# Infinite-State Model Checking of LTLR Formulas Using Narrowing

Kyungmin Bae(✉) and José Meseguer

Department of Computer Science, University of Illinois at Urbana-Champaign,
Urbana, IL 61801, USA
{kbae4,meseguer}@cs.uiuc.edu

**Abstract.** The linear temporal logic of rewriting (LTLR) is a simple extension of LTL that adds *spatial action patterns* to the logic, expressing that a specific instance of an action described by a rewrite rule has been performed. Although the theory and algorithms of LTLR for finite-state model checking are well-developed [2], no theoretical foundations have yet been developed for infinite-state LTLR model checking. The main goal of this paper is to develop such foundations for *narrowing-based logical model checking* of LTLR properties. A key theme in this paper is the systematic relationship, in the form of a simulation with remarkably good properties, between the concrete state space and the symbolic state space. A related theme is the use of additional state space reduction methods, such as folding and equational abstractions, that can in some cases yield a finite symbolic state space.

**Keywords:** Model checking · Infinite-state systems · LTLR · Narrowing

## 1 Introduction

This paper further develops previous efforts to use rewriting logic and narrowing to perform symbolic model checking of infinite-state systems.[1] Those efforts have gradually increased the expressiveness of the properties that can be verified, first focusing on reachability analysis [16] and then expanding the range to general LTL formulas [1,6]. It is by now clear that state-based temporal logics are not expressive enough to deal with properties involving events, such as message sends and receives; and that the temporal logic of rewriting [14] is a perfect match—at the level of property specification—for rewriting logic—at the level of system specification—so that both can be used seamlessly as a tandem for model checking. For finite-state systems, the authors have developed model checkers that demonstrate the power and usefulness of this tandem of logics [2]. The question asked and positively answered in this paper is: can properties of a rewrite theory $\mathcal{R}$ expressed in the *linear temporal logic of rewriting* (LTLR) [14] be model checked symbolically by narrowing under reasonable assumptions?

[1] The temporal logics that can be verified by infinite-state model checking techniques are generally less expressive than those supported by finite-state model checkers.

S. Escobar (Ed.): WRLA 2014, LNCS 8663, pp. 113–129, 2014.
DOI: 10.1007/978-3-319-12904-4_6

The answer to this question is nontrivial, because of a difficulty which can be best explained by briefly recalling how narrowing-based reachability analysis and LTL model checking are performed for a rewrite theory $\mathcal{R}$. For reachability analysis, *any* non-variable term $t$, symbolically denoting a typically infinite set of concrete state instances, can be narrowed to try to reach an instance of a goal pattern term $g$. However, for LTL model checking, *not all such terms* $t$ denote states in the symbolic state space. The reason is that LTL formulas have a set $AP$ of state propositions, but for a symbolic term $t$ such propositions may not be defined: different term instances of $t$ may satisfy different state propositions. The solution proposed in [1,6] is to *specialize* $t$ to most general instances $t_1, \ldots, t_n$ for which all state propositions in $AP$ are either true or false. If the equations defining such propositions have the finite variant property, this can be done by variant narrowing [1,6]. Therefore, narrowing-based LTL model checking symbolically explores the state space of all such $AP$*-instantiated symbolic terms*.

Suppose that we now want to perform not just LTL model checking but symbolic LTLR model checking, and that our formula $\varphi$ involves both state propositions in $AP$ and spatial action patterns. For example, a spatial action pattern $l(\theta)$ can appear in $\varphi$, stating that a rule $l : q \longrightarrow r$ has been performed with an instantiation that further specializes the substitution $\theta$. As part of the model checking verification of $\varphi$ we may reach a symbolic state $t$ where we need to check whether the action specified by $l(\theta)$ can be performed. This check will succeed if $t$ can be narrowed with a rule $l$ and a substitution $\sigma$ such that $\theta$ is an *instance* of $\sigma$. However, $\sigma$ can be *incomparable* to $\theta$ in general; that is, $\sigma$ may have instances for which this property holds, and other instances for which it *definitely fails*. This is analogous to the lack of $AP$-instantiation discussed above for narrowing-based LTL model checking. Let $ACT$ be the set of spatial action patterns we are using, so that, say, $l(\theta) \in ACT$. Our problem is that the symbolic transitions in the LTLR state space need to be $ACT$*-instantiated*, while the symbolic states are $AP$*-instantiated*.

Lack of $ACT$-instantiations is a subtler problem than lack of $AP$-instantiation. After all, state propositions in $AP$ are equationally defined as Boolean predicates *in both their positive and negative cases*, so that variant narrowing can automate $AP$-instantiation. The problem of $ACT$-instantiation has to do with effectively characterizing the *negative cases* in which an action pattern does *not* hold. This turns out to be closely related to the problem of computing *complement patterns* of a pattern term; e.g., for a pattern $l(\theta)$, terms $u_1, \ldots, u_k$ such that any ground term is an instance of *exactly one term* in the set

$$\{l(\theta), u_1, \ldots, u_k\}.$$

Not all terms have such complements. For example, for an unsorted signature with constant 0, unary operator $s$, and free binary operator $f$, the term $f(x, x)$ has *no* such complements. However, effective methods have been developed to check when a term $t$ has complements and to compute them (for example, [8,9,12]). Under appropriate assumptions, they can provide a method to solve the $ACT$-instantiation problem.

Having identified conditions under which the state space for narrowing-based LTRL model checking can be built, the rest of the paper develops the theoretical foundations of narrowing-based LTLR model checking. A key theme in such foundations is the systematic relationship between concrete and symbolic states. This takes the form of a simulation relation from concrete to symbolic states that preserves both state propositions and spatial action patterns. A related theme is the use of additional state space reduction methods, such as folding and equational abstractions, that can in some cases yield a finite symbolic state space. How these foundations can be used in practice to prove nontrivial LTLR properties of infinite-state systems is illustrated with a running example.

## 2 Preliminaries

**Rewriting Logic.** An order-sorted signature is a triple $\Sigma = (S, \leq, \Sigma)$ with poset of sorts $(S, \leq)$ and operators $\Sigma = \{\Sigma_{w,k}\}_{(w,k) \in S^* \times S}$ typed in $(S, \leq)$. The set $\mathcal{T}_\Sigma(\mathcal{X})_\mathsf{s}$ denotes the set of $\Sigma$-terms of sort $\mathsf{s}$ over $\mathcal{X}$ an infinite set of $S$-sorted variables, and $\mathcal{T}_{\Sigma,\mathsf{s}}$ denotes the set of ground $\Sigma$-terms of sort $\mathsf{s}$. We assume that $\mathcal{T}_{\Sigma,\mathsf{s}} \neq \emptyset$ for each sort $\mathsf{s}$ in $\Sigma$. *Positions* in a term $t$ represent tree positions when $t$ is parsed as a tree, and the replacement in $t$ of a subterm at a position $p$ by another term $u$ is denoted by $t[u]_p$. A *substitution* $\sigma : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X})$ is a function that maps variables to terms of the same sort, and is homomorphically extended to $\mathcal{T}_\Sigma(\mathcal{X})$ in a natural way. The *domain* of $\sigma$ is a finite subset $dom(\sigma) \subseteq \mathcal{X}$, where $\sigma x = x$ for any $x \notin dom(\sigma)$. The restriction of $\sigma$ to $Y \subseteq \mathcal{X}$ is the substitution $\sigma|_Y$ such that $\sigma|_Y(x) = \sigma(x)$ if $x \in Y$, and $\sigma|_Y(x) = x$ otherwise.

A rewrite theory is a formal specification of a concurrent system [13]. To apply narrowing-based methods, we consider *unconditional order-sorted rewrite theories* $\mathcal{R} = (\Sigma, E, R)$, where: (i) $(\Sigma, E)$ is an equational theory with $\Sigma$ an order-sorted signature and $E$ a set of equations, specifying the system's states as the initial algebra $\mathcal{T}_{\Sigma/E}$ (i.e., each state is an $E$-equivalence class $[t]_E \in \mathcal{T}_{\Sigma/E}$ of ground terms); and $R$ is a set of unconditional *rewrite rules* of the form $l : q \longrightarrow r$ with label $l$ and $\Sigma$-terms $q, r \in \mathcal{T}_\Sigma(\mathcal{X})_\mathsf{s}$, specifying the system's transitions as a *one-step rewrite*

$$t[l(\theta)]_p : [t[\theta q]_p]_E \longrightarrow_\mathcal{R} [t[\theta r]_p]_E$$

from a state $[t[\theta q]_p]_E \in \mathcal{T}_{\Sigma/E}$ containing a substitution instance $\theta q$ of $q$ to the corresponding state $[t[\theta r]_p]_E \in \mathcal{T}_{\Sigma/E}$ in which $\theta q$ has been replaced by $\theta r$, where $t[l(\theta)]_p$ is called a *one-step proof term.*

We also require $\mathcal{R} = (\Sigma, E, R)$ being *topmost* for narrowing-based methods. That is, there is sort $\mathsf{State}$ at the top of one of the connected component of $(S, \leq)$ such that: (i) for each rule $l : q \longrightarrow r \in R$, both $q$ and $r$ have the top sort $\mathsf{State}$; and no operator in $\Sigma$ has $\mathsf{State}$ or any of its subsorts as an argument sort. This ensures that all rewrites with rules in $R$ must take place at the top of the term. In practice, many concurrent systems, including object-oriented systems and communication protocols, can be specified by topmost rewrite theories [16].

We can associate to $\mathcal{R}$ a corresponding Kripke structure for LTL model checking. A *Kripke structure* is a 4-tuple $\mathcal{K} = (S, AP, \mathcal{L}, \longrightarrow_{\mathcal{K}})$ with $S$ a set of *states*, $AP$ a set of atomic *state propositions*, $\mathcal{L} : S \to \mathcal{P}(AP)$ a *state-labeling function*, and $\longrightarrow_{\mathcal{K}} \subseteq S \times S$ a *total transition relation* in which every state $s \in S$ has a next state $s' \in S$ with $s \longrightarrow_{\mathcal{K}} s'$. A state proposition is defined as a term of sort $\mathsf{Prop}$, whose meaning is defined by equations using the auxiliary operator $\_ \models \_ : \mathsf{State}\ \mathsf{Prop} \to \mathsf{Bool}$. By definition, $p \in \mathcal{T}_{\Sigma/E,\mathsf{Prop}}$ is satisfied on a state $[t]_E$ iff $(t \models p) =_E \mathit{true}$. We assume that sort $\mathsf{Bool}$ has two constants $\mathit{true}$ and $\mathit{false}$ with $\mathit{true} \neq_E \mathit{false}$ and any $t \in \mathcal{T}_{\Sigma,\mathsf{Bool}}$ is provably equal to either $\mathit{true}$ or $\mathit{false}$.

**Definition 1.** *Given $\mathcal{R} = (\Sigma, E, R)$ and a set $AP \subseteq \mathcal{T}_{\Sigma/E,\mathsf{Prop}}$ defined by $E$, the corresponding Kripke structure is $\mathcal{K}(\mathcal{R})_{AP} = (\mathcal{T}_{\Sigma/E,\mathsf{State}}, AP, \mathcal{L}_E, \longrightarrow_{\mathcal{R}})$,*[2] *where $\mathcal{L}_E([t]_E) = \{p \in AP \mid (t \models p) =_E \mathit{true}\}$.*

**Linear Temporal Logic of Rewriting.** The *linear temporal logic of rewriting* (LTLR) is a state/event extension of LTL with *spatial action patterns* [2]. An LTLR formula $\varphi$ may include spatial action patterns $\delta_1, \ldots, \delta_n$ as well as state propositions $p_1, \ldots, p_m$, and therefore may describe properties involving both states and events. Given a set of state propositions $AP$ and a set of spatial action patterns $ACT$, the syntax of LTLR is defined by

$$\varphi ::= p \mid \delta \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc\varphi \mid \varphi\, \mathbf{U}\, \varphi,$$

where $p \in AP$ and $\delta \in ACT$. Other operators can be defined by equivalences, e.g., $\Diamond\varphi \equiv \mathit{true}\, \mathbf{U}\, \varphi$ and $\Box\varphi \equiv \neg\Diamond\neg\varphi$.

*Spatial action patterns* describe properties of one-step rewrites by defining a set of matching one-step proof terms. For example, a pattern $l$ describes that a rule with label $l$ is applied, and a pattern $l(\theta)$ describes that a rule with label $l$ is applied and the related variable instantiation is a further instantiation of the substitution $\theta$ [2,14]. In a similar way that state propositions of LTL are defined by equations, the matching relation $\models$ between a one-step proof term $\gamma$ and a spatial action pattern $\delta$ can be defined by equations using the auxiliary operator $\_ \models \_ : \mathsf{ProofTerm}\ \mathsf{Action} \to \mathsf{Bool}$, where $\gamma \models \delta \iff (\gamma \models \delta) =_E \mathit{true}$.

The semantics of an LTLR formula is defined on a *labeled Kripke structure* (LKS), an extension of a Kripke structure with transition labels [2,3]. An LKS is a 5-tuple $\bar{\mathcal{K}} = (S, AP, \mathcal{L}, ACT, \longrightarrow_{\bar{\mathcal{K}}})$ with $S$ a set of *states*, $AP$ a set of *state propositions*, $\mathcal{L} : S \to \mathcal{P}(AP)$ a *state-labeling function*, $ACT$ a set of *spatial action patterns*, and $\longrightarrow_{\bar{\mathcal{K}}} \subseteq S \times \mathcal{P}(ACT) \times S$ a total *labeled transition relation*. A *path* $(\pi, \alpha)$ is a pair of functions $\pi : \mathbb{N} \to S$ and $\alpha : \mathbb{N} \to \mathcal{P}(ACT)$ such that $\pi(i) \xrightarrow{\alpha(i)}_{\bar{\mathcal{K}}} \pi(i+1)$, and $(\pi, \alpha)^k$ denotes the suffix of $(\pi, \alpha)$ beginning at position $k$ such that $(\pi, \alpha)^k = (\pi \circ s^k, \alpha \circ s^k)$ with $s$ the successor function.

We can associate to a rewrite theory $\mathcal{R}$ a corresponding LKS $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$ for LTLR model checking, provided that the state propositions $AP$ and the spatial action patterns $ACT$ are defined by its equations.

[2] Since $\longrightarrow_{\mathcal{R}}$ needs to be total, we also assume that $\mathcal{R}$ is deadlock-free. Note that $\mathcal{R}$ can be easily transformed into an equivalent deadlock-free theory [15].

**Definition 2.** *Given a rewrite theory $\mathcal{R} = (\Sigma, E, R)$, sets $AP \subseteq \mathcal{T}_{\Sigma/E,\mathsf{Prop}}$ and $ACT \subseteq \mathcal{T}_{\Sigma/E,\mathsf{Action}}$ defined by $E$, the corresponding LKS is*

$$\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT} = (\mathcal{T}_{\Sigma/E,\mathsf{State}}, AP, \mathcal{L}_E, ACT, \longrightarrow_{\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}}),$$

*where $\mathcal{L}_E([t]_E) = \{p \in AP \mid (t \models p) =_E true\}$, and $[t]_E \xrightarrow{A}_{\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}} [t']_E$ iff $\gamma : [t]_E \longrightarrow_{\mathcal{R}} [t']_E$ and $A = \{\delta \in ACT \mid (\gamma \models \delta) =_E true\}$.*

Given an LTLR formula $\varphi$ and an initial state $s_0 \in S$, the satisfaction relation $\bar{\mathcal{K}}, s_0 \models \varphi$ holds iff for each path $(\pi, \alpha)$ of $\bar{\mathcal{K}}$ beginning at $s_0$, the path satisfaction relation $\bar{\mathcal{K}}, (\pi, \alpha) \models \varphi$ holds, which is defined inductively as follows:

- $\bar{\mathcal{K}}, (\pi, \alpha) \models p$ iff $p \in \mathcal{L}(\pi(0))$
- $\bar{\mathcal{K}}, (\pi, \alpha) \models \delta$ iff $\delta \in \alpha(0)$
- $\bar{\mathcal{K}}, (\pi, \alpha) \models \neg\varphi$ iff $\bar{\mathcal{K}}, (\pi, \alpha) \not\models \varphi$
- $\bar{\mathcal{K}}, (\pi, \alpha) \models \varphi \wedge \varphi'$ iff $\bar{\mathcal{K}}, (\pi, \alpha) \models \varphi$ and $\bar{\mathcal{K}}, (\pi, \alpha) \models \varphi'$
- $\bar{\mathcal{K}}, (\pi, \alpha) \models \bigcirc\varphi$ iff $\bar{\mathcal{K}}, (\pi, \alpha)^1 \models \varphi$
- $\bar{\mathcal{K}}, (\pi, \alpha) \models \varphi\, \mathbf{U}\, \varphi'$ iff $\exists k \geq 0.\ \bar{\mathcal{K}}, (\pi, \alpha)^k \models \varphi'$, $\forall 0 \leq i < k.\ \bar{\mathcal{K}}, (\pi, \alpha)^i \models \varphi$.

**Example.** We present a topmost rewrite theory $\mathcal{R} = (\Sigma, E, R)$ that specifies Lamport's bakery protocol for mutual exclusion of an unbounded number of processes (adapted from [1,6]), and its corresponding LKS $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$. Each state of the system has the form $n\ ;\ m\ ;\ [i_1, d_1] \ldots [i_k, d_k]$, given by the operator $\_;\_;\_$ : $\mathsf{Nat}\ \mathsf{Nat}\ \mathsf{ProcSet} \rightarrow \mathsf{State}$, where $n$ is the current number in the bakery's number dispenser, $m$ is the number currently being served, and $[i_1, d_1] \ldots [i_k, d_k]$ are a set of customer processes, each with a name $i_l$ and in a *mode* $d_l$. A mode can be *idle* (not yet picked a number), $wait(n)$ (waiting with number $n$), or $crit(n)$ (being served with number $n$). The behavior is specified by the following *topmost* rewrite rules in the Maude language:

```
rl [wake]: N ; M ; [I,idle] PS =>  s N ; M ; [I,wait(N)] PS .
rl [crit]: N ; M ; [I,wait(M)] PS => N ; M ; [I,crit(M)] PS .
rl [exit]: N ; M ; [I,crit(M)] PS => N ; s M ; [I,idle] PS .
```

where natural numbers are modeled as multisets of $s$ with the multiset union operator $\_\,\_$ (empty syntax) and the empty multiset $\mathtt{0}$ (e.g., $0 = \mathtt{0}$, and $3 = \mathtt{s\ s\ s}$).

We are interested in verifying the liveness property "*process 0 is eventually served,*" under the fairness assumption "*if process 0 can eventually pick a number forever, it must pick a number infinitely often,*" expressed as the LTLR formula

$$(\Diamond\Box enabled.wake(0) \rightarrow \Box\Diamond wake(0)) \rightarrow \Diamond in.crit(0),$$

where the spatial action pattern $wake(0)$ holds if the *wake* rule is applied for process 0 (i.e., the variable `I` in the *wake* rule is matched to the term `0`), the state proposition $enabled.wake(0)$ holds in a state where process 0 is idle, and the state proposition $in.crit(0)$ holds in a state where process 0 is being served (see [1] for the mutual exclusion property).

For the set of state propositions $AP = \{in.crit(0), enabled.wake(0)\}$ and the set of spatial action patterns $ACT = \{wake(0)\}$, we can construct the related LKS $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$ for the bakery protocol specification $\mathcal{R}$. For example, given the initial state `0 ; 0 ; [0,idle]`, we obtain the infinite path in Fig. 1 within $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$ that contains an infinite number of different states. Notice that this system is infinite-state since: (i) the counters $n$ and $m$ are unbounded; and the number of customer processes is unbounded.

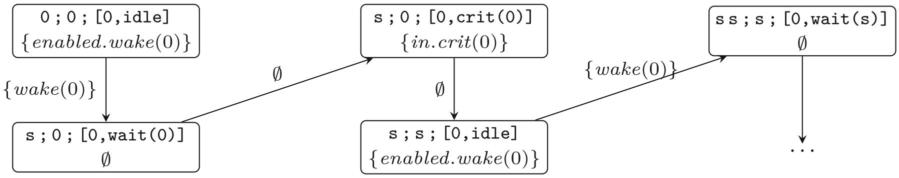

**Fig. 1.** A path from `0 ; 0 ; [0,idle]` in the LKS $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$ for the bakery protocol.

## 3 Narrowing-Based LTLR Model Checking

Narrowing [10,11] generalizes term rewriting by allowing free variables in terms and by performing unification instead of matching. An *E-unifier* of $t = t'$ is a substitution $\sigma$ such that $\sigma t =_E \sigma t'$ and $dom(\sigma) \subseteq vars(t) \cup vars(t')$, and $CSU_E(t = t')$ denotes a *complete set of E -unifiers* in which any $E$-unifier $\rho$ of $t = t'$ has a more general substitution $\sigma \in CSU_E(t = t')$, i.e., $(\exists \eta)\ \rho =_E \eta \circ \sigma$. We assume that there exists a finitary $E$-unification procedure to find a *finite* complete set $CSU_E(t = t')$ of $E$-unifiers (e.g., there exists a finitary $E$-unification procedure if $E$ has the *finite variant property* as explained in [5,7]).

**Definition 3.** *Given a* topmost *rewrite theory* $\mathcal{R} = (\Sigma, E, R)$*, each rewrite rule* $l : q \longrightarrow r \in R$ *specifies a topmost* narrowing step $t \rightsquigarrow_{l,\sigma,\mathcal{R}} t'$ *(or* $t \rightsquigarrow_{\mathcal{R}} t'$*) iff there exists an E-unifier* $\sigma \in CSU_E(t = q)$ *such that* $t' = \sigma r$*.*

For LTL model checking we can associate to $\mathcal{R} = (\Sigma, E, R)$ a corresponding *logical Kripke structure* $\mathcal{N}(\mathcal{R})_{AP}$ [6]. The states of $\mathcal{N}(\mathcal{R})_{AP}$ are $AP$-instantiated elements of $\mathcal{T}_{\Sigma/E}(\mathcal{X})_{\mathsf{State}}$ and its transitions are specified by $\rightsquigarrow_{\mathcal{R}}$. A state of $\mathcal{N}(\mathcal{R})_{AP}$ is not a *concrete state*, but a *state pattern* $t(x_1, \ldots, x_n)$ with *logical variables* $x_1, \ldots, x_n$, representing the set of all concrete states $[\theta t]_E$ that are its *ground instances*. Such a logical Kripke structure $\mathcal{N}(\mathcal{R})_{AP}$ can be considered as an abstraction of the (possibly infinite) concrete system $\mathcal{K}(\mathcal{R})_{AP}$; that is, for an LTL formula $\varphi$ and a state pattern $t$, we have:

$$\mathcal{N}(\mathcal{R})_{AP}, [t]_E \models \varphi \implies (\forall \theta : \mathcal{X} \to \mathcal{T}_\Sigma)\ \mathcal{K}(\mathcal{R})_{AP}, [\theta t]_E \models \varphi.$$

Generalizing such narrowing-based LTL model checking, this section presents narrowing-based LTLR model checking for infinite-state systems.

**One-Step Proof Terms for Narrowing.** Spatial action patterns for rewriting define their matching one-step proof terms, representing the corresponding one-step rewrites. For a topmost rewrite theory $\mathcal{R} = (\Sigma, E, R)$, one-step proof terms have the form $l(\theta)$, indicating that a rule $l : q \longrightarrow r \in R$ has been applied with a substitution $\theta$ (at the top position of the term), where $dom(\theta) \subseteq vars(q) \cup vars(r)$.

In order to define spatial action patterns for narrowing steps, we also need to have an appropriate notion of one-step proof terms for narrowing. Consider a topmost narrowing step $t \rightsquigarrow_{l,\sigma,\mathcal{R}} t'$ using a rule $l : q \longrightarrow r$. Intuitively, the rule label $l$ and the restriction of the substitution $\sigma$ to the variables in the rule[3] give the one-step proof term for the narrowing step $t \rightsquigarrow_{l,\sigma,\mathcal{R}} t'$.

**Definition 4.** *Given a* topmost *rewrite theory $\mathcal{R} = (\Sigma, E, R)$, for a topmost narrowing step $t \rightsquigarrow_{l,\sigma,\mathcal{R}} t'$ using a rule $l : q \longrightarrow r$, its one-step proof term is given by $l(\sigma|_{vars(q)\cup vars(r)})$, often denoted by $l(\sigma_l)$.*

The following lemma implies that a one-step proof term $l(\sigma_l)$ for narrowing faithfully captures its corresponding one-step proof terms $l(\theta)$ for rewriting, in the sense that $\theta =_E \eta \circ \sigma_l$ for some substitution $\eta$. This lemma is adapted from the soundness and completeness results of topmost narrowing in [16].

**Lemma 1.** *Given a topmost rewrite theory $\mathcal{R} = (\Sigma, E, R)$, for a non-variable term $u$ and a substitution $\rho$, assuming no variable in $u$ appears in the rules $R$:*

$$\begin{array}{lc} & (\exists t', \theta)\ \ l(\theta) : \rho u \longrightarrow_{\mathcal{R}} t' \\ \iff & (\exists u', \sigma, \eta)\ \ u \rightsquigarrow_{l,\sigma,\mathcal{R}} u' \ \ \wedge\ \ \rho|_{vars(u)} =_E (\eta \circ \sigma)|_{vars(u)} \end{array}$$

*where $\theta =_E (\eta \circ \sigma)|_{dom(\theta)}$ and $t' =_E \eta u'$.*

*Proof.* ($\Rightarrow$) Suppose that $l(\theta) : \rho u \longrightarrow_{\mathcal{R}} t'$ for a topmost rule $l : q \longrightarrow r$, where $dom(\theta) \subseteq vars(q) \cup vars(r)$. Then, $\theta q =_E \rho u$ and $t' = \theta r$. Since *no* variable in $u$ appears in $l : q \longrightarrow r$, we have $dom(\theta) \cap vars(u) = \emptyset$. Thus, we can define the substitution $\theta \cup \rho|_{vars(u)}$ with domain $dom(\theta) \cup vars(u)$ such that $(\theta \cup \rho|_{vars(u)})|_{dom(\theta)} = \theta$ and $(\theta \cup \rho|_{vars(u)})|_{vars(u)} = \rho|_{vars(u)}$. Since $\theta \cup \rho|_{vars(u)}$ is an $E$-unifier of $q = u$, there exist substitutions $\sigma \in CSU_E(u = q)$ and $\eta'$ satisfying $(\theta \cup \rho|_{vars(u)})|_{vars(q)\cup vars(u)} =_E \eta' \circ \sigma$ with domain $vars(q) \cup vars(u)$. Therefore, $u \rightsquigarrow_{l,\sigma,\mathcal{R}} u'$ for $u' = \sigma r$. Next, let $\eta$ be the extended substitution such that $\eta x = \eta' x$ if $x \in vars(q) \cup vars(u)$, and $\eta x = \theta x$ otherwise. Then, $\rho|_{vars(u)} =_E (\eta \circ \sigma)|_{vars(u)}$ and $\theta =_E (\eta \circ \sigma)|_{dom(\theta)}$, since $dom(\theta) \cap vars(u) = \emptyset$ and $dom(\theta) \subseteq vars(q) \cup vars(r)$. Furthermore, $t' = \theta r =_E (\eta \circ \sigma) r = \eta u'$.
($\Leftarrow$) Suppose that $u \rightsquigarrow_{l,\sigma,\mathcal{R}} u'$ and $\rho|_{vars(u)} =_E (\eta \circ \sigma)|_{vars(u)}$. Then, for a topmost rule $l : q \longrightarrow r$, $\sigma \in CSU_E(u = q)$ and $u' = \sigma r$. Since $\sigma u =_E \sigma q$ and $(vars(q) \cup vars(r)) \cap vars(u) = \emptyset$, we have $l(\sigma|_{vars(q)\cup vars(r)}) : \sigma u \longrightarrow_{\mathcal{R}} u'$. Thus, we have $l(\eta \circ \sigma|_{vars(q)\cup vars(r)}) : (\eta \circ \sigma) u \longrightarrow_{\mathcal{R}} \eta u'$, where $(\eta \circ \sigma) u =_E \rho u$, since rewrites are stable under substitutions. □

[3] Since one-step proof terms for rewriting only contain variables in rules, we restrict one-step proof terms for narrowing in the same way.

**Equational Definition of State/Event Predicates.** The semantics of a spatial action pattern can be defined by means of equations using the auxiliary operator $\_ \models \_ : \mathsf{ProofTerm}\ \mathsf{Action} \to \mathsf{Bool}$ [2]. By definition, $\delta \in \mathcal{T}_{\Sigma/E,\mathsf{Action}}$ is matched to a one-step proof term $\gamma$ iff $(\gamma \models \delta) =_E$ *true*. For a topmost rewrite theory $\mathcal{R}$, a one-step proof term $l(\theta)$ can be represented as a term

$$\{'l \ : \ 'x_1 \backslash \theta x_1 \ ; \ \dots \ ; \ 'x_m \backslash \theta x_m\}$$

of sort $\mathsf{ProofTerm}$ using the operator $\{\_ \mathord{:} \_\} : \mathsf{Qid}\ \mathsf{Substitution} \to \mathsf{ProofTerm}$, where $'l, 'x_1, \dots, 'x_m$ are quoted identifiers of sort $\mathsf{Qid}$ and $'x_1 \backslash \theta x_1; \dots ; 'x_m \backslash \theta x_m$ is a semicolon separated set of variable assignments. For the bakery example, a topmost narrowing step from the term `N ; N ; [0,idle]` by the *wake* rule gives the one-step proof term `{'wake : 'N \ N ; 'M \ N ; 'I \ 0 ; 'PS \ none}`.

For narrowing-based model checking we further require that there exists a finitary $E$-unification procedure. If a spatial action pattern $\delta$ is identified by a one-step proof term *pattern* $u_\delta$ (i.e., $(\gamma \models \delta) =_E$ *true* iff $\gamma$ is an instance of the pattern $u_\delta$),[4] and if $u_\delta$ *has complement patterns* $u_1, \dots, u_k$ (i.e., any ground one-step proof term is an instance of exactly one term in $\{u_\delta, u_1, \dots, u_k\}$), then $\delta$ can be defined by the equations:

$$u_\delta \models \delta = \mathit{true}, \quad u_1 \models \delta = \mathit{false}, \quad \dots, \quad u_k \models \delta = \mathit{false}.$$

Because the right-hand sides are all constants, these equations have the finite variant property [5], and therefore they provide a finitary $E$-unification algorithm using variant narrowing [7]. This method can also be applied for "pattern-like" state propositions (see below).

As mentioned in the introduction, effective methods have been developed to check when a term $t$ has complements and to compute such complement patterns, not only in the free case [12], but also modulo AC and modulo permutative theories [8,9]. Therefore, for unconditional rewrite theories with axioms $B$ such as those used in [8,9,12], we can determine if a one-step proof term pattern $u_\delta$ of $\delta$ has complements, compute such complement patterns, and define pattern satisfaction of $\delta$ by equations. For example, consider the spatial action pattern *wake*(0) in the bakery example (which holds if the variable `I` in the rule is matched to 0). The positive case can be defined by the following equation, where `SUBST` is a variable of sort $\mathsf{Substitution}$:

```
eq {'wake : 'I \ 0; SUBST} |= wake(0) = true .
```

For the negative cases, *wake*(0) does *not* hold when the rule label is *not* `'wake` or the value of `'I` is *not* `0`. Therefore, they can be defined by the complement patterns of 0 and `'wake` as follows.

```
eq {'wake : 'I \ s J ; SUBST} |= wake(0) = false .
eq {'crit : SUBST} |= wake(0) = false .
eq {'exit : SUBST} |= wake(0) = false .
```

[4] Many spatial action patterns, including $l$ and $l(\theta)$, are identified in this way [2,14].

The use of order-sorted signatures can greatly facilitate the existence of complement patterns that may not exist in an unsorted setting. For example, the unsorted term $y + 0 + 0$ for a signature with a constant 0, a unary $s$, and an AC symbol $+$ is shown not to have complements in [8], but can be easily shown to have complements when the signature is refined to an order-sorted signature. We illustrate this greater ease of computing complements by using the state propositions $in.crit(0)$ and $enabled.wake(0)$, whose positive cases are defined by the following equations, where `PS` is a variable of sort ProcSet:

```
eq N ; M ; [0,crit(K)] PS |= in.crit(0) = true .
eq N ; M ; [0,idle] PS |= enabled.wake(0) = true .
```

In order to define the negative cases we need to find the complement patterns for `[0,crit(K)] PS` and `[0,idle] PS`. Using subsort relations, we can define sort ModeIdleWait for *idle* and *wait(n)*, ModeWaitCrit for *wait* and *crit(n)*, and ProcSet{N0Nat} for a set of processes with non-zero identifiers as follows:[5]

```
subsorts ModeIdle ModeWait < ModeIdleWait < Mode .
subsorts ModeWait ModeCrit < ModeWaitCrit < Mode .
subsorts N0Nat < Nat .
subsorts Proc{N0Nat} < ProcSet{N0Nat} Proc < ProcSet .
```

The negative cases for the above state propositions can then be defined by the following equations, where the variable `DIW` has sort ModeIdleWait, `DWC` has sort ModeWaitCrit, and `NZPS` has sort ProcSet{N0Nat}:

```
eq N ; M ; [0,DIW] NZPS |= in.crit(0) = false .
eq N ; M ; [0,DWC] NZPS |= enabled.wake(0) = false .
```

**Narrowing-Based LKS.** For a set $AP = \{p_1, \ldots, p_n\}$ of state propositions and a set $ACT = \{\delta_1, \ldots, \delta_m\}$ of spatial action patterns defined by the equations $E$, we can also associate to a topmost rewrite theory $\mathcal{R} = (\Sigma, E, R)$ a corresponding *narrowing-based logical LKS* $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$, where:

- each state of the LKS $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ is a term in which the truth of every state proposition is decided into either *true* or *false*; and
- a transition of $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ is specified by a topmost narrowing step $\leadsto_{\mathcal{R}}$, but further instantiated into possibly several transitions so that the truth $b_i$ of each state proposition $p_i$, $1 \leq i \leq n$, and the truth $b_{n+j}$ of each spatial action pattern $\delta_j$, $1 \leq j \leq m$, are decided into either *true* or *false*.

For the bakery example, given the *logical* initial state `N ; N ; [0,idle]`, we obtain within the logical LKS $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ the infinite path in Fig. 2, which captures an infinite number of concrete paths in the concrete LKS $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$ starting from each ground instance of `N ; N ; [0,idle]`. The narrowing-based logical LKS $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ of a topmost rewrite theory $\mathcal{R}$ is formally defined as follows:

[5] Generally, to define the negative cases for $k \in \mathbb{N}$, we can define $k + 2$ subsorts Nat0, ..., Nat$k$, N$k$Nat of sort Nat, where N$k$Nat denotes a number greater than $k$.

**Definition 5.** *Given a topmost rewrite theory $\mathcal{R} = (\Sigma, E, R)$, and* finite *sets $AP = \{p_1, \ldots, p_n\} \subseteq \mathcal{T}_{\Sigma/E,\mathsf{Prop}}$ and $ACT = \{\delta_1, \ldots, \delta_m\} \subseteq \mathcal{T}_{\Sigma/E,\mathsf{Action}}$ defined by its equations $E$, the narrowing-based* logical LKS *is*

$$\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT} = (N(\mathcal{R})_{AP}, AP, \mathcal{L}_E, ACT, \longrightarrow_{\bar{\mathcal{N}}(\mathcal{R})}),$$

*where $\mathcal{L}_E([t]_E) = \{p \in AP \mid (t \models p) =_E true\}$, and:*

- *$[t]_E \in N(\mathcal{R})_{AP}$ iff $[t]_E \in \mathcal{T}_{\Sigma/E}(\mathcal{X})_{\mathsf{State}} - \mathcal{X}$, and for every state proposition $p \in AP$, either $(t \models p) =_E true$ or $(t \models p) =_E false$.*
- *$[t]_E \xrightarrow{A}_{\bar{\mathcal{N}}(\mathcal{R})} [t']_E$ iff there exist a term $u$, a substitution $\zeta$, and Boolean values $b_1, \ldots, b_{n+m} \in \{true, false\}$ such that*

$$t \rightsquigarrow_{l,\sigma,\mathcal{R}} u \ \wedge \ t' = \zeta u, \ \wedge \ A = \{\delta \in ACT \mid (\zeta(l(\sigma_l)) \models \delta) =_E true\} \ \wedge$$
$$\zeta \in CSU_E\big(\textstyle\bigwedge_{1 \le i \le n}(u \models p_i) = b_i \wedge \bigwedge_{1 \le j \le m}(l(\sigma_l) \models \delta_j) = b_{n+j}\big)$$

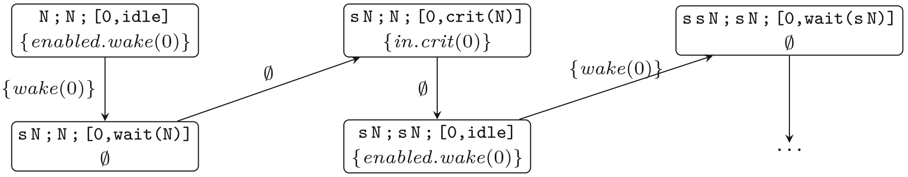

**Fig. 2.** A path from `N ; N ; [0,idle]` in the LKS $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$ for the bakery protocol.

A narrowing-based LKS $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ captures any behavior of the related concrete LKS $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$, in terms of a *simulation relation.* In the following definition we extend the usual notion of a simulation for Kripke structures to one for LKSs, which also takes into account spatial action patterns.

**Definition 6.** *Given two LKS $\bar{\mathcal{K}}_i = (S_i, AP, \mathcal{L}_i, ACT, \longrightarrow_{\bar{\mathcal{K}}_i})$, $i = 1, 2$, a binary relation $H \subseteq S_1 \times S_2$ is a* simulation *from $\bar{\mathcal{K}}_1$ to $\bar{\mathcal{K}}_2$ iff: (i) if $s_1 \, H \, s_2$, then$\mathcal{L}_1(s_1) = \mathcal{L}_2(s_2)$, and if $s_1 \, H \, s_2$ and $s_1 \xrightarrow{A}_{\bar{\mathcal{K}}} s_1'$, there exists $s_2' \in S_2$ such that $s_1' \, H \, s_2'$ and $s_2 \xrightarrow{A}_{\bar{\mathcal{K}}} s_2'$. A simulation $H$ is a* bisimulation *iff $H^{-1}$ is also a simulation, and is* total *iff for any $s_1 \in S_1$ there exists $s_2 \in S_2$ such that $s_1 \, H \, s_2$.*

As expected, if an LKS $\bar{\mathcal{K}}_2$ *simulates* $\bar{\mathcal{K}}_1$, then each infinite path in $\bar{\mathcal{K}}_1$ has a corresponding path in $\bar{\mathcal{K}}_2$, as shown in the following lemma.

**Lemma 2.** *Given a simulation $H$ from an LKS $\bar{\mathcal{K}}_1$ to $\bar{\mathcal{K}}_2$, if $s_1 \, H \, s_2$, then for each path $(\pi_1, \alpha)$ of $\bar{\mathcal{K}}_1$ beginning at $s_1$, there exists a corresponding path $(\pi_2, \alpha)$ beginning at $s_2$ such that $\pi_1(i) \, H \, \pi_2(i)$ for each $i \in \mathbb{N}$.*

*Proof.* We construct $\pi_2$ by induction. Let $\pi_2(0) = s_2$. Clearly, $\pi_1(0) \, H \, \pi_2(0)$. Next, suppose that $\pi_1(k) \, H \, \pi_2(k)$ for some $k \in \mathbb{N}$. Since $\pi_1(k) \, H \, \pi_2(k)$ and $\pi_1(k) \xrightarrow{\alpha(k)}_{\bar{\mathcal{K}}} \pi_1(k+1)$, there exists a state $s_2'$ such that $\pi_1(k+1) \, H \, s_2'$ and $\pi_2(k) \xrightarrow{\alpha(k)}_{\bar{\mathcal{K}}} s_2'$. Then, we choose $\pi_2(k+1) = s_2'$. □

Suppose that $s_0^1 \, H \, s_0^2$ for a simulation $H$ from $\bar{\mathcal{K}}_1$ to $\bar{\mathcal{K}}_2$. If there exists a counterexample $(\pi_1, \alpha_1)$ in $\bar{\mathcal{K}}_1$ starting from $s_0^1$, then by the above lemma, there exists a corresponding counterexample $(\pi_2, \alpha_2)$ in $\bar{\mathcal{K}}_2$ starting from $s_0^2$ such that $\mathcal{L}_1(\pi_1(i)) = \mathcal{L}_2(\pi_2(i))$ and $\alpha_1(i) = \alpha_2(i)$ for each $i \in \mathbb{N}$. Therefore:

**Corollary 1.** *Given a simulation $H$ from an LKS $\bar{\mathcal{K}}_1$ to $\bar{\mathcal{K}}_2$, if $s_0^1 \, H \, s_0^2$, then for any LTLR formula $\varphi$, $\bar{\mathcal{K}}_2, s_0^2 \models \varphi$ implies $\bar{\mathcal{K}}_1, s_0^1 \models \varphi$. In particular, if $H$ is a bisimulation, then $\bar{\mathcal{K}}_2, s_0^2 \models \varphi$ iff $\bar{\mathcal{K}}_1, s_0^1 \models \varphi$.*

For a narrowing-based LKS $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$, each logical state is clearly related to a concrete state in $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$ in terms of the *$E$-subsumption* relation. The $E$-subsumption $t \preccurlyeq_E t'$ holds iff there exists a substitution $\sigma$ with $t =_E \sigma t'$, meaning that $t'$ is *more general* than $t$ modulo $E$.

**Lemma 3.** *Given a topmost rewrite theory $\mathcal{R} = (\Sigma, E, R)$ and sets $AP$ and $ACT$ defined by $E$, $\preccurlyeq_E$ is a total simulation from $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$ to $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$.*

*Proof.* Suppose that $[t]_E \xrightarrow{A}_{\bar{\mathcal{K}}(\mathcal{R})} [t']_E$ and $t \preccurlyeq_E u$ for $u \in N(\mathcal{R})_{AP}$. Given $AP = \{p_1, \ldots, p_n\}$ and $ACT = \{\delta_1, \ldots, \delta_m\}$, fix $b_1, b_2, \ldots, b_{n+m} \in \{\mathit{true}, \mathit{false}\}$ such that $b_i =_E (t' \models p_i)$ for $1 \le i \le n$ and $b_{n+j} =_E (l(\theta) \models \delta_j)$ for $1 \le j \le m$. By definition, there is an one-step rewrite $l(\theta) : t \longrightarrow_{\mathcal{R}} t'$. By Lemma 1, there is a narrowing step $u \rightsquigarrow_{l,\sigma,\mathcal{R}} u'$ such that $t' =_E \eta u'$ and $\theta =_E (\eta \circ \sigma)|_{dom(\theta)}$. Thus, there exists $\zeta \in CSU_E(\bigwedge_{1 \le i \le n}(u' \models p_i) = b_i \;\wedge\; \bigwedge_{1 \le j \le m}(l(\sigma_l) \models \delta_j) = b_{n+j})$. By definition, $[u]_E \xrightarrow{A}_{\bar{\mathcal{N}}(\mathcal{R})} [\zeta u']_E$. Notice that $\bigwedge_{1 \le i \le n} \eta\big((u' \models p_i) =_E b_i\big)$ and $\bigwedge_{1 \le j \le m} \eta\big((l(\sigma_l) \models \delta_j) =_E b_{n+j}\big)$. Therefore, $\eta \preccurlyeq_E \zeta$, and $t' =_E \eta u \preccurlyeq_E \zeta u'$. □

By Corollary 1, this lemma implies that any LTLR formula $\varphi$ satisfied in a narrowing-based LKS $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ from a logical state $t$ is also satisfied in the concrete LKS $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$ from each ground instance of $t$.

In general, $\preccurlyeq_E$ is *not* a bisimulation between $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$ and $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$. For the bakery example, although `0 ; 0 ; [I,wait(0)]` $\preccurlyeq_E$ $\mathtt{N ; M ; PS}_1$ holds, there exists the transition $\mathtt{N ; M ; PS}_1 \xrightarrow{\{wake(0)\}}_{\bar{\mathcal{N}}(\mathcal{R})} \mathtt{s\,N ; M ; PS}_2\ \mathtt{[0,wait(N)]}$, in $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ with the substitution $\mathtt{PS}_1 \backslash \mathtt{PS}_2\ \mathtt{[0,idle]}$, but *no corresponding transition* exists from `0 ; 0 ; [I,wait(0)]` in $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$. However, any *finite* path in $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ can be *instantiated* to a corresponding concrete path in $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$ (e.g., the above transition can be instantiated as the transition $\mathtt{0 ; 0 ; [0,idle]} \xrightarrow{\{wake(0)\}}_{\bar{\mathcal{K}}(\mathcal{R})} \mathtt{s ; 0 ; [0,wait(0)]}$ in $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$).

**Lemma 4.** *For a* finite *path $u_1 \xrightarrow{A_1}_{\bar{\mathcal{N}}(\mathcal{R})} \cdots \xrightarrow{A_{n-1}}_{\bar{\mathcal{N}}(\mathcal{R})} u_n$ of $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$, there is $t_1 \xrightarrow{A_1}_{\bar{\mathcal{K}}(\mathcal{R})} \cdots \xrightarrow{A_{n-1}}_{\bar{\mathcal{K}}(\mathcal{R})} t_n$ in $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$ with $t_i \preccurlyeq_E u_i$, $1 \le i \le\ n$.*

*Proof.* Since $u_1 \xrightarrow{A_1}_{\bar{\mathcal{N}}(\mathcal{R})} u_2$, by definition, there are substitutions $\sigma_1$ and $\zeta_1$ such that $u_1 \rightsquigarrow_{l_1,\sigma_1,\mathcal{R}} u_2'$ by a topmost rule $l_1 : q_1 \to r_1 \in R$ and $u_2 = \zeta_1 u_2'$. Since $\sigma u_1 =_E \sigma q_1$ and $u_2 = \zeta_1 u_2' = (\zeta_1 \circ \sigma_1) r_1$, $(\zeta_1 \circ \sigma_1) u_1 \longrightarrow_{\mathcal{R}} u_2$. Similarly, $(\zeta_2 \circ \sigma_2) u_2 \longrightarrow_{\mathcal{R}} u_3$, etc. By composing them, $(\zeta_{n-1} \circ \sigma_{n-1} \circ \cdots \circ \zeta_2 \circ \sigma_2 \circ \zeta_1 \circ \sigma_1) u_1 \longrightarrow_{\mathcal{R}} \cdots \longrightarrow_{\mathcal{R}} (\zeta_{n-1} \circ \sigma_{n-1}) u_{n-1} \longrightarrow_{\mathcal{R}} u_n$. Let $\rho$ be a ground substitution instantiating every variable in the path. Then, $(\rho \circ \zeta_{n-1} \circ \sigma_{n-1} \circ \cdots \circ \zeta_2 \circ \sigma_1) u_1 \longrightarrow_{\mathcal{R}} \cdots \longrightarrow_{\mathcal{R}} (\rho \circ \zeta_{n-1} \circ \sigma_{n-1}) u_{n-1} \longrightarrow_{\mathcal{R}} \rho u_n$ gives the desired path. □

Recall that counterexamples of *safety properties* are characterized by finite sequences [4]. Therefore, the above lemma guarantees that $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ does *not* generate spurious counterexamples for safety properties, since any finite counterexample in $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ has a corresponding *real* counterexample in $\bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}$. Together with Corollary 1 and Lemma 3, we have:

**Theorem 1.** *Given a topmost rewrite theory* $\mathcal{R} = (\Sigma, E, R)$*, and* finite *sets* $AP$ *and* $ACT$ *defined by* $E$*, for a safety LTLR formula* $\varphi$ *and a pattern* $t \in N(\mathcal{R})_{AP}$*:* $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}, [t]_E \models \varphi \iff (\forall \theta : \mathcal{X} \to \mathcal{T}_\Sigma)\ \bar{\mathcal{K}}(\mathcal{R})_{AP,ACT}, [\theta t]_E \models \varphi$.

## 4 Abstract Narrowing-Based LTLR Model Checking

A narrowing-based LKS $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ often has an infinite number of *logical* states (e.g., Fig. 2). For narrowing-based LTL model checking, the paper [1] has proposed two abstraction methods to reduce an infinite narrowing-based Kripke structure, namely, *folding abstractions* and *equational abstractions*. This section extends those abstraction techniques to narrowing-based LTLR model checking for trying to reduce an infinite *narrowing-based LKS* to a finite one.

**Folding Abstractions.** Given a *transition system* $\mathcal{A} = (A, \longrightarrow_{\mathcal{A}})$ with a set of states $A$ and a transition relation $\longrightarrow_{\mathcal{A}} \subseteq A^2$, we can reduce it by collapsing each state $a$ into a *previously seen* state $b$, while traversing $\mathcal{A}$ from a set of initial states $I \subseteq A$, whenever $b$ is *more general* than $a$ according to a folding relation $a \preccurlyeq b$ [6]. For a set of states $B \subseteq A$, let $Post_{\mathcal{A}}(B) = \{a \in A \mid \exists b \in B.\, b \longrightarrow_{\mathcal{A}} a\}$ (i.e., the *successors* of $B$) and $Post^*_{\mathcal{A}}(B) = \bigcup_{i \in \mathbb{N}} (Post_{\mathcal{A}})^i(B)$.

**Definition 7.** *Given* $\mathcal{A} = (A, \longrightarrow_{\mathcal{A}})$ *and a folding relation* $\preccurlyeq\, \subseteq A^2$*, the* folding abstraction *of* $\mathcal{A}$ *from* $I \subseteq A$ *is* $\mathcal{R}each^{\preccurlyeq}_{\mathcal{A}}(I) = (Post^*_{\mathcal{A}\preccurlyeq}(I), \longrightarrow_{\mathcal{R}each^{\preccurlyeq}_{\mathcal{A}}(I)})$*, where:* $Post^*_{\mathcal{A}\preccurlyeq}(I) = \bigcup_{i \in \mathbb{N}} Post^i_{\mathcal{A}\preccurlyeq}(I)$ *and* $\longrightarrow_{\mathcal{R}each^{\preccurlyeq}_{\mathcal{A}}(I)} = \bigcup_{i \in \mathbb{N}} \longrightarrow^{\preccurlyeq}_{\mathcal{A},i}$ *such that:*

$$\begin{aligned}
&Post^0_{\mathcal{A}\preccurlyeq}(I) = I, \qquad\qquad \longrightarrow^{\preccurlyeq}_{\mathcal{A},0} = \emptyset,\\
&Post^{n+1}_{\mathcal{A}\preccurlyeq}(I) = \{a \in Post_{\mathcal{A}}(Post^n_{\mathcal{A}\preccurlyeq}(I)) \mid \forall l \leq n\ \forall b \in Post^l_{\mathcal{A}\preccurlyeq}(I).\, a \not\preccurlyeq b\},\\
&\longrightarrow^{\preccurlyeq}_{\mathcal{A},n+1} = \{(a, a') \in Post^n_{\mathcal{A}\preccurlyeq}(I) \times \bigcup_{0 \leq i \leq n+1} Post^i_{\mathcal{A}\preccurlyeq}(I) \mid \exists b \in Post_{\mathcal{A}}(a).\, b \preccurlyeq a'\}.
\end{aligned}$$

For the bakery example, using the $E$-subsumption $\preccurlyeq_E$ as a folding relation, we have the *finite* folding abstraction $\mathcal{R}each^{\preccurlyeq_E}_{\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}}(\{$`N ; N ; [0,idle][s,idle]`$\})$ of $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ from the initial state `N ; N ; [0,idle][s,idle]` in Fig. 3.

If a folding relation $\preccurlyeq$ is a total simulation from $\mathcal{A}$ to $\mathcal{A}$, then $\mathcal{R}each^{\preccurlyeq}_{\mathcal{A}}(I)$ simulates the *reachable* subsystem $\mathcal{R}each_{\mathcal{A}}(I) = (Post^*_{\mathcal{A}}(I), \longrightarrow_{\mathcal{A}} \cap\, Post^*_{\mathcal{A}}(I)^2)$ that only contains reachable states from $I$ (i.e., $\preccurlyeq$ is a total simulation from $\mathcal{R}each_{\mathcal{A}}(I)$ to $\mathcal{R}each^{\preccurlyeq}_{\mathcal{A}}(I)$) [1]. Indeed, $\preccurlyeq_E$ for a topmost rewrite theory $\mathcal{R}$ is a total simulation from $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ to $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ (which can be proved in a similar way to Lemma 3). Therefore, $\preccurlyeq_E$ defines a total simulation from $\mathcal{R}each_{\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}}(I)$ to $\mathcal{R}each^{\preccurlyeq_E}_{\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}}(I)$. Consequently, by Corollary 1:

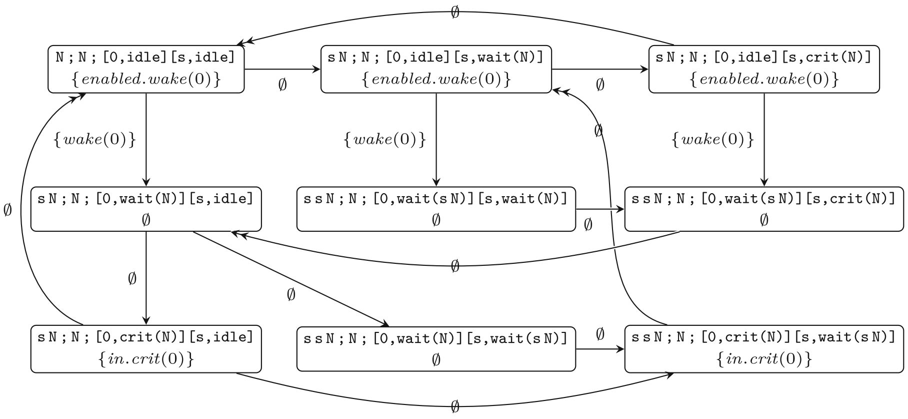
N ; N ; [0,idle][s,idle]
{enabled.wake(0)}
s N ; N ; [0,idle][s,wait(N)]
{enabled.wake(0)}
s N ; N ; [0,idle][s,crit(N)]
{enabled.wake(0)}
s N ; N ; [0,wait(N)][s,idle]
∅
s s N ; N ; [0,wait(s N)][s,wait(N)]
∅
s s N ; N ; [0,wait(s N)][s,crit(N)]
∅
s N ; N ; [0,crit(N)][s,idle]
{in.crit(0)}
s s N ; N ; [0,wait(N)][s,wait(s N)]
∅
s s N ; N ; [0,crit(N)][s,wait(s N)]
{in.crit(0)}
{wake(0)}
{wake(0)}
{wake(0)}
∅

**Fig. 3.** A folding abstraction for the bakery protocol using the folding relation $\preccurlyeq_E$, where a double-headed arrow denotes a "folded" transition.

**Theorem 2.** *For an LTLR formula $\varphi$ and a pattern $t \in N(\mathcal{R})_{AP}$, we have that $\mathcal{R}each^{\preccurlyeq_E}_{\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}}(\{[t]_E\}), [t]_E \models \varphi$ implies $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}, [t]_E \models \varphi$.*

For the bakery example, the liveness property $\Diamond in.crit(0)$ under the fairness assumption $\Diamond\Box enabled.wake(0) \rightarrow \Box\Diamond wake(0)$ holds in the folding abstraction $\mathcal{R}each^{\preccurlyeq_E}_{\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}}(\{$`N ; N ; [0,idle][s,idle]`$\})$ of Fig. 3, because any infinite paths continuously staying in the first row violate the fairness assumption. Hence, this property is also satisfied for any related concrete system.

**Equational Abstractions.** In general, a folding abstraction of a narrowing-based LKS is *not* finite. For the bakery example, there exists an infinite path within the folding abstraction from `N ; N ; [0,idle] IS` in Fig. 4, which keeps incrementing the number of processes with instantiations. To further reduce an infinite logical state space, we can apply equational abstractions to eventually obtain a finite abstract narrowing-based LKS for LTLR model checking.

Given a rewrite theory $\mathcal{R} = (\Sigma, E, R)$, by adding a set of equations $G$ such that $true \neq_{E\cup G} false$, we define an *equational abstraction* $\mathcal{R}/G = (\Sigma, E \cup G, R)$ [15]. It specifies the quotient abstraction $\bar{\mathcal{N}}(\mathcal{R}/G)_{AP,ACT}$ by the equivalence relation $\equiv_G$ on states, namely, $[t]_E \equiv_G [t']_E$ iff $t =_{E\cup G} t'$. Provided that a set of state propositions $AP$ and a set of spatial action patterns $ACT$ are defined by $E$, the condition $true \neq_{E\cup G} false$ ensures that any two states with $t =_{E\cup G} t'$ satisfy the same set of state propositions. Similarly, any two one-step proof terms with $l(\sigma_l) =_{E\cup G} l'(\sigma_{l'})$ satisfy the same set of spatial action patterns.

Similar to the cases of LTL model checking [1,15], an equational abstraction $\bar{\mathcal{N}}(\mathcal{R}/G)_{AP,ACT}$ simulates the narrowing-based LKS $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$.

**Lemma 5.** *Given a topmost rewrite theory $\mathcal{R} = (\Sigma, E, R)$, finite sets $AP$ and $ACT$ defined by $E$, and a set $G$ of equations, if $true \neq_{E\cup G} false$, then there exists a total simulation from $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ to $\bar{\mathcal{N}}(\mathcal{R}/G)_{AP,ACT}$.*

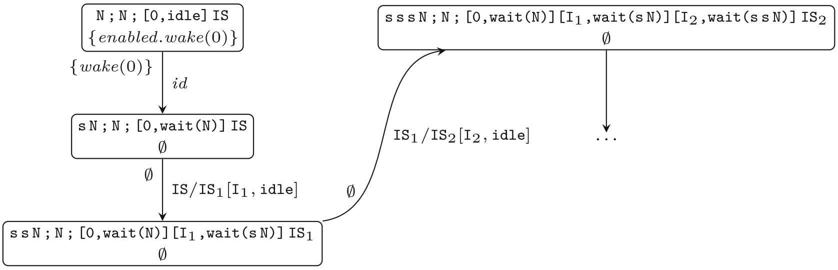

**Fig. 4.** An infinite path in the folding abstraction for the bakery protocol with an unbounded number of processes, where `IS` stands for a set of *idle* processes.

*Proof.* Let $H_G = \{([t]_E, [t]_{E\cup G}) \mid t \in N(\mathcal{R})_{AP}\}$. Suppose that $[t]_E \xrightarrow{A}_{\bar{\mathcal{N}}(\mathcal{R})} [t']_E$ and $t =_{E\cup G} u$. By definition, there are $\sigma$ and $\zeta$ such that $t \rightsquigarrow_{l,\sigma,\mathcal{R}} t''$ by a rule $l : q \longrightarrow r \in R$ and $t' = \zeta t''$, where $\sigma \in \mathit{CSU}_E(t = q)$, $t'' = \sigma r$, and $\zeta \in \mathit{CSU}_E(\bigwedge_{1\le i\le n}(t'' \models p_i) = b_i \;\wedge\; \bigwedge_{1\le j\le m}(l(\sigma_l) \models \delta_j) = b_{n+j})$ for some $b_1, \ldots, b_{n+m} \in \{\mathit{true}, \mathit{false}\}$, given $AP = \{p_1, \ldots, p_n\}$ and $ACT = \{\delta_1, \ldots, \delta_m\}$. Since $\sigma \in \mathit{CSU}_E(t = q)$, $\exists \sigma' \in \mathit{CSU}_{E\cup G}(u = q)$ such that $\sigma =_{E\cup G} \sigma'$. Then, $u \rightsquigarrow_{l,\sigma',\mathcal{R}/G} u'$ using the same rule $l : q \longrightarrow r$, where $u' = \sigma' r =_{E\cup G} \sigma r = t''$. Notice that $(t'' \models p_i) =_{E\cup G} (u' \models p_i)$ and $(l(\sigma_l) \models \delta_j) =_{E\cup G} (l(\sigma'_l) \models \delta_j)$. Thus, $\exists \zeta' \in \mathit{CSU}_{E\cup G}(\bigwedge_{1\le i\le n}(u' \models p_i) = b_i \;\wedge\; \bigwedge_{1\le j\le m}(l(\sigma'_l) \models \delta_j) = b_{n+j})$ with $\zeta =_{E\cup G} \zeta'$. Thus, $[u]_{E\cup G} \xrightarrow{A}_{\bar{\mathcal{N}}(\mathcal{R}/G)} [\zeta' u']_{E\cup G}$, where $\zeta' u' =_{E\cup G} \zeta t'' = t'$. Since $\mathit{true} \neq_{E\cup G} \mathit{false}$, $[t']_E$ and $[\zeta' u']_{E\cup G}$ satisfy the same state propositions. Therefore, $H_G$ is a total simulation from $\bar{\mathcal{N}}(\mathcal{R})_{AP,ACT}$ to $\bar{\mathcal{N}}(\mathcal{R}/G)_{AP,ACT}$. □

For the bakery example, by adding the following equations that collapses extra waiting processes with non-zero identifiers, where `ICPS` denotes a set of *idle* or *crit* processes, and `WP3` denotes *zero or at most three wait* processes:

```
eq [NZ,D] = [D] .                               --remove non-zero identifiers
eq s s s N M ; M ; ICPS WP3 [wait(s N M)] [wait(s s N M)]
 =   s s N M ; M ; ICPS WP3 [wait(s N M)] .
```

we have the folded abstract narrowing-based LKS in Fig. 5, provided with the extra spatial action pattern *wake* that holds if the *wake* rule is applied.

We can easily see that there is a counterexample of the property $\Diamond \mathit{in.crit}(0)$ under $\Diamond\Box \mathit{enabled.wake}(0) \rightarrow \Box\Diamond \mathit{wake}(0)$ in which the *wake* rule is continuously applied forever, which is impossible if there is a finite number of processes. Assuming the extra fairness assumption $\Box\Diamond\neg \mathit{wake}$, the property $\Diamond \mathit{in.crit}(0)$ is now satisfied since any infinite paths staying in the first column forever violate $\Diamond\Box \mathit{enabled.wake}(0) \rightarrow \Box\Diamond \mathit{wake}(0)$, and any paths staying in a self loop forever violate $\Box\Diamond\neg \mathit{wake}$. Consequently, under the fairness assumptions, $\Diamond \mathit{in.crit}(0)$ is satisfied for an unbounded number of processes.

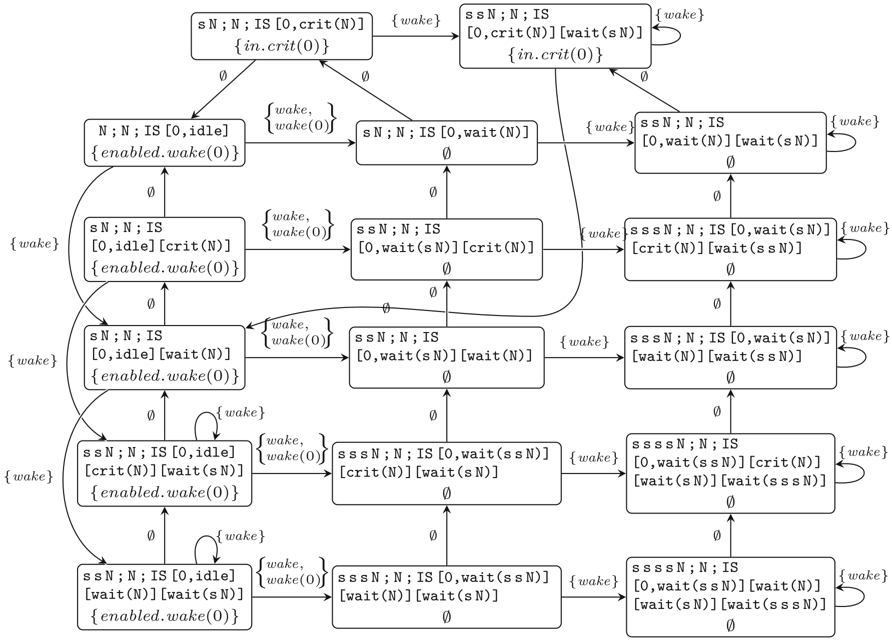

**Fig. 5.** An folded equational abstraction for the bakery protocol.

## 5 Related Work and Conclusions

A number of infinite-state model checking methods have been developed based on symbolic and abstraction techniques; see [1,6] for an overview and comparison with narrowing-based model checking. To the best of our knowledge, our work proposes the first *symbolic* model checking method to verify LTLR properties of infinite-state systems. For finite-state systems the paper [2] presents various model checking algorithms for LTLR properties. LTLR is a sublogic of $TLR^*$ that generalizes the state-based logic $CTL^*$ (see [14] for related work). On the topic of *complement patterns*, the most closely related work is [8,9,12]. We plan to use their ideas, as well as ongoing work by Skeirik and Meseguer on the concept of *B-linear terms* in order-sorted signatures, which are pattern terms whose syntactic structure guarantees the existence of complements modulo $B$, to automate the full equational definition of satisfaction of spatial action patterns.

In conclusion, this work should be understood as a contribution that increases the expressive power of infinite-state model checking methods. Specifically, the expressive power of narrowing-based infinite-state logical model checking has been extended form LTL to LTLR, allowing temporal properties that can use both state propositions and spatial action patterns. This extension is nontrivial because of the need for building a symbolic transition system where states are $AP$-instantiated and transitions are $ACT$-instantiated.

All the necessary theoretical foundations are now in place for embarking into a future implementation of a narrowing-based LTLR model checker in Maude in the spirit of the similar LTL tool described in [1]. As done in [1], for the LTLR tool we will be able to rely on the extensive body of work on efficient LTLR model checking algorithms described in [2]. Beyond these goals, the integration of constraints and SMT solving within the planned narrowing-based LTLR model checker, as well as the study of more flexible "stuttering" $AP/ACT$-simulations, are also exciting possibilities.

**Acknowledgments.** This work has been supported in part by NSF Grant CNS 13-19109 and AFOSR Grant FA8750-11-2-0084.

## References

1. Bae, K., Escobar, S., Meseguer, J.: Abstract logical model checking of infinite-state systems using narrowing. In: RTA, LIPIcs, vol. 21, pp. 81–96 (2013)
2. Bae, K., Meseguer, J.: Model checking linear temporal logic of rewriting formulas under localized fairness. Sci. Comput. Program (2014). http://dx.doi.org/10.1016/j.scico.2014.02.006 (To appear)
3. Chaki, S., Clarke, E.M., Ouaknine, J., Sharygina, N., Sinha, N.: State/event-based software model checking. In: Boiten, E.A., Derrick, J., Smith, G.P. (eds.) IFM 2004. LNCS, vol. 2999, pp. 128–147. Springer, Heidelberg (2004)
4. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. The MIT Press (2001)
5. Comon-Lundh, H., Delaune, S.: The finite variant property: how to get rid of some algebraic properties. In: Giesl, J. (ed.) RTA 2005. LNCS, vol. 3467, pp. 294–307. Springer, Heidelberg (2005)
6. 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)
7. Escobar, S., Sasse, R., Meseguer, J.: Folding variant narrowing and optimal variant termination. J. Algebraic Logic Program. **81**, 898–928 (2012)
8. Fernández, M.: AC complement problems: satisfiability and negation elimination. J. Symb. Comput. **22**(1), 49–82 (1996)
9. Fernández, M.: Negation elimination in empty or permutative theories. J. Symb. Comput. **26**(1), 97–133 (1998)
10. Hullot, J.M.: Canonical forms and unification. In: Bibel, W., Kowalski, R. (eds.) 5th Conference on Automated Deduction Les Arcs. LNCS. Springer, Heidelberg (1980)
11. Jouannaud, J.P., Kirchner, C., Kirchner, H.: Incremental construction of unification algorithms in equational theories. In: Diaz, J. (ed.) ICALP. LNCS, pp. 361–373. Springer, Heidelberg (1983)
12. Lassez, J.L., Marriott, K.: Explicit representation of terms defined by counter examples. J. Autom. Reasoning **3**(3), 301–317 (1987)
13. Meseguer, J.: Conditional rewriting logic as a unified model of concurrency. Theor. Comput. Sci. **96**(1), 73–155 (1992)
14. 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)

15. Meseguer, J., Palomino, M., Martí-Oliet, N.: Equational abstractions. Theor. Comput. Sci. **403**(2–3), 239–264 (2008)
16. Meseguer, J., Thati, P.: Symbolic reachability analysis using narrowing and its application to verification of cryptographic protocols. Higher-Order Symbolic Comput. **20**(1–2), 123–160 (2007)