ such that for any E-unifier α for $t = u$ there exists a $\theta \in \mathrm{CSU}_E(t = u)$ and substitution β such that $\alpha =_E \theta\beta$, where if $\mathrm{dom}(\theta_1) = Y = \mathrm{dom}(\theta_2)$, then $\theta_1 =_E \theta_2$ denotes that $\theta_1(x) =_E \theta_2(x)$ for each $x \in Y$. $\mathrm{CSU}_E(t = u)$ is called *finitary* if it contains a finite number of E-unifiers. A *ground* E-*unifier* for a Σ-equality $t = u$ is a ground substitution $\gamma : X \longrightarrow T_\Sigma$ such that $t\gamma =_E u\gamma$. We let $\mathrm{GU}_E(t = u)$ denote the set of ground E-unifiers for the Σ-equality $t = u$.

Given an order-sorted signature Σ, a rewrite atom over Σ is a Σ-*sequent* $t \to u$, with $t, u \in T_\Sigma(X)_s$ for some sort $s \in S$, and an (order-sorted) Σ-*rewrite*

rule (or *Σ-rewrite sentence*) is a Horn clause $(\forall X)\ t \to u$ **if** $C \wedge D$, where $t \to u$ is a Σ-sequent and the *condition* is a finite conjunction of Σ-equations $C = \bigwedge_{i\in I} t_i = u_i$ and Σ-sequents $D = \bigwedge_{j\in J} t_j \to u_j$ such that $\mathrm{vars}(u) \subseteq \mathrm{vars}(t)$, $\mathrm{vars}(\{t_i, u_i\}) \subseteq \mathrm{vars}(t)$ for each $i \in I$, and $\mathrm{vars}(\{t_j, u_j\}) \subseteq \mathrm{vars}(t)$ for each $j \in J$. A rewrite rule is said to have an *equational condition* when its condition does not contain any sequents. It is called *unconditional* when its condition is empty; otherwise, it is called *conditional*. An (order-sorted) *rewrite theory* is a tuple $\mathcal{R} = (\Sigma, E, R)$ consisting of an order-sorted equational theory $\mathcal{E}_\mathcal{R} = (\Sigma, E)$ and a finite set of Σ-rewrite rules R. A *topmost rewrite theory* is a rewrite theory $\mathcal{R} = (\Sigma, E, R)$ such that each $(\forall X)\ l \to r$ **if** $C \in R$ has $l, r \in T_\Sigma(X)_\mathfrak{s}$ for some top sort $\mathfrak{s} = [\mathfrak{s}]$, $l \notin X$, C is equational, and furthermore no operator in Σ has $\mathfrak{s}$ as argument sort.

A rewrite theory $\mathcal{R} = (\Sigma, E, R)$ *entails* a Σ-sentence $(\forall X)\varphi$, of the form $(\forall X)\ t \to u$ **if** C or of the form $(\forall X)\ t \to u$ **if** $C \wedge D$, written $\mathcal{R} \vdash (\forall X)\varphi$, if and only if $(\forall X)\varphi$ can be obtained by finite application of the deduction rules in [1] if and only if $(\forall X)\varphi$ is valid in all models of $\mathcal{R}$. By definition, for $(\forall X)\varphi'$ an equational Σ-sentence, $\mathcal{R} \vdash (\forall X)\varphi'$ if and only if $\mathcal{E}_\mathcal{R} \vdash (\forall X)\varphi'$. A rewrite theory $\mathcal{R} = (\Sigma, E, R)$ induces the rewrite relation $\to_\mathcal{R}$ on $T_{\Sigma/E}(X)$ defined for every $t, u \in T_\Sigma(X)$ by $[t]_E \to_\mathcal{R} [u]_E$ if and only if there is an *one-step* rewrite proof $\mathcal{R} \vdash t \xrightarrow{1} u$. In the rest of this paper $\mathcal{R} \vdash t \to u$ and $\mathcal{R} \vdash t \xrightarrow{*} u$ respectively denote a one-step rewrite proof and an arbitrary length (but finite) rewrite proof in $\mathcal{R}$ of sequent $t \to u$. $\mathcal{T}_\mathcal{R} = (\mathcal{T}_{\Sigma/E}, \xrightarrow{*}_\mathcal{R})$ is the *initial reachability model* of $\mathcal{R} = (\Sigma, E, R)$ [1]. We write $\mathcal{R} \Vdash (\forall X)\varphi$ as an abbreviation for $\mathcal{T}_\mathcal{R} \models (\forall X)\varphi$, i.e., sentence $(\forall X)\varphi$ is an inductive consequence of $\mathcal{R}$. By definition, for $(\forall X)\varphi'$ an equational Σ-sentence, $\mathcal{R} \Vdash (\forall X)\varphi'$ if and only if $\mathcal{E}_\mathcal{R} \Vdash (\forall X)\varphi'$.

We assume that all rewrite theories treated in this paper satisfy the following *executability conditions*; these conditions guarantee agreement between the equational semantics and the operational semantics obtained by rewriting.

First, it is reasonable to have a disjoint union $E \uplus A$ of sets of equations in $\mathcal{R} = (\Sigma, E \uplus A, R)$, with A a collection of axioms (such as associativity, and/or commutativity, and/or identity) for which there exists a *matching algorithm modulo A* producing a finite number of A-matching substitutions, or failing otherwise. The second condition is that the equations E can be oriented into a set of *ground sort-decreasing*, *ground confluent*, and *ground terminating* rewrite rules $\overrightarrow{E}$ modulo A. This means that in the rewrite theory $\mathcal{R}_E = (\Sigma, A, \overrightarrow{E})$: (i) for each $s \in S$ and $[t]_A \in T_{\Sigma/A,s}$, $[t]_A \to_{\mathcal{R}_E} [u]_A$ implies $[u]_A \in T_{\Sigma/A,s}$, and (ii) for each sort $s \in S$ and for each $[t]_A \in T_{\Sigma/A,s}$ all maximal $\to_{\mathcal{R}_E}$-sequences beginning with $[t]_A$ terminate in a *unique* A-equivalence class $[\mathrm{can}_{\Sigma,E/A}(t)]_A \in T_{\Sigma/A,s}$, called the *$E$-canonical form* of $[t]_A$. The third condition is that the rules R should be *ground coherent* relative to the equations E modulo A [20]. This precisely means that, in the rewrite theories $\mathcal{R}_E = (\Sigma, A, \overrightarrow{E})$ and $\mathcal{R}_R = (\Sigma, A, R)$ (which have decidable rewrite relations $\to_{\mathcal{R}_E}$ and $\to_{\mathcal{R}_R}$ because of the assumptions on A), for each A-equivalence class $[t]_A$ such that $[t]_A \to_{\mathcal{R}_R} [u]_A$ there is a rewrite $[\mathrm{can}_{\Sigma,E/A}(t)]_A \to_{\mathcal{R}_R} [v]_A$ such that $[\mathrm{can}_{\Sigma,E/A}(u)]_A = [\mathrm{can}_{\Sigma,E/A}(v)]_A$. Intuitively, ground coherence means that any rewriting with R modulo $E \uplus A$

can be equivalently achieved by adopting the strategy of first simplifying a term to canonical form with E modulo A, and then applying a rule in R modulo A.

Given a rewrite theory $\mathcal{R} = (\Sigma, E \uplus A, R)$ we distinguish a signature $\Omega \subseteq \Sigma$ of *E-free constructors* modulo A, i.e., an order-sorted signature Ω such that for each sort s in Σ and $t \in T_{\Sigma,s}$ there is $u \in T_{\Omega,s}$ such that $t =_{E \uplus A} u$, and for any $v \in T_{\Omega,s}$ $\mathrm{can}_{\Sigma,E/A}(v) =_A v$, and require for each $(\forall X)\ l \to r$ **if** $C \in R$ that $l \in T_\Omega(X)$. The requirement of having a signature of E-free constructors for $\mathcal{E}_\mathcal{R}$ is met in practice by a very wide class of rewrite theories $\mathcal{R}$ specifying concurrent systems, since the equations E are introduced solely for the purpose of axiomatizing defined function symbols in terms of constructors. Following the same philosophy, and because $\mathcal{R}$ is ground coherent and $\mathcal{E}_\mathcal{R}$ is sufficiently complete relative to Ω, it is enough have Ω-constructor terms $l \in T_\Omega(X)_\mathtt{s}$ in the left-hand side of each rewrite rule $(\forall X)\ l \to r$ **if** $C \in R$ in order to axiomatize any transition $[t]_{E \uplus A} \to_\mathcal{R} [u]_{E \uplus A}$ in $\mathcal{T}_\mathcal{R}$ since $\mathrm{can}_{\Sigma,E/A}(t) \in T_{\Omega,\mathtt{s}}$.

3 Ground Stability

Rewrite theories are used to specify concurrent systems. Specifically, the initial reachability model $\mathcal{T}_\mathcal{R} = (\mathcal{T}_{\Sigma/E}, \overset{*}{\to}_\mathcal{R})$ of a topmost rewrite theory $\mathcal{R} = (\Sigma, E, R)$ specifies a concurrent system whose set of states is $T_{\Sigma/E,\mathtt{s}}$ and whose one-step transition relation is $\to_\mathcal{R}$. $\mathcal{T}_\mathcal{R}$ may be infinite-state, so that standard model checking algorithms may not be usable to verify its properties and a deductive approach is needed. We seek proof methods that can reduce reasoning about basic safety properties of $\mathcal{T}_\mathcal{R}$ to inductive equational reasoning about $\mathcal{T}_{\Sigma/E}$.

Let p be a state predicate defined on the set of states $T_{\Sigma/E,\mathtt{s}}$ of $\mathcal{T}_\mathcal{R}$. The property p being *(ground) stable* for $\mathcal{R}$ is the safety property

$$\mathcal{T}_\mathcal{R} \models p \Rightarrow \Box p.$$

That is, whenever p holds in a given state $[t]_E \in T_{\Sigma/E,\mathtt{s}}$, then p holds for all states $[t']_E$ such that $[t]_E \overset{*}{\to}_\mathcal{R} [t']_E$, i.e., once p becomes true, it remains true forever. This setting has to be generalized slightly, since, in general, the predicate p may not be defined in $\mathcal{R}$: it may be defined only later in the development process, when the safety properties of $\mathcal{T}_\mathcal{R}$ are specified and verified by extending $\mathcal{E}_\mathcal{R}$ with a set Π of suitable state predicates and their equational definitions.

The set of state predicates Π for $\mathcal{R} = (\Sigma, E, R)$ can be equationally-defined in an equational theory $\mathcal{E}_\Pi = (\Sigma_\Pi, E \uplus E_\Pi)$. The order-sorted signature Σ_Π contains Σ, two sorts Bool $\leq$ [Bool] with constants $\top$ and $\bot$ of sort Bool, predicate symbols $p : \mathtt{s} \longrightarrow$ [Bool] for each $p \in \Pi$, and optionally some auxiliary function symbols. The equations in E_Π define the predicate symbols in Σ_Π and the auxiliary function symbols, if any, and they protect[1] both (Σ, E) and the theory BOOL specifying the sort Bool, $\top$, $\bot$, and the Boolean operations.

[1] A theory inclusion $(\Sigma, E) \subseteq (\Sigma', E')$ is *protecting* if and only if the unique Σ-homomorphism $T_{\Sigma/E} \longrightarrow T_{\Sigma'/E'}|_\Sigma$ to the Σ-reduct of the initial algebra $\mathcal{T}_{\Sigma'/E'}$ is an isomorphism.

Given a state predicate $p \in \Pi$ and given a state $[t]_E \in T_{\Sigma/E,\mathfrak{s}}$, $\mathcal{E}_\Pi$ then defines the *semantics of* p in $\mathcal{T}_\mathcal{R}$ as follows: we say that $p([t]_E)$ *holds* in $\mathcal{T}_\mathcal{R}$ if and only if

$$\mathcal{E}_\Pi \vdash p(t) = \top.$$

This defines a Kripke structure $\mathcal{K}^\Pi_\mathcal{R} = (T_{\Sigma/E,\mathfrak{s}}, \rightarrow_\mathcal{R}, L_\Pi)$ with labeling function L_Π such that, for each $[t]_E \in T_{\Sigma/E,\mathfrak{s}}$, $p \in L_\Pi([t]_E)$ if and only if $\mathcal{E}_\Pi \vdash p(t) = \top$. Therefore, we can interpret all of LTL (also its first-order version) in $\mathcal{K}^\Pi_\mathcal{R}$ in the standard way [3]. We will use this remark in what follows to make explicit how some of our results can be understood as inference rules for reasoning about LTL properties of $\mathcal{K}^\Pi_\mathcal{R}$.

Note that only the positive case is needed to define p's semantics. The reason why p has type $p : \mathfrak{s} \longrightarrow$ [Bool] instead of $p : \mathfrak{s} \longrightarrow$ Bool, is to allow partial definitions of p with equations that only define the *positive* case by equations $p(t) = \top$ **if** C, and either leave the *negative* case implicit or may only define some negative cases with equations $p(t') = \bot$ **if** C' without necessarily covering all the cases, i.e., without p's definition having to be sufficiently complete. This possibly partial specification of predicates (yet, with *full* specification in the *positive* case) can be very convenient, since the full definition of the negative cases can sometimes be quite involved. However, the sort Bool is *protected*: only when a term $p(t)$ can be proved equal to either $\top$ or $\bot$ can the term $p(t)$ have sort Bool. Nevertheless, for proving purposes it is often useful to define some negative cases for which a state predicate p does not hold, since this helps in discarding proof obligations in the form of an implication whose antecedent is false.

It is important to note that a state predicate $p \in \Pi$ can act as a *definitional extension* of a Boolean combination of other state predicates $\{p_1, \ldots, p_n\}$ in Σ_Π, so that our choice of focusing on atomic state predicates is mainly to simplify the exposition but does not limit the general applicability of the results that follow. In a rewriting logic language implementation such as Maude [4], definitional extensions can be conveniently obtained by having $\mathcal{E}_\Pi$ protecting Maude's predefined equational theory BOOL-OPS, which declares constants $\top$ and $\bot$ of sort Bool along with Boolean function symbols such as conjunction, disjunction, negation, etc.

The concept of ground stability for a topmost rewrite theory $\mathcal{R}$ is intimately related with the notion of sets of states of $\mathcal{T}_\mathcal{R}$ being closed under the rewrite relation $\rightarrow_\mathcal{R}$. Namely, $\mathcal{R}$ being ground p-stable exactly means that the set of states of $\mathcal{T}_\mathcal{R}$ satisfying p is closed under the rewrite relation $\rightarrow_\mathcal{R}$.

Definition 1. *Let $\mathcal{R} = (\Sigma, E, R)$ be a topmost rewrite theory and let Π be a set of state predicates for $\mathcal{R}$ equationally defined in $\mathcal{E}_\Pi = (\Sigma_\Pi, E \uplus E_\Pi)$. For $p \in \Pi$, $\mathcal{R}$ is called* ground p-stable *under $R_0 \subseteq R$ if and only if, for each $t, u \in T_{\Sigma,\mathfrak{s}}$, $\mathcal{E}_\Pi \vdash p(t) = \top$ and $(\Sigma, E, R_0) \vdash t \stackrel{*}{\rightarrow} u$ imply $\mathcal{E}_\Pi \vdash p(u) = \top$. $\mathcal{R}$ is* ground p-stable*, written $\mathcal{R} \Vdash p \Rightarrow \Box p$, if and only if $\mathcal{R}$ is ground p-stable under R.*

For a topmost rewrite theory $\mathcal{R} = (\Sigma, E, R)$, the reachability condition in the definition of ground stability can be reduced to a simpler 1-step rewrite condition,

resulting in an equivalent notion of ground stability that avoids arbitrary depth proof-search.

Lemma 1. *Let $\mathcal{R}$, $\mathcal{E}_{\Pi}$, p, and R_0 be as in Definition 1. Then $\mathcal{R}$ is ground p-stable under R_0 if and only if, for each $t, u \in T_{\Sigma,\mathtt{s}}$, $\mathcal{E}_{\Pi} \vdash p(t) = \top$ and $(\Sigma, E, R_0) \vdash t \to u$ imply $\mathcal{E}_{\Pi} \vdash p(u) = \top$.*

Proof. Let t and u be as above and let $\mathcal{R}_0 = (\Sigma, E, R_0)$. ($\Rightarrow$) $\mathcal{R}_0 \vdash t \to u$ implies $\mathcal{R}_0 \vdash t \xrightarrow{*} u$, and therefore $\mathcal{E} \vdash p(t) = \top$ and $\mathcal{R}_0 \vdash t \to u$ imply $\mathcal{E}_{\Pi} \vdash p(u) = \top$ because $\mathcal{R}$ is p-stable under $\mathcal{R}_0$ by hypothesis. ($\Leftarrow$) By induction on the proof length n of $\mathcal{R}_0 \vdash t \xrightarrow{n} u$. If $n = 0$ then $\mathcal{R}_0 \vdash t = u$, and by definition $t =_E u$. Since $\mathcal{E}_{\Pi} \vdash p(t) = \top$, it must be the case that $\mathcal{E}_{\Pi} \vdash p(u) = \top$. If $\mathcal{R}_0 \vdash t \xrightarrow{n+1} u$, then there is $u_0 \in T_{\Sigma,\mathtt{s}}$ such that $\mathcal{R}_0 \vdash t \to u_0 \wedge u_0 \xrightarrow{n} u$. If $\mathcal{E}_{\Pi} \vdash p(t) = \top$, then $\mathcal{E}_{\Pi} \vdash p(u_0) = \top$ by hypothesis, which together with the induction hypothesis $\mathcal{R}_0 \vdash u_0 \xrightarrow{n} u$, imply $\mathcal{E}_{\Pi} \vdash p(u) = \top$. □

In the notation of Linear Time Temporal Logic (LTL), Lemma 1 justifies the soundness of the the inference rule G-St in Figure 1, which shows how to reason in LTL about the p-stability of the Kripke structure $\mathcal{K}^{\Pi}_{\mathcal{R}} = (T_{\Sigma/E,\mathtt{s}}, \to_{\mathcal{R}}, L_{\Pi})$ associated to a topmost rewrite theory $\mathcal{R} = (\Sigma, E, R)$ and a set of state predicates Π. Symbol "$\bigcirc$" corresponds to the next operator in LTL and symbol "$\Rightarrow$" to strong implication in LTL (see [10] for details). So, for $\mathcal{K}^{\Pi}_{\mathcal{R}} \models p \Rightarrow \Box p$ to hold, it is enough to show that $\mathcal{K}^{\Pi}_{\mathcal{R}} \models p \Rightarrow \bigcirc p$ holds, i.e., p holds for any state $[t']_E$ reachable in one-step with $\to_{\mathcal{R}}$ from a state $[t]_E$ satisfying p. As a matter of fact, Lemma 1 also shows that the converse of rule G-St is also sound.

$$\frac{\mathcal{R} \Vdash p \Rightarrow \bigcirc p}{\mathcal{R} \Vdash p \Rightarrow \Box p}\ \text{G-St}$$

$$\frac{\begin{array}{c}\textit{for each } ((\forall X)\ l \to r \ \textbf{if}\ C) \in R \textit{ and } (\theta, w, D) \in \Theta_{(l,r,C)} : \\ \mathcal{E}_{\Pi} \Vdash (\forall \mathrm{ran}(\theta))\ (C\theta \wedge D\theta \wedge w\theta = \top) \Rightarrow p(r\theta) = \top\end{array}}{\mathcal{R} \Vdash p \Rightarrow \bigcirc p}\ \text{Nr1}$$

Fig. 1. Checking $\mathcal{R} = (\Sigma, E, R)$ ground p-stable (with $\Theta_{(l,r,C)}$ in rule Nr1 defined as in Theorem 1).

The next question to ask is how to reduce to inductive equational reasoning the verification of the simpler condition $p \Rightarrow \bigcirc p$ in Rule G-St under some assumptions about the equations E and rewrite rules R in $\mathcal{R}$. Our approach is to reduce the inductive reachability problem of p-stability for $\mathcal{T}_{\mathcal{R}}$ to equational inductive properties of $\mathcal{T}_{\mathcal{E}_{\mathcal{R}}}$ using the idea of (one-step) *narrowing with equations modulo axioms* [9], a sound and complete method for ground stability analysis.

Under the executability assumptions, $\mathcal{R}$ has a disjoint union $E \uplus A$ of equations, with A a collection of structural axioms on the function symbols in Σ such as associativity, commutativity, identity, etc., and E a ground sort-decreasing, ground confluent, ground terminating, and ground coherent w.r.t. R set of equations modulo A. For a combination of free and associative and/or commutative and/or identity axioms, except for symbols f that are associative but not commutative, a finitary A-unification algorithm exists. Instead, in general there is no finitary $E \uplus A$-unification algorithm. Nevertheless, for $\Omega \subseteq \Sigma$ a signature of E-free equational constructors modulo A (which is assumed to exists for the rewrite theories considered in this paper –see Section 2) and an Ω-equality $t = u$, $\mathrm{CSU}_A(t = u)$ exactly characterizes as instances the set $\mathrm{GU}_{E \uplus A}(t = u)$ of ground $E \uplus A$-unifiers for $t = u$.

Lemma 2. *Let $\mathcal{E} = (\Sigma, E \uplus A)$ be an (executable) order-sorted equational theory with finitary A-unification algorithm, and let $\Omega \subseteq \Sigma$ be a signature of E-free constructors modulo A. Then, for any Ω-equality $t = u$, $\alpha \in \mathrm{GU}_{E \uplus A}(t = u)$ if and only if there exists $\theta \in \mathrm{CSU}_A(t = u)$ and ground substitution $\gamma : \mathrm{vars}(\theta) \longrightarrow T_\Omega$ such that $\theta\gamma =_{E \uplus A} \alpha$.*

Proof. Let $t, u \in T_\Omega(X)_s$ for some sort s in Σ. ($\Rightarrow$) Assume $\alpha \in \mathrm{GU}_{E \uplus A}(t = u)$, i.e., $\alpha : \mathrm{vars}(\{t, u\}) \longrightarrow T_\Sigma$ is such that $t\alpha =_{E \uplus A} u\alpha$. Since Ω is a subsignature of constructors and Ω is E-free, there is $\beta : \mathrm{dom}(\alpha) \longrightarrow T_\Omega$ such that $\alpha =_{E \uplus A} \beta$ and for $x \in \mathrm{dom}(\alpha)$, $\mathrm{can}_{\Sigma, E/A}(\alpha(x)) =_A \beta(x)$. Consequently $t\beta =_A u\beta$, and therefore there is $\theta \in \mathrm{CSU}_A(t = u)$ and $\gamma : \mathrm{vars}(\beta) \longrightarrow T_\Omega$ such that $\theta\gamma =_A \beta =_{E \uplus A} \alpha$. ($\Leftarrow$) Suppose $\theta \in \mathrm{CSU}_A(t = u)$ and let $\gamma : \mathrm{vars}(\theta) \longrightarrow T_\Omega$ be a ground substitution. Then, we have $t\theta\gamma =_A u\theta\gamma$ and, a fortiori, $t\theta\gamma =_{E \uplus A} u\theta\gamma$. Therefore, $\theta\gamma \in \mathrm{GU}_{E \uplus A}(t = u)$ as desired. □

In order to show the ground p-stability of $\mathcal{R} = (\Sigma, E \uplus A, R)$, we need to prove for each rule $(\forall X)\ l \to r$ **if** $C \in R$ that if $p(l) = \top$, then $p(r) = \top$. The key observation here is that, since by assymption $l \in T_{\Omega,s}(X)$, if all left hand-sides $p(v)$ of equations $p(v) = w$ **if** $D \in E_\Pi$ defining the state predicate $p \in \Pi$ are *Ω-patterns* (i.e., $v \in T_\Omega(X)$), then we can compute $\mathrm{CSU}_A(l = v)$ and obtain substitutions θ which, by Lemma 2, exactly characterize any ground $E \uplus A$-unifier in $\mathrm{GU}_{E \uplus A}(l = v)$. Each substitution $\theta \in \mathrm{CSU}_A(l = v)$ is such that $p(l\theta) = \top$, or at least $p(l\theta)$ *could* be equal to $\top$, and thus we are left with the task of proving inductively that $p(r\theta) = \top$ holds under the assumptions $C\theta \wedge D\theta \wedge w\theta = \top$. In this way, the inductive reachability problem of p-stability for $\mathcal{T}_\mathcal{R}$ can be reduced to the simpler aforementioned equational inductive properties of $\mathcal{T}_{\Sigma/E \uplus A}$, in the sense that if $\mathcal{T}_{\Sigma/E \uplus A}$ satisfies these inductive properties then $\mathcal{T}_\mathcal{R}$ is ground p-stable. Theorem 1 justifies the soundness of the narrowing inference rule NR1 in Figure 1.

Theorem 1. *Let $\mathcal{R} = (\Sigma, E \uplus A, R)$ be a topmost rewrite theory with signature $\Omega \subseteq \Sigma$ of equational E-free constructors modulo A and with finitary A-unification algorithm, and let $\mathcal{E}_\Pi = (\Sigma_\Pi, E \uplus A \uplus E_\Pi)$ be an equational definition of Π for $\mathcal{R}$. Let $p \in \Pi$ and $(\forall Y)\ l \to r$ **if** $C \in R$. Without loss of generality,*

assume that the equations $E^p_\Pi \subseteq E_\Pi$ defining $p \in \Pi$ are all conditional, have no variables in Y, and have Ω-patterns as left-hand sides, and define

$$\Theta_{(l,r,C)} = \bigcup_{((\forall Z)\ p(v)=w\ \mathbf{if}\ D)\in E^p_\Pi} \{(\theta, w, D) \mid \theta \in \mathrm{CSU}_A(v = l)\}.$$

*Then, $\mathcal{R}$ is ground p-stable under $(\forall Y)\ l \to r$ **if** C if and only if for each $(\theta, w, D) \in \Theta_{(l,r,C)}$*

$$\mathcal{E}_\Pi \Vdash (\forall \mathrm{ran}(\theta))\ C\theta \wedge D\theta \wedge w\theta = \top \ \Rightarrow\ p(r\theta) = \top.$$

Proof. Let $R_0 = \{(\forall Y)\ l \to r\ \mathbf{if}\ C\}$ and $\mathcal{R}_0 = (\Sigma, E \uplus A, R_0)$:

$\mathcal{R}$ is ground p-stable under R_0
iff { by definition of ground p-stability and Lemma 1 }
$(\forall t, u \in T_{\Sigma,\mathfrak{s}})$
$\mathcal{E}_\Pi \vdash p(t) = \top$ and $\mathcal{R}_0 \vdash t \to u$ implies $\mathcal{E}_\Pi \vdash p(u) = \top$
iff { by definition of rewriting and by $\mathcal{E}_{\mathcal{R}_0} = \mathcal{E}_\mathcal{R}$ }
$(\forall \alpha : Y \longrightarrow T_\Sigma)$
$\mathcal{E}_\Pi \vdash p(l\alpha) = \top$ and $\mathcal{E}_\mathcal{R} \vdash C\alpha$ implies $\mathcal{E}_\Pi \vdash p(r\alpha) = \top$
iff { by $\mathcal{E}_\Pi$ protecting $\mathcal{E}_\mathcal{R}$ and $C\alpha$ a ground Σ-formula }
$(\forall \alpha : Y \longrightarrow T_\Sigma)$
$\mathcal{E}_\Pi \vdash (p(l\alpha) = \top \wedge C\alpha) \Rightarrow p(r\alpha) = \top$
iff { by $\mathcal{E}_\Pi$ ground confluent, with $Z_v = \mathrm{vars}(v)$ }
$(\forall \alpha : Y \longrightarrow T_\Sigma)(\forall (p(v) = w\ \mathbf{if}\ D) \in E^p_\Pi)(\forall \beta : Z_v \longrightarrow T_\Sigma)$
$\mathcal{E}_\Pi \vdash (l\alpha = v\beta \wedge D\beta \wedge w\beta = \top \wedge C\alpha) \Rightarrow p(r\alpha) = \top$
iff { by assumption $Y \cap Z_v = \emptyset$, with $\eta = \alpha \uplus \beta$ and $X_v = Y \uplus Z_v$ }
$(\forall (p(v) = w\ \mathbf{if}\ D) \in E^p_\Pi)(\forall \eta : X_v \longrightarrow T_\Sigma)$
$\mathcal{E}_\Pi \vdash (l\eta = v\eta \wedge D\eta \wedge w\eta = \top \wedge C\eta) \Rightarrow p(r\eta) = \top$
iff { by Lemma 2: $l, v \in T_\Omega(X)_\mathfrak{s}$ and $\mathcal{E}_\Pi$ protecting $\mathcal{E}_\mathcal{R}$ }
$(\forall (p(v) = w\ \mathbf{if}\ D) \in E^p_\Pi)(\forall \theta \in \mathrm{CSU}_A(l = v))$
$(\forall \gamma : \mathrm{ran}(\theta) \longrightarrow T_\Sigma)$
$\mathcal{E}_\Pi \vdash (D\theta\gamma \wedge w\theta\gamma = \top \wedge C\theta\gamma) \Rightarrow p(r\theta\gamma) = \top$
iff { by definition of $\Vdash$ }
$(\forall (p(v) = w\ \mathbf{if}\ D) \in E^p_\Pi)(\forall \theta \in \mathrm{CSU}_A(l = v))$
$\mathcal{E}_\Pi \Vdash (\forall \mathrm{ran}(\theta))\ (C\theta \wedge D\theta \wedge w\theta = \top) \Rightarrow p(r\theta) = \top$
iff { by definition of $\Theta_{(l,r,C)}$ }
$(\forall (\theta, w, D) \in \Theta_{(l,r,C)})$
$\mathcal{E}_\Pi \Vdash (\forall \mathrm{ran}(\theta))\ (C\theta \wedge D\theta \wedge w\theta = \top) \Rightarrow p(r\theta) = \top$

□

Observe that obtaining the complete set of unifiers in the definition of $\Theta_{(l,r,C)}$ in Theorem 1 only involves Σ-terms and not Σ_Π-terms. This is useful in practice because the generation of proof obligations from $\Theta_{(l,r,C)}$ does not depend on the state predicates defined in $\mathcal{E}_\Pi$ and therefore is not affected by their equational definitions, no matter how involved these definitions may be. Also observe that, since the complete set of A-unifiers is finite, the set $\Theta_{(l,r,C)}$ is also finite for

each rewrite rule $(\forall Y)\ l \to r$ **if** $C \in R$. Therefore, the set of proof obligations ensuring ground p-stability for $\mathcal{R}$ is finite because of the finiteness assumptions on E and R (see Section 2). As a final remark, observe that the case when w is $\bot$ in an equation $(\forall Z)\ p(v) = w$ **if** $D \in E^p_\Pi$, each proof obligation $(\forall \mathrm{ran}(\theta))\ C\theta \wedge D\theta \wedge w\theta = \top \Rightarrow p(r\theta) = \top$ can be *soundly* ignored because $w\theta = \bot\theta = \bot$ and $\mathcal{E}_\Pi$ protects the sort Bool (i.e., $\top$ and $\bot$ are never provably equal).

Example 1. Consider the following version of Lamport's bakery protocol, borrowed and slightly adapted from [6], in which there are several processes, each with its internal state and possibly with a natural number, that achieve mutual exclusion by the usual method common in bakeries and deli shops: there is a number dispenser and customers are served in sequential order according to the ticket that they hold. This system can be specified in Maude as a topmost rewrite theory BAKERY with top sort State, as follows:

```
fmod BAKERY-SYNTAX is
 pr NAT .
 sorts ProcIdle ProcWait Proc .
 sorts ProcIdleSet ProcWaitSet ProcSet .
 subsorts ProcIdle < ProcIdleSet .
 subsorts ProcWait < ProcWaitSet .
 subsorts ProcIdle ProcWait < Proc < ProcSet .
 subsorts ProcIdleSet < ProcWaitSet < ProcSet .
 op idle : -> ProcIdle [ctor] .
 op wait : Nat -> ProcWait [ctor] .
 op crit : Nat -> Proc [ctor] .
 op none : -> ProcIdleSet [ctor] .
 op __ : ProcIdleSet ProcIdleSet
           -> ProcIdleSet [ctor assoc comm id: none] .
 op __ : ProcWaitSet ProcWaitSet
           -> ProcWaitSet [ditto] .
 op __ : ProcSet ProcSet -> ProcSet [ditto] .
 sort State .
 op _:_[_] : Nat Nat ProcSet -> State [ctor] .
endfm

mod BAKERY is
 pr BAKERY-SYNTAX .
 var Ps : ProcSet .  vars N M : Nat .
 rl [get]   : N : M [idle Ps] => s(N) : M [wait(N) Ps] .
 rl [serve] : N : M [wait(M) Ps] => N : M [crit(M) Ps] .
 rl [leave] : N : M [crit(M) Ps] => N : s(M) [idle Ps] .
endm
```

The equations in BAKERY are *all* structural axioms, namely, associativity, commutativity, and identity for sets of processes, for which an algorithm computing a complete set of unifiers exists. Since there are no equations besides the structural axioms, BAKERY's signature is trivially a signature of equational free constructors modulo BAKERY's structural axioms.

A ground term "$n : m\ [ps]$" of sort State describes the state in which the natural number n is the number of the next available ticket, the natural number m is the number of the next ticket to be served, and ps is the set of customers currently in the "bakery".

Here we are interested in the set of state predicates $\Pi = \{\text{bounded-tickets}\}$, expressing that tickets among customers are all bounded from above. State predicate bounded-tickets can be equationally defined by means of the auxiliary functions sub-bag, tickets, and tickets-below. BAKERY-PROPS defines sort NatBag for bags (or multisets) of natural numbers; mtbag denotes the empty bag and bag union is denoted by juxtaposition modulo associativity, commutativity, and identity (with mtbag being the identity bag).

```
fmod BAKERY-PROPS is
 pr BAKERY-SYNTAX .
 pr BOOL-OPS .
 sort NatBag .
 subsort Nat < NatBag .
 op mtbag : -> NatBag .
 op __ : NatBag NatBag -> NatBag [assoc comm id: mtbag] .
 op bounded-tickets : State -> [Bool] .
 op tb : Nat -> NatBag .
 op tkts : ProcSet -> NatBag .
 op sb : NatBag NatBag -> [Bool] .
 var Is : ProcIdleSet .  var Ps : ProcSet .
 vars N M : Nat .  vars NB NB' : NatBag .
 eq [1] : bounded-tickets(N : M [Ps]) = sb(tkts(Ps),tb(N)) .
 eq [a.1] : sb(NB,NB NB') = ⊤ .
 eq [a.2] : sb(N NB,N NB') = sb(NB,NB') .
ceq [a.3] : sb(N NB,NB') = ⊥ if in?(N,NB') = ⊤.
 eq [b.1] : tkts(Is) = mtbag .
 eq [b.2] : tkts(idle Ps) = tkts(Ps) .
 eq [b.3] : tkts(wait(N) Ps) = N tkts(Ps) .
 eq [b.4] : tkts(crit(N) Ps) = N tkts(Ps) .
 eq [c.1] : tb(0) = mtbag .
 eq [c.2] : tb(s(N)) = N tb(N) .
endfm
```

We want to prove that BAKERY is ground bounded-tickets-stable. According to Lemma 1 and Theorem 1, this holds if the following three sentences are inductive theorems of $\mathcal{E}_{\text{BAKERY-PROPS}}$:

$$(\forall x_1, x_2 : \text{Nat}; x_3 : \text{ProcSet})$$

$$\text{bounded-tickets}(s(x_1) : x_2[\text{wait}(x_1)]) = \top \textbf{ if } \text{sb}(\text{tkts}(\text{idle}), \text{tb}(x_1)) = \top, \quad (1)$$

$$\text{bounded-tickets}(s(x_1) : x_2[x_3\ \text{wait}(x_1)]) = \top \textbf{ if } \text{sb}(\text{tkts}(\text{idle}\, x_3), \text{tb}(x_1)) = \top, \quad (2)$$

$$\text{bounded-tickets}(x_1 : x_2[\text{crit}(x_2)]) = \top \textbf{ if } \text{sb}(\text{tkts}(\text{wait}(x_2)), \text{tb}(x_1)) = \top, \quad (3)$$

$$\text{bounded-tickets}(x_1 : x_2[\text{crit}(x_2)\, x_3]) = \top \textbf{ if } \text{sb}(\text{tkts}(\text{wait}(x_2)\, x_3), \text{tb}(x_1)) = \top, \quad (4)$$

$$\text{bounded-tickets}(x_1 : s(x_2)[\text{idle}]) = \top \textbf{ if } \text{sb}(\text{tkts}(\text{crit}(x_2)), \text{tb}(x_1)) = \top, \quad (5)$$

$$\text{bounded-tickets}(x_1 : s(x_2)[\text{idle}\, x_3]) = \top \textbf{ if } \text{sb}(\text{tkts}(\text{crit}(x_2)\, x_3), \text{tb}(x_1)) = \top. \quad (6)$$

Sentences (1) and (2), (3) and (4), and (5) and (6) are obtained from equation 1 and rules get, serve, and leave, respectively. Sentences (1) and (5) have trivial consequents that can be automatically discharged by equational rewriting. Sentences (2)–(4) follow automatically by assuming the conditions (see Section 6 for a brief explanation of this technique). Sentence (6) can be discharged by Maude's ITP [8] with minor user interaction.

4 Ground Invariance

The most important safety properties are *invariants*. Given a set of initial states characterized by $I \in \Pi$, $p \in \Pi$ is a *ground invariant* for a topmost rewrite theory $\mathcal{R}$ from initial states in I if and only if

$$\mathcal{T}_{\mathcal{R}} \models I \Rightarrow \Box p.$$

That is, for any $[t]_E \in T_{\Sigma/E,\mathfrak{s}}$ such that $I([t]_E)$ holds, and for any state $[t']_E \in T_{\Sigma/E,\mathfrak{s}}$ such that $[t]_E \xrightarrow{*}_{\mathcal{R}} [t']_E$, then $p([t']_E)$ must hold. In other words, the invariant p holds for all states reachable from I. Since the set of initial states is defined in $\mathcal{E}_\Pi$ as a state predicate $I \in \Pi$, an equational definition of I can of course capture an infinite set of initial states.

Definition 2. *Let $\mathcal{R} = (\Sigma, E, R)$ be a topmost rewrite theory and let Π be a set of state predicates for $\mathcal{R}$ equationally defined by $\mathcal{E}_\Pi = (\Sigma_\Pi, E \uplus E_\Pi)$. For $p, I \in \Pi$, $\mathcal{R}$ is called* ground p-invariant *from I under $R_0 \subseteq R$ if and only if, for each $t, u \in T_{\Sigma,\mathfrak{s}}$, $\mathcal{E}_\Pi \vdash I(t) = \top$ and $(\Sigma, E, R_0) \vdash t \xrightarrow{*} u$ imply $\mathcal{E}_\Pi \vdash p(u) = \top$. $\mathcal{R}$ is* ground p-invariant *from I, written $\mathcal{R} \Vdash I \Rightarrow \Box p$, if and only if $\mathcal{R}$ is ground p-invariant from I under R.*

Ground p-invariance of a topmost rewrite theory $\mathcal{R}$ is intimately related to its ground p-stability in the sense that if every initial state defined by a predicate I satisfies p and $\mathcal{R}$ is p-stable, then $\mathcal{R}$ is p-invariant from I. Of course, the converse does not necessarily hold, because even if $\mathcal{R}$ is ground p-invariant from I, the set of states of $\mathcal{T}_{\mathcal{R}}$ satisfying p need not be closed under $\rightarrow_{\mathcal{R}}$. The key observation is that in $\mathcal{T}_{\mathcal{R}}$, when every initial state defined by I satisfies p, the set of states satisfying p characterizes an over-approximation of the set of reachable states from the set of initial states specified by I.

Theorem 2. *Let $\mathcal{R}$, Π, $\mathcal{E}_\Pi$, p, and I be as in Definition 2. Then, $\mathcal{R}$ is ground p-invariant from I under $R_0 \subseteq R$ if*

1. *$\mathcal{E}_\Pi \Vdash (\forall x : \mathfrak{s})\, I(x) = \top \Rightarrow p(x) = \top$, and*
2. *$\mathcal{R}$ is ground p-stable under R_0.*

Proof. Let $t, u \in T_{\Sigma,\mathfrak{s}}$ and $\mathcal{R}_0 = (\Sigma, E, R_0)$. Assume $\mathcal{E}_\Pi \vdash I(t) = \top$ and $\mathcal{R}_0 \vdash t \xrightarrow{*} u$. Since $\mathcal{E}_\Pi \Vdash (\forall x : \mathfrak{s})\, I(x) = \top \Rightarrow p(x) = \top$ and $\mathcal{E}_\Pi \vdash I(t) = \top$, it follows that $\mathcal{E}_\Pi \vdash p(t) = \top$. $\mathcal{R}_0$ is ground p-stable and $\mathcal{R}_0 \vdash t \xrightarrow{*} u$, therefore $\mathcal{E}_\Pi \vdash p(u) = \top$. □

For a topmost rewrite theory $\mathcal{R} = (\Sigma, E, R)$ and state predicates $p, q \in \Pi$, we write $q \Rightarrow p$ as a shorthand for $(\forall x : \mathfrak{s})\, q(x) = \top \Rightarrow p(x) = \top$, and let $[\![q]\!]^{\mathcal{E}_\Pi} = \{[t]_E \in T_{\Sigma/E,\mathfrak{s}} \mid \mathcal{E}_\Pi \vdash q(t) = \top\}$ (or simply $[\![q]\!]$), for any $q \in \Pi$. Condition 1 in Theorem 2 states that every initial state specified by I must satisfy property p. That is, for Π and $\mathcal{E}_\Pi$ defined as in Theorem 2, $\mathcal{E}_\Pi \Vdash I \Rightarrow p$ holds if and only if $[\![I]\!] \subseteq [\![p]\!]$. Observe that this condition does not depend on the dynamics of $\mathcal{T}_\mathcal{R}$, but only on its set of states $T_{\Sigma/E,\mathfrak{s}}$. Conditions 1 and 2 in Theorem 2 are used in the literature to define the notion of *inductive invariant*, i.e., of a predicate holding in the set of initial states and mantained true by every transition.

In LTL terms, Theorem 2 justifies the soundness of the inference rule G-Inv in Figure 2 to prove that p is an invariant from I in the Kripke structure $\mathcal{K}^\Pi_\mathcal{R}$. The only remaining question is how to prove $I \Rightarrow p$. This question is answered by Theorem 3, which gives a sufficient and necessary condition for proving statements of the form $q \Rightarrow p$, for state predicates $p, q \in \Pi$. Theorem 3 justifies the soundness of the the inference rule C⇒ in Figure 2.

$$\frac{\mathcal{R} \Vdash I \Rightarrow p \qquad \mathcal{R} \Vdash p \Rightarrow \Box p}{\mathcal{R} \Vdash I \Rightarrow \Box p}\ \text{G-Inv}$$

$$\frac{\begin{array}{l}\textit{for each } ((\forall Y)\ q(v) = w \textbf{ if } C) \in E^q_\Pi : \\ \mathcal{E}_\Pi \Vdash (\forall Y) C \wedge w = \top \Rightarrow p(v) = \top\end{array}}{\mathcal{R} \Vdash q \Rightarrow p}\ \text{C}\Rightarrow$$

Fig. 2. Checking $\mathcal{R} = (\Sigma, E, R)$ ground p-invariant from I (with E^q_Π in rule C⇒ as defined in Theorem 3).

Theorem 3. *Let $\mathcal{R}$, Π, $\mathcal{E}_\Pi$, and p be as in Definition 2, and let $q \in \Pi$. Without loss of generality, assume equations $E^q_\Pi \subseteq E_\Pi$ defining $q \in \Pi$ are all conditional. Then $[\![q]\!] \subseteq [\![p]\!]$ if and only if for each $(\forall Y)\ q(v) = w$ **if** $C \in E^q_\Pi$*

$$\mathcal{E}_\Pi \Vdash (\forall Y)\ C \wedge w = \top \Rightarrow p(v) = \top.$$

Proof. (⇒) Let $(\forall Y)\ q(v) = w$ **if** $C \in E^q_\Pi$ and assume $\mathcal{E}_\Pi \vdash C\alpha \wedge w\alpha = \top$ for some ground substitution $\alpha : Y \longrightarrow T_\Sigma$; need to show $\mathcal{E}_\Pi \vdash p(v\alpha) = \top$. $\mathcal{E}_\Pi \vdash C\alpha \wedge w\alpha = \top$ implies $\mathcal{E}_\Pi \vdash q(v\alpha) = \top$ by means of equation $(\forall Y)\ q(v) = w$ **if** C, and therefore $\mathcal{E}_\Pi \vdash p(v\alpha) = \top$ by hypothesis. (⇐) Let $t \in T_{\Sigma,\mathfrak{s}}$ and assume $\mathcal{E}_\Pi \vdash q(t) = \top$. Then, there is $(\forall Y)\ q(v) = w$ **if** $C \in E^q_\Pi$ and ground substitution $\alpha : Y \longrightarrow T_\Sigma$ such that $\mathcal{E}_\Pi \vdash C\alpha \wedge w\alpha = \top$ and $t =_E v\alpha$. That is $C \wedge w = \top$ is satisfiable in $\mathcal{T}_\mathcal{R}$ and $\mathcal{E}_\Pi \Vdash (\forall Y)\ C \wedge w = \top \Rightarrow p(v) = \top$ by supposition, it follows that $\mathcal{E}_\Pi \vdash p(t) = \top$. □

Example 2. Recall Example 1 from Section 3. Here we are interested in state predicates $\Pi = \{\text{bounded-tickets}, \text{init}\}$ for `BAKERY`, with bounded-tickets as defined in BAKERY-PROPS. State predicate init defines the set of initial states for $\mathcal{T}_{\text{BAKERY}}$ in module BAKERY-PROPS-EXT1, which extends BAKERY-PROPS.

```
fmod BAKERY-PROPS-EXT1 is
 pr BAKERY-PROPS .
 op init : State -> [Bool] .
 var Is : ProcIdleSet .
 eq [2] : init(0 : 0 [Is]) = ⊤ .
endfm
```

An initial state for $\mathcal{T}_{\text{BAKERY}}$ is any state in which numbers corresponding to the next available ticket and the ticket to be served next are both zero, and all customers in the "bakery" are in state idle. Observe that no constraint is imposed on the initial number of customers.

We want to prove that BAKERY is ground bounded-tickets-invariant for init. According to Theorem 2, it is sufficient to prove (i) $\mathcal{E}_{\text{BAKERY-PROPS-EXT1}} \Vdash (\forall x : \mathfrak{s})\text{init}(x) = \top \Rightarrow \text{bounded-tickets}(x) = \top$ and (ii) BAKERY is ground bounded-tickets-stable. Example 1 gives a proof of condition (ii). By means of Theorem 3, condition (i) holds if Sentence (7) is an inductive theorem of $\mathcal{E}_{\text{BAKERY-PROPS-EXT1}}$:

$$(\forall x_1 : \text{ProcIdleSet})\ \text{bounded-tickets}(0 : 0\,[x_1]) = \top. \tag{7}$$

Observe that Sentence (7) admits a simple equational proof because $\text{tickets}(x_1) = \text{mtbag}$ and $\text{tickets-below}(0) = \text{mtbag}$. Therefore, we trivially have that property $\text{BAKERY} \Vdash \text{init} \Rightarrow \Box\text{bounded-tickets}$ holds.

5 Strengthenings for Ground Invariance

Strengthening of invariants is a key technique for verifying safety properties. For state predicates $p, I \in \Pi$, a *strengthening* for the ground p-invariance from I of a topmost rewrite theory $\mathcal{R}$ is given by a state predicate $q \in \Pi$ such that $\mathcal{R}$ is ground q-invariant from I and, moreover, q can be used to prove $\mathcal{R} \Vdash I \Rightarrow \Box p$. Traditionally, state predicate q is the result of a gradual refinement of a too-weakly defined p for which $\mathcal{R}$ being ground p-invariant cannot be proven directly by means of Theorem 2. This section presents two strengthening techniques for ground invariance, proves their correctness, and illustrates their application using our running example.

Recall Theorem 2 in Section 4, which states that if each state of $\mathcal{T}_{\mathcal{R}}$ in $I \in \Pi$ satisfies $p \in \Pi$ and moreover $\mathcal{R}$ is ground p-stable, then $\mathcal{R}$ is ground p-invariant from I.

The first key observation for an strengthening technique is the one made previously in Section 4: a topmost rewrite theory $\mathcal{R}$ may be ground p-invariant

from I and yet not be ground p-stable. As a matter of fact, for the ground p-invariance from I of $\mathcal{R}$, the only states from which p need not be falsified are precisely those $[t]_E$ reachable from an state in $[\![I]\!]$. The idea is then to strengthen p in the following way: if $\mathcal{R}$ is ground q-invariant from I and every state satisfying q also satisfies p (i.e., $[\![q]\!] \subseteq [\![p]\!]$), then clearly $\mathcal{R}$ is ground p-invariant from I, because any state in $\mathcal{T}_{\mathcal{R}}$ reachable from $[\![I]\!]$ is also in $[\![p]\!]$.

Theorem 4 gives an strengthening technique based on a generalization of the previous observation.

Theorem 4. *Let $\mathcal{R}$, Π, $\mathcal{E}_\Pi$, and p be as in Definition 2, and let $q, J \in \Pi$. If $\mathcal{R}$ is ground q-invariant from J and $[\![q]\!] \subseteq [\![p]\!]$, then $\mathcal{R}$ is ground p-invariant for any $I \in \Pi$ such that $[\![I]\!] \subseteq [\![J]\!]$.*

Proof. Let $I \in \Pi$ be such that $[\![I]\!] \subseteq [\![J]\!]$, and let $t, u \in T_{\Sigma,\mathfrak{s}}$ be such that $\mathcal{E}_\Pi \vdash I(t) = \top$ and $\mathcal{R} \vdash t \xrightarrow{*} u$; the goal is to prove $\mathcal{E}_\Pi \vdash p(u) = \top$. $[\![I]\!] \subseteq [\![J]\!]$ implies $\mathcal{E}_\Pi \vdash J(t) = \top$, and because $\mathcal{R}$ is ground q-invariant from J, $\mathcal{E}_\Pi \vdash q(u) = \top$ holds, and because $[\![q]\!] \subseteq [\![p]\!]$, then $\mathcal{E}_\Pi \vdash p(u) = \top$ follows. □

According to Theorem 4, in order to prove $\mathcal{R} \Vdash I \Rightarrow \Box p$ assuming $\mathcal{R} \Vdash J \Rightarrow \Box q$, it is sufficient to prove $[\![q]\!] \subseteq [\![p]\!]$ and $[\![I]\!] \subseteq [\![J]\!]$, for example, by means of Theorem 3. In LTL terms, Theorem 4 justifies the soundness of inference rule Str1 in Figure 4 to prove that p is an invariant from I in $\mathcal{K}^{\Pi}_{\mathcal{R}}$.

The second strengthening technique follows the next observation when $\mathcal{R}$ is to be proved ground p-invariant from I: if $\mathcal{R} \Vdash I \Rightarrow \Box q$, then any transition $[t]_E \rightarrow_{\mathcal{R}} [u]_E$ in $\mathcal{T}_{\mathcal{R}}$, with $[t]_E$ reachable from a state in $[\![I]\!]$, must satisfy $[t]_E \in [\![q]\!]$. More precisely, for any state $[t]_E$ reachable from $[\![I]\!]$ in $\mathcal{T}_{\mathcal{R}}$, the equivalence

$$[t]_E \rightarrow_{\mathcal{R}} [u]_E \quad \text{if and only if} \quad [t]_E \rightarrow_{\mathcal{R}} [u]_E \wedge [t]_E \in [\![q]\!]$$

is logically valid.

Before formally presenting this second strengthening technique for ground invariance, we introduce an alternative but equivalent notion of ground invariance.

Lemma 3. *Let $\mathcal{R}$, Π, $\mathcal{E}_\Pi$, p, I, and R_0 be as in Definition 2. Define $\mathcal{R}_0 = (\Sigma, E, R_0)$ and $\mathrm{reach}^{\mathcal{E}_\Pi}_{\mathcal{R}_0}(I) = \{[t]_E \in T_{\Sigma/E,\mathfrak{s}} \mid (\exists t_0 \in T_{\Sigma,\mathfrak{s}}) \mathcal{E}_\Pi \vdash I(t_0) = \top$ and $\mathcal{R}_0 \vdash t_0 \xrightarrow{*} t\}$. Then $\mathcal{R}$ is ground p-invariant from I under R_0 if and only if, $[\![I]\!] \subseteq [\![p]\!]$ and for each $t, u \in T_{\Sigma,\mathfrak{s}}$ such that $[t]_E \in \mathrm{reach}^{\mathcal{E}_\Pi}_{\mathcal{R}_0}(I)$, $\mathcal{E}_\Pi \vdash p(t) = \top$ and $\mathcal{R}_0 \vdash t \rightarrow u$ implies $\mathcal{E}_\Pi \vdash p(u) = \top$.*

Proof. ($\Rightarrow$) Let $t, u \in T_{\Sigma,\mathfrak{s}}$ be such that $[t]_E \in \mathrm{reach}^{\mathcal{E}_\Pi}_{\mathcal{R}_0}(I)$, and assume $\mathcal{E}_\Pi \vdash p(t) = \top$ and $\mathcal{R}_0 \vdash t \rightarrow u$. Hence $[u]_E \in \mathrm{reach}^{\mathcal{E}_\Pi}_{\mathcal{R}_0}(I)$ and since $\mathcal{R}_0$ is ground p-invariant from I, $\mathcal{E}_\Pi \vdash p(u) = \top$ follows. ($\Leftarrow$) Let $t, u \in T_{\Sigma,\mathfrak{s}}$ be such that $\mathcal{E}_\Pi \vdash I(t) = \top$ and $\mathcal{R}_0 \vdash t \xrightarrow{*} u$; the goal is to prove $\mathcal{E}_\Pi \vdash p(u) = \top$. By induction on the proof length n of $\mathcal{R}_0 \vdash t \xrightarrow{n} u$. If $n = 0$, then $t =_E u$ and hence $t =_{E \uplus E_\Pi} u$; since $\mathcal{E}_\Pi \vdash I(t) = \top$ and $[\![I]\!] \subseteq [\![p]\!]$, we have $\mathcal{E}_\Pi \vdash p(u) = \top$. If $\mathcal{R}_0 \vdash t \xrightarrow{n+1} u$, then there is $u_0 \in T_{\Sigma,\mathfrak{s}}$ such that $\mathcal{R}_0 \vdash t \rightarrow u_0 \wedge u_0 \xrightarrow{n} u$.

$\mathcal{E}_\Pi \vdash I(t) = \top$ implies $[t]_E \in \text{reach}(I)^{\mathcal{E}_\Pi}_{\mathcal{R}_0}$ and since $\mathcal{R}_0 \vdash t \to u_0$, we have $\mathcal{E}_\Pi \vdash p(u_0) = \top$ from the assumption. Moreover, $[u_0] \in \text{reach}^{\mathcal{E}_\Pi}_{\mathcal{R}_0}(I)$ and then by induction hypothesis $\mathcal{E}_\Pi \vdash p(u) = \top$. □

The second strengthening technique is contained in Theorem 5.

Theorem 5. *Let $\mathcal{R} = (\Sigma, E \uplus A, R)$ be a topmost rewrite theory with signature $\Omega \subseteq \Sigma$ of equational E-free constructors modulo A and with finitary A-unification algorithm, and let $\mathcal{E}_\Pi = (\Sigma_\Pi, E \uplus A \uplus E_\Pi)$ be an equational definition of Π for $\mathcal{R}$. Let $p \in \Pi$ and $(\forall Y)\ l \to r$ **if** $C \in R$. Without loss of generality, assume that the equations $E^p_\Pi \subseteq E_\Pi$ defining $p \in \Pi$ are all conditional, have no variables in Y, and have Ω-patterns as left-hand sides, and define*

$$\Theta_{(l,r,C)} = \bigcup_{((\forall Z)\ p(v)=w \text{ if } D) \in E^p_\Pi} \{(\theta, w, D) \mid \theta \in \text{CSU}_A(v = l)\}.$$

*Then, $\mathcal{R}$ is ground p-invariant from $I \in \Pi$ under $(\forall Y)\ l \to r$ **if** C if $\mathcal{R}$ is ground q-invariant from I, $[\![I]\!] \subseteq [\![p]\!]$, and for each $(\theta, w, D) \in \Theta_{(l,r,C)}$*

$$\mathcal{E}_\Pi \Vdash (\forall \text{ran}(\theta))\ C\theta \wedge D\theta \wedge w\theta = \top \wedge q(l\theta) = \top \ \Rightarrow\ p(r\theta) = \top.$$

Proof. Let $R_0 = \{(\forall Y)\ l \to r \text{ **if** } C\}$ and $\mathcal{R}_0 = (\Sigma, E \uplus A, R_0)$, and assume $\mathcal{R}$ is ground q-invariant from I and $[\![I]\!] \subseteq [\![p]\!]$:

$\mathcal{R}$ is ground p-invariant from I under R_0
iff { by Lemma 3 using $[\![I]\!] \subseteq [\![p]\!]$ }
$(\forall t, u \in T_{\Sigma,\mathfrak{s}})$
$[t]_{E \uplus A} \in \text{reach}^{\mathcal{E}_\Pi}_{\mathcal{R}_0}(I)$ implies
$(\mathcal{E}_\Pi \vdash p(t) = \top$ and $\mathcal{R}_0 \vdash t \to u$ implies $\mathcal{E}_\Pi \vdash p(u) = \top)$
iff { by definition of $\text{reach}^{\mathcal{E}_\Pi}_{\mathcal{R}_0}(I)$ }
$(\forall t, u, t_0 \in T_{\Sigma,\mathfrak{s}})$
$\mathcal{E}_\Pi \vdash I(t_0) = \top$ and $\mathcal{R}_0 \vdash t_0 \overset{*}{\to} t$ implies
$(\mathcal{E}_\Pi \vdash p(t) = \top$ and $\mathcal{R}_0 \vdash t \to u$ implies $\mathcal{E}_\Pi \vdash p(u) = \top)$
if { by $\mathcal{R}$ (thus $\mathcal{R}_0$) ground q-invariant from I }
$(\forall t, u \in T_{\Sigma,\mathfrak{s}})$
$\mathcal{E}_\Pi \vdash q(t) = \top$ implies
$(\mathcal{E}_\Pi \vdash p(t) = \top$ and $\mathcal{R}_0 \vdash t \to u$ implies $\mathcal{E}_\Pi \vdash p(u) = \top)$
iff { by definition of rewriting and by $\mathcal{E}_{\mathcal{R}_0} = \mathcal{E}_\mathcal{R}$ }
$(\forall \alpha : Y \longrightarrow T_\Sigma)$
$\mathcal{E}_\Pi \vdash q(l\alpha) = \top$ implies
$(\mathcal{E}_\Pi \vdash p(l\alpha) = \top$ and $\mathcal{E}_\mathcal{R} \vdash C\alpha$ implies $\mathcal{E}_\Pi \vdash p(r\alpha) = \top)$
iff { by $\mathcal{E}_\Pi$ protecting $\mathcal{E}_\mathcal{R}$ and $C\alpha$ a ground Σ-formula }
$(\forall \alpha : Y \longrightarrow T_\Sigma)$
$\mathcal{E}_\Pi \vdash (p(l\alpha) = \top \wedge q(l\alpha) = \top \wedge C\alpha) \Rightarrow p(r\alpha) = \top)$
iff { by proof of Theorem 1: from step 3 to step 7 }
$(\forall (p(v) = w \text{ **if** } D) \in E^p_\Pi)(\forall \theta \in \text{CSU}_A(l = v))(\forall \gamma : \text{ran}(\theta) \longrightarrow T_\Sigma)$
$(\mathcal{E}_\Pi \vdash (q(l\theta\gamma) = \top \wedge D\theta\gamma \wedge w\theta\gamma = \top \wedge C\theta\gamma) \Rightarrow p(r\theta\gamma) = \top$

iff { by definition of $\Vdash$ }
$(\forall (p(v) = w \textbf{ if } D) \in E_{\Pi}^{p})(\forall \theta \in \mathrm{CSU}_A(l = v))$
$\mathcal{E}_{\Pi} \Vdash (\forall \mathrm{ran}(\theta))(C\theta \wedge D\theta \wedge w\theta = \top \wedge q(l\theta) = \top) \Rightarrow p(r\theta) = \top$
iff { by definition of $\Theta_{(l,r,C)}$ }
$(\forall (\theta, w, D) \in \Theta_{(l,r,C)})$
$\mathcal{E}_{\Pi} \Vdash (\forall \mathrm{ran}(\theta))(C\theta \wedge D\theta \wedge w\theta = \top \wedge q(l\theta) = \top) \Rightarrow p(r\theta) = \top$

□

The strengthening technique for $p \in \Pi$ assuming ground q-invariance from I presented introduced in Theorem 5 is specially useful in situations in which $[\![q]\!] \not\subseteq [\![p]\!]$ or even in situations in which proving $[\![q]\!] \subseteq [\![p]\!]$ can be difficult.

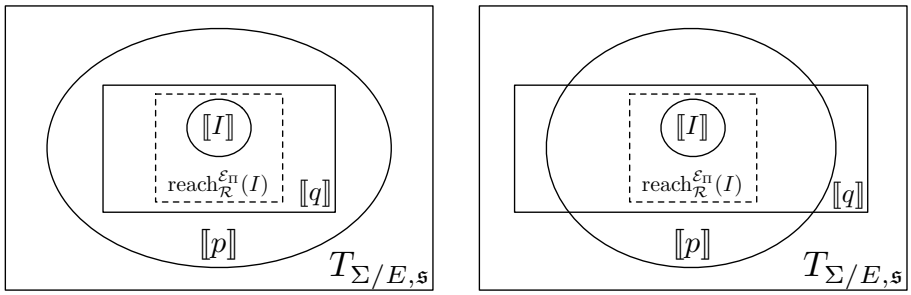

Fig. 3. $\mathcal{R} = (\Sigma, E, R)$ and $p, q, I \in \Pi$. Sets of states closed under $\rightarrow_{\mathcal{R}}$ in $\mathcal{T}_{\mathcal{R}}$ are depicted with rectangles. $\mathcal{R}$ is ground q-stable and ground q-invariant for I; $\mathcal{R}$ is ground p-invariant for I but not necessarily ground p-stable. (a) $[\![q]\!] \subseteq [\![p]\!]$. (b) $[\![q]\!] \not\subseteq [\![p]\!]$.

Figure 3 depicts situations (a) and (b) in which a topmost rewrite theory $\mathcal{R} = (\Sigma, E, R)$ is ground p-invariant and ground q-invariant for I, and also q-stable. In (a) both theorems 4 and 5 can be useful, but in (b) only Theorem 5 can be useful.

Theorem 5 justifies the soundness of inference rule STR2 in Figure 4, for reasoning in LTL, assuming q-invariance from I, about p-invariance from I of the Kripke structures $\mathcal{K}_{\mathcal{R}}^{\Pi}$.

Example 3. Recall examples 1 and 2 from sections 3 and 4, respectively. Here we are interested in state predicates $\Pi = \{\text{bounded-tickets}, \text{init}, \text{unique-tickets}, \text{good-state}, \text{mutex}\}$ for BAKERY, with bounded-tickets and init as defined in BAKERY-PROPS-EXT1. State predicate mutex defines a mutual exclusion property for BAKERY. State predicates unique-tickets and good-state are strengthenings for mutex. These predicates and the auxiliary function set, in, and = are equationally defined in module BAKERY-PROPS-EXT2, which extends BAKERY-PROPS-EXT1.

```
fmod BAKERY-PROPS-EXT2 is
 pr BAKERY-PROPS-EXT1 .
 ops unique-tickets good-state mutex : State -> [Bool] .
 op set : NatBag -> Bool .
 op in : Nat NatBag -> Bool .
 op = : Nat Nat -> Bool [comm] .
 var Ws : ProcWaitSet .  var Ps : ProcSet .
 vars N M M' N' : Nat .
 ...
 eq [3]   : unique-tickets(N : M [Ps]) = set(tkts(Ps)) .
 eq [4.1] : good-state(N : M [Ws]) = true .
 eq [4.2] : good-state(N : M [crit(M) Ws]) = true .
 eq [4.3] : good-state(N : M [crit(M') crit(N') Ps]) = false .
 eq [5.1] : mutex(N : M [Ws]) = true .
 eq [5.2] : mutex(N : M [crit(M') Ws]) = true .
 eq [5.3] : mutex(N : M [crit(M') crit(N') Ps]) = false .
endfm
```

The mutual exclusion property, completely defined by mutex for the sort Bool, holds in a state if and only if such state has at most one customer being served. State predicate good-state is a stronger version of mutex in which, for it to hold, the customer being served must have the appropriate ticket number. State predicate unique-tickets holds whenever the tickets among the customers are all distinct. Auxiliary predicates set, in, and = exactly hold, respectively, whenever a bag of natural numbers is a set, whenever a natural number belongs to a bag of natural numbers, and whenever two natural numbers are equal.

$$\frac{\mathcal{R} \Vdash J \Rightarrow \Box q \qquad \mathcal{R} \Vdash I \Rightarrow J \qquad \mathcal{R} \Vdash q \Rightarrow p}{\mathcal{R} \Vdash I \Rightarrow \Box p}\ \textsc{Str1}$$

$$\frac{\mathcal{R} \Vdash I \Rightarrow \Box q \qquad \mathcal{R} \Vdash I \Rightarrow p \qquad \mathcal{R} \Vdash q \wedge p \Rightarrow \bigcirc p}{\mathcal{R} \Vdash I \Rightarrow \Box p}\ \textsc{Str2}$$

$$\frac{\begin{array}{c}\textit{for each } ((\forall X)\ l \to r \textbf{ if } C) \in R \textit{ and } (\theta, w, D) \in \Theta_{(l,r,C)} : \\ \mathcal{E}_\Pi \Vdash (\forall \mathrm{ran}(\theta))\ (C\theta \wedge D\theta \wedge w\theta = \top \wedge q(l\theta) = \top) \Rightarrow p(r\theta) = \top\end{array}}{\mathcal{R} \Vdash q \wedge p \Rightarrow \bigcirc p}\ \textsc{Nr2}$$

Fig. 4. Checking $\mathcal{R} = (\Sigma, E, R)$ ground p-invariant under strengthenings (with $\Theta_{(l,r,C)}$ in rule NR2 as defined in Theorem 5).

The goal is to prove BAKERY ground mutex-invariant for init, which decomposes in three simpler goals. Namely, (i) to prove, by means of Theorem 5, BAKERY ground unique-tickets-invariant for init under the assumption of it being ground bounded-tickets-invariant for init (which was concluded in Ex-

ample 2), (ii) to prove, by means of Theorem 5, BAKERY ground good-state-invariant for init under the assumption of it being ground unique-tickets-invariant for init, and (iii) to prove, by means of Theorem 4, BAKERY ground mutex-invariant for init assuming it is ground good-state-invariant for init.

The proof obligations corresponding to (i), (ii), and (iii) amount to 30 in total. The following sentences are the proof obligations corresponding to (i):

$(\forall x_1, x_2 : \mathrm{Nat}; x_3 : \mathrm{ProcSet}; x_4 : \mathrm{ProcIdleSet})$

$$\mathrm{set}(\mathrm{tkts}(x_4)) = \top, \tag{8}$$

$$\mathrm{set}(x_2\ \mathrm{tkts}(x_3)) = \top\ \textbf{if}\ \mathrm{sb}(x_2\ \mathrm{tkts}(x_3), \mathrm{tb}(x_1)) = \top \wedge \mathrm{set}(x_2\ \mathrm{tkts}(x_3)) = \top, \tag{9}$$

$$\mathrm{set}(\mathrm{tkts}(x_3)) = \top\ \textbf{if}\ \mathrm{sb}(x_2\ \mathrm{tkts}(x_3), \mathrm{tb}(x_1)) = \top \wedge \mathrm{set}(x_2\ \mathrm{tkts}(x_3)) = \top, \tag{10}$$

$$\mathrm{set}(x_1\ \mathrm{tkts}(x_3)) = \top\ \textbf{if}\ \mathrm{sb}(\mathrm{tkts}(x_3), \mathrm{tb}(x_1)) = \top \wedge \mathrm{set}(\mathrm{tkts}(x_3)) = \top, \tag{11}$$

Sentence (8) trivially follows because $\mathrm{tkts}(x_4) = \mathrm{mtset}$ since x_4 has sort ProcIdleSet, and Sentence (9) follows because the consequent is a conjunct in the antecedent. Sentences 11 follows by a simple narrowing argument and 11 requires structural induction on the sort Nat. Section 6 gives an account of all proof obligations for (i), (ii), and (iii).

Observe that using Theorem 4 for proving (i), i.e., for proving BAKERY $\Vdash$ init $\Rightarrow \Box$good-state assuming BAKERY $\Vdash$ init $\Rightarrow \Box$unique-tickets would give the following proof obligation:
$(\forall x_1, x_2 : \mathrm{Nat}; x_3 : \mathrm{ProcSet})$

$$\text{good-state}(x_1 : x_2\ [x_3]) = \top\ \textbf{if}\ \text{unique-tickets}(x_1 : x_2\ [x_3]) = \top. \tag{12}$$

For proving Sentence 12, one could try structural induction on the complexity of x_3, but in this particular case such a proof would probably be involved because of the relationship between the counters x_1 and x_2, and the set of processes represented by x_3.

Observe also that BAKERY is mutex-invariant for init but it is not ground mutex-stable. For example, for state terms $t =$ "2 : 1 [wait(1) crit(1)]" and $u =$ "2 : 1 [crit(1) crit(1)]" we have BAKERY-PROPS-EXT2 $\vdash \mathrm{mutex}(t) = \top$ and also BAKERY $\vdash t \rightarrow u$ but BAKERY-PROPS-EXT2 $\vdash \mathrm{mutex}(u) = \bot$. The observation here is that the state represented by t is not BAKERY-reachable from any initial state specified by init.

6 Maude's Invariant Analyzer

Our approach for proving ground stability and ground invariance of rewrite theories, depicted in Figure 5, adds tool support to a part currently missing in the Maude environment: theorem proving for safety properties of infinite-state concurrent systems specified by topmost rewrite theories.

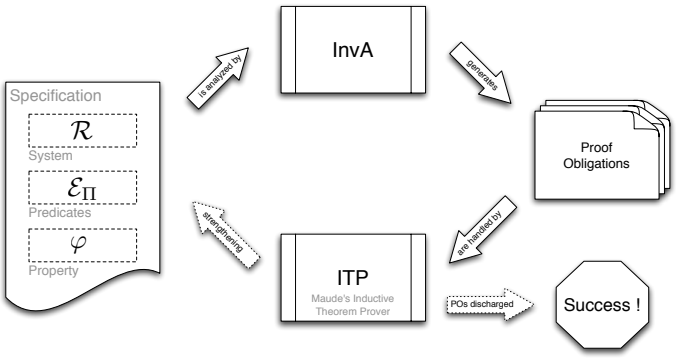

Fig. 5. Approach for checking ground invariance and ground stability of rewrite theories.

For a topmost rewrite theory $\mathcal{R}$ and of a set of state predicates Π in the language of Maude, the InvA tool mechanizes inference rules G-St, G-Inv, Str1, Str2, Nr1, and Nr2. Given a ground stability or ground invariance property φ, it generates equational proof obligations such that if they hold, then $\mathcal{T}_\mathcal{R} \models \varphi$. It also has support for properties of the form $q \Rightarrow p$. Thanks to the availability in Maude 2.6 of unification modulo commutativity (C), associativity and commutativity (AC), and modulo these theories plus identities (U), and to the narrowing modulo infrastructure available in Full Maude 2.6, InvA can handle modules with operators declared C, CU, AC, and ACU.

Automatic discharge of proof obligations. After applying rules G-St, G-Inv, Str1, Str2, Nr1, and Nr2 according to the user commands, the InvA tool uses rewriting-based reasoning and narrowing procedures for automatically discharging as many of the generated equational proof obligations as possible. For $\mathcal{E} = (\Sigma, E \uplus A)$ and a conditional proof obligation $\varphi = (\forall X)\ t = u$ **if** C, the InvA tool applies a proof-search strategy such that if it succeeds, then $\mathcal{T}_\mathcal{E} \models \varphi$. Otherwise, the proof obligation is output to the user. Let $\overline{t}$, $\overline{u}$, $\overline{C}$ be obtained by replacing each variable $x \in X$ by a new constant $\overline{x} \in \overline{X}$, with $\Sigma \cap \overline{X} = \emptyset$. First, the strategy checks if φ holds *trivially*, i.e., if $\mathrm{can}_{\Sigma,E/A}(t) =_A \mathrm{can}_{\Sigma,E/A}(u)$ or there is $t_i = u_i$ in C such that $\mathrm{can}_{\Sigma,E/A}(t_i), \mathrm{can}_{\Sigma,E/A}(u_i) \in T_\Sigma$ but $\mathrm{can}_{\Sigma,E/A}(t_i) \neq_A \mathrm{can}_{\Sigma,E/A}(u_i)$. Second, it checks if φ is *context-joinable* [5]: φ is context-joinable if $\overline{t}$ and $\overline{u}$ are joinable in the rewrite theory $\mathcal{R}^\varphi_\mathcal{E} = (\Sigma(\overline{X}), A, \overrightarrow{E} \cup \overrightarrow{C})$, obtained from orienting equations E as rewrite rules $\overrightarrow{E}$ and *heuristically* orienting each equality $t_i = u_i$ in C as a sequent $\overline{t_i} \to \overline{u_i}$ in $\overrightarrow{C}$. Third, it checks if the proof obligation is *unfeasible* [5]: φ is unfeasible if there is a conjunct $\overline{t_i} \to \overline{u_i}$ in $\overrightarrow{C}$ and $v, w \in T_\Sigma(X)$ such that $\mathcal{R}^\varphi_\mathcal{E} \vdash \overline{t_i} \to \overline{v} \wedge \overline{t_i} \to \overline{w}$, $\mathrm{CSU}_A(v = w) = \emptyset$, and v and w are strongly irreducible with $\overrightarrow{E}$ modulo A. Because of the executability assumptions on $(\Sigma, E \uplus A)$, the first test of the strategy either succeeds or

fails in finitely many equational rewrite steps. For the second and third tests, the strategy is not guaranteed to succeed or fail in finitely many rewrite steps because the oriented sequents $\overrightarrow{C}$ can falsify the termination assumption. So, for these last two checks, InvA uses a bound for the depth of the proof-search.

Commands in InvA**.** The commands available in the InvA tool are the following:

- **(help .)** shows the list of commands available in the tool.
- **(analyze-stable <pred> in <eqmodule> <rmodule> .)** generates the proof obligations for proving the premise of inference G-St with inference Nr1, for the given predicate and the given modules. The first module equationally specifies the state predicate and the second one the topmost rewrite theory. This command tries to eagerly discharge the proof obligations; those that cannot be discharged are shown to the user.
- **(analyze-stable <pred> in <eqmodule> <rmodule> assuming <pred> .)** generates the proof obligations for proving the third premise of inference Str2 with inference Nr2, for the given predicate and the given modules. The first module equationally specifies the state predicates and the second one the topmost rewrite theory. This command tries to eagerly discharge the proof obligations and those that cannot be discharged are shown to the user.
- **(analyze <pred> implies <pred> in <eqmodule> .)** generates the proof obligations for proving the given implication in the given module, according to inference C⇒. This command tries to eagerly discharge the proof obligations; those that cannot be discharged are shown to the user.
- **(show pos .)** shows the proof obligations computed in the last `analyze...` command that could not be discharged; those that were discharged are not shown.
- **(show-all pos .)** shows the proof obligations computed in the last `analyze...` command.

Observe that the analysis commands in InvA give direct tool support for deductive reasoning with *some* of the inference rules presented in this paper, but not for all of them. For example, there is no command in InvA directly supporting deduction with inference rule G-Inv. Nevertheless, deduction with *all* inference rules in this paper is supported by InvA via *combination of commands*. For example, deduction with inference rule G-Inv can be achieved by combining the `analyze` and `analyze-stable` commands.

Tool interaction. Once InvA has been started, and a rewrite theory module $\mathcal{R}$ and an equational theory module $\mathcal{E}_{\Pi}$ has been entered, we can check whether $\mathcal{R}$ is ground stable or ground invariant for some predicates in Π. The following snapshot shows an interaction with InvA in which BAKERY $\Vdash$ init $\Rightarrow \Box$mutex is proved assuming BAKERY $\Vdash$ init $\Rightarrow \Box$good-state:

```
Checking BAKERY-PROPS ||- init => good-state ...
```

```
Proof obligations generated:  1
Proof obligations discharged: 1
rewrites: 4241 in 16ms cpu (18ms real) (253345 rewrites/second)
Success!

Checking BAKERY U BAKERY-PROPS ||- good-state => O good-state under
    strengthening unique-tickets ...
Proof obligations generated:  19
Proof obligations discharged: 19
rewrites: 26879 in 70ms cpu (73ms real) (381019 rewrites/second)
Success!

Checking BAKERY-PROPS ||- good-state => mutex ...
Proof obligations generated:  3
Proof obligations discharged: 3
rewrites: 9121 in 15ms cpu (18ms real) (592580 rewrites/second)
Success!
```

A complete proof summary of the ground mutex-invariance from init for BAKERY is depicted in Table 1. The InvA tool generates 37 proof obligations in total, of which it automatically discharges 34 of them in less than 200 milliseconds. The remaining 4 proof obligations can be discharged in Maude's ITP by structural induction on the sort Nat with the help of some lemmas.

Property	InvA/Time	ITP	Total
(a) init ⇒ □bounded-tickets	6/35ms	1	7
(b) init ⇒ □unique-tickets assuming (a)	5/41ms	2	7
(c) init ⇒ □good-state assuming (b)	20/91ms	-	20
(d) init ⇒ □mutex assuming (c)	3/18ms	-	3
Total	34/194ms	3	37

Table 1. Proof summary of the ground mutex-invariance from init for BAKERY in InvA. Column InvA/Time indicates the number of proof obligations automatically discharged by the InvA and the time in milliseconds, column ITP indicates the number of proof obligations requiring user interaction, and column Total indicates the total number of proof obligations for a given property.

7 Related Work and Concluding Remarks

Chandy and Misra [2] and Manna and Pnueli [11] pursued the idea of using a deductive methodology to prove the invariance properties of concurrent systems specified in imperative languages. The notion of stability was inspired by the definition of the *stable* predicate in [16]. A comprehensive account of the vast literature on deductive approaches for verifying invariants of concurrent systems is beyond the scope of the present work; the aim here is more modest, namely, we focus on related work using rewriting techniques in the deductive verification of invariants.

Rusu [18] proposes an approach for verifying invariant properties of a (possibly infinite-state) concurrent system specified by an unconditional topmost rewrite theory, following the ideas of Bruni and Meseguer [1]. His approach consists in casting an invariance problem of the form $\mathcal{R} \Vdash I \Rightarrow \Box p$ as an inductive problem of an equational theory $\mathcal{M}(\mathcal{R}, I)$ in membership equational logic, an equational sublogic of rewriting logic, as follows: $\mathcal{R} \Vdash I \Rightarrow \Box p$ if and only if $\mathcal{M}(\mathcal{R}, I) \Vdash p(t) = \top$ for every ground term t of sort *Reachable*, and t has sort *Reachable* in $\mathcal{M}(\mathcal{R}, I)$ if and only if t is $\mathcal{R}$-reachable from I. The approach in [18] is complemented by bounded symbolic execution, achieved by narrowing modulo, so that a property can be symbolically tested before trying to prove it invariant. The key difference between this approach and ours is that the proof obligations generated for proving $\mathcal{M}(\mathcal{R}, I) \Vdash p(t) = \top$ do not take advantage of p's equational definition, in contrast to our approach in which theorems 1 and 5 are very useful for simplifying the user's interactive theorem proving burden. As a matter of fact, a version of the bakery protocol for unbounded number of processes similar to the one presented in this paper, is proved invariant for the mutual exclusion property in [19] following the approach in [18]: it requires more than 50 lemmas of which some demand non-trivial user interaction. Our approach can benefit from using narrowing for the symbolic testing of state predicates, although more research is required for handling conditional rewrite theories.

Proof scores in the OTS/CafeOBJ method are used to prove invariant properties of concurrent systems specified by *observational transition systems* [17]. This approach has been applied for verifying safety properties of large specifications, including communication protocols. The approach is to divide a formula stating an invariant property into reasonably smaller ones by exploiting properties of the Boolean operators, each of which is proved by writing proof scores (or proof obligations) to be discharged individually by equational rewriting. The main difference between this approach and ours is that proof scores are constructed and manipulated manually by the user, resulting in time in a verification process. The interesting idea of exploiting the properties of Boolean operators needs to be further studied and considered within our approach, as well the development of more challenging case studies.

Combinations of deductive and algorithmic techniques have also been proposed for proving temporal logic properties φ of a (possibly infinite-state) concurrent system specified by a rewrite theory $\mathcal{R} = (\Sigma, E, R)$. Equational abstractions [14] reduces the problem of whether $\mathcal{R}$ satisfies φ to model checking ϕ on

a finite state abstract version $\mathcal{R}/\Delta = (\Sigma, E \uplus E_\Delta, R \uplus R_\Delta)$. Invisible transitions [7] approach the problem of whether $\mathcal{R}$ satisfies φ by identifying a subset $S \subseteq R$ of rewrite rules that are φ-invisible (i.e., rewriting with S does not change the truth value of the predicate φ) to model checking that property on a finite state simplified version $\mathcal{R}/S = (\Sigma, E \uplus S, R \setminus S)$. Both equational abstractions and invisible transition techniques tackle the verification problem of infinite-state systems by making finite the state space explosion so that model checking methods are decidable. These two approaches, as it is also the case in our approach, require user-intervention for defining, respectively, the abstraction predicates and the invisible rewrite rules, and for discharging the inductive proof obligations resulting from the corresponding transformations (i.e., executability conditions plus the proof obligations specific to each method). In particular, the checking algorithms based on narrowing presented in this paper can be used to generate proof obligations for checking the rewrite rules $S \subseteq R$ of $\mathcal{R}$, p-invisible for a state predicate p, as S is p-invisible if and only if $\mathcal{R}$ is ground p-stable under S. We believe that these approaches complement each other and can be combined with our approach, resulting in a powerful and versatile framework for proving temporal properties of rewrite theories. The mechanization of these three approaches in order to reduce user intervention is an open question for further investigation.

Narrowing-based symbolic model checking techniques for topmost rewrite theories $\mathcal{R}$ have been previously studied in [6], where the idea is to "fold" the narrowing tree for $\mathcal{R}$ that can in practice result in finite-state system that symbolically simulates $\mathcal{R}$. It is worth pursuing an extension of these narrowing symbolic model checking techniques for conditional rewrite theories, so they can be combined with our approach for symbolic model checking and for symbolic simulation (following the idea of Rusu in [18]).

We have presented both a deductive methodology and a framework for proving ground stability and ground invariance of a (possibly infinite-state) concurrent system specified by conditional topmost rewrite theories. The proof obligations of the verification task are equational Horn clauses, into which the ones related to the dynamics of the concurrent system are reduced by the inference rules and the 1-step ground narrowing procedure.

Much work remains ahead. First of all, all the results presented here have a straightforward generalization to *state predicates with parameters*; that is, instead of state predicates of the form $p(s)$ with s a state, it is often very convenient to use state predicates of the form $p(s, d_1, \ldots, d_n)$, with s a state, and the $d_1, \ldots, d_n$ data parameters. All the ideas presented here can be extend to deal with predicates with data parameters. Also, a wider range of case studies stressing the tool's capabilities should be developed. Among such case studies, the application InvA as a generic tool to the verification of programs in *specific programming languages* should be given high priority, since this will demonstrate the wide applicability of the reasoning methods presented here. Fortunately, thanks to recent advances in the rewriting logic semantics project [15], there is already a wealth of language specifications available, to which the generic model

checking verification approach using the Maude LTL model checker has already been successfully applied.

More ambitiously, the transformational approach to safety property verification presented here should be extended to a wider set of of LTL formulas, including formulas stating liveness properties. A tighter integration between a more general tool for deductive verification of LTL properties and Maude's LTL model checker is yet another longer-range goal, since this will allow the seamless combination of deductive and algorithmic methods within an overall verification effort, and will exploit the genericity advantages of both methods in the rewriting logic framework.

References

1. R. Bruni and J. Meseguer. Semantic foundations for generalized rewrite theories. *Theoretical Computer Science*, 360(1-3):386–414, 2006.
2. K. M. Chandy and J. Misra. *Parallel Program Design, A foundation.* Addison Wesley 1988, 1988.
3. E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking.* The MIT Press, Cambridge, Massachusetts, 1999.
4. M. Clavel, F. Durán, S. Eker, J. Meseguer, P. Lincoln, N. Martí-Oliet, and C. Talcott. *All About Maude - A High-Performance Logical Framework.* Springer LNCS Vol. 4350, 1st edition, 2007.
5. F. Durán and J. Meseguer. A church-rosser checker tool for conditional order-sorted equational maude specifications. In P. C. Ölveczky, editor, *WRLA*, volume 6381 of *Lecture Notes in Computer Science*, pages 69–85. Springer, 2010.
6. S. Escobar and J. Meseguer. Symbolic model checking of infinite-state systems using narrowing. *Lecture Notes in Computer Science*, 4533:153–168, 2007.
7. A. Farzan and J. Meseguer. State space reduction of rewrite theories using invisible transitions. *Lecture Notes in Computer Science*, 4019:142–157, 2006.
8. J. Hendrix. *Decision Procedures for Equationally Based Reasoning.* PhD thesis, University of Illinois at Urbana-Champaign, April 2008.
9. J.-P. Jouannaud, C. Kirchner, and H. Kirchner. Incremental construction of unification algorithms in equational theories. *Lecture Notes in Computer Science*, pages 361–373, 1983.
10. Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems.* Springer-Verlag, New York, 1992.
11. Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems.* Springer-Verlag, New York, 1995.
12. J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, 1992.
13. J. Meseguer. Membership algebra as a logical framework for equational specification. *Lecture Notes in Computer Science*, 1376:18–61, 1997.
14. J. Meseguer, M. Palomino, and N. Martí-Oliet. Equational abstractions. *Theoretical Computer Science*, 403(2-3):239–264, 2008.
15. J. Meseguer and G. Rosu. The rewriting logic semantics project. *Theoretical Computer Science*, 373(3):213–237, 2007.
16. J. Misra. *A Discipline of Multiprogramming: Programming Theory for Distributed Aplications.* Monographs in Computer Science. Springer-Verlag, New York, 2001.
17. K. Ogata and K. Futatsugi. Proof scores in the ots/cafeobj method. *Lecture Notes in Computer Science*, 2884:170–184, 2003.
18. V. Rusu. Combining theorem proving and narrowing for rewriting-logic specifications. *Lecture Notes in Computer Science*, 6143:135–150, 2010.
19. V. Rusu and M. Clavel. Vérification d'invariants pour des systèmes spécifiés en logique de réécriture. *Vingtièmes Journées Francophones des Langages Applicatifs*, 7.2:317–350, 2009.
20. P. Viry. Equational rules for rewriting logic. *Theoretical Computer Science*, 285:487–517, 2002.