Identification and Counter Abstraction for Full Virtual Symmetry

Ou Wei, Arie Gurfinkel, and Marsha Chechik

Department of Computer Science, University of Toronto
Email: {owei,arie,chechik}@cs.toronto.edu

Abstract. Symmetry reduction is an effective approach for dealing with the state explosion problem: when applicable, it enables exponential statespace reduction. Thus, it is appealing to extend the power of symmetry reduction to systems which are "not quite symmetric". Emerson et al. identified a class of these, called *virtually* symmetric [9]. In this paper, we study symmetry from the point of view of abstraction, which allows us to present an efficient procedure for identifying full virtual symmetry. We also explore techniques for combining virtual symmetry with symbolic model-checking and report on experiments that illustrate the feasibility of our approach.

1 Introduction

Symmetry reduction (e.g., [7, 10]) is a technique for solving the state explosion problem in model-checking. Symmetry is naturally exhibited in systems or protocols that consist of synchronization and coordination of several identical processes. Such symmetry can be seen as a form of redundancy, and model checking can then be performed on the symmetry-reduced quotient structure which is bisimilar to, and often substantial smaller than, the original system structure. Unfortunately, many protocols are not symmetric: even in cases where process descriptions exhibit a high degree of similarity, a slight difference among them results in an asymmetric global behavior. To extend symmetry reduction to such systems, Emerson et al. [9] defined *virtual symmetry* as the most general condition under which the structure of a system is bisimilar to its symmetry-reduced quotient structure, and thus symmetry reduction can be applied.

Although virtual symmetry increases a potential domain of problems that can be symmetry reduced, its practical application depends on successful solutions to the following questions: (1) How to identify virtual symmetry without building the entire system (which is typically infeasible)? (2) How to apply the knowledge that a system is virtually symmetric to effectively solve the resulting *symbolic* model-checking problem?

In this paper, we answer these questions for *fully* virtually symmetric systems, i.e., systems which are virtually symmetric up to exchanging the roles of processes. This form of symmetry typically arises in systems composed of many similar, but not identical processes, due to different process priorities for accessing a shared resource. An example of such a system is Readers-and-Writers (R&W): a variant of a well-known mutual exclusion protocol (MUTEX), where writer processes are given a higher priority than reader processes for entering the critical section [9]. Like full symmetry, full virtual symmetry may lead to an exponential reduction on the statespace of the system.

We start by formalizing the connection between symmetry reduction and abstraction in Section 3, and use it to derive an alternative (and simpler) characterization of virtual symmetry. The remainder of the work reported here is based on this characterization.

We then address the problem of an effective identification of cases where the system exhibits full virtual symmetry. In practice, full symmetry of a system is ensured by restricting its description using a special syntax [11, 12]. In Section 4, we show that lack of regularity in asymmetric systems makes it difficult to ensure full virtual symmetry through such a syntactic restriction. However, based on our characterization of virtual symmetry, we show that identification of full virtual symmetry can be reduced to satisfiability of a quantifier-free Presburger arithmetic (QFP) formula built directly from the syntactic description of the system.

Afterwards, we turn to the problem of combining symbolic model-checking with symmetry reduction. The naive construction of a symmetry-reduced quotient structures requires building an orbit relation, which defines the orbit equivalence between states. Clarke et al. [7] proved that BDD-based symbolic representation of the orbit relation is often exponential. Thus, it was assumed that symmetry and symbolic model-checking do not mix well. However, Emerson et al. [11, 12] have shown that the quotient of a fully symmetric system can be constructed without the orbit relation via *generic representatives* (or a counter abstraction). In Section 5, we extend this technique to handle fully virtually symmetric systems.

In Section 6, we report on experiments of identifying full virtual symmetry and applying counter abstraction-based symbolic model-checking on two families of problems. Section 7 concludes the paper and compares our result with related work.

2 Background

We assume that the reader is familiar with symmetry reduction. Below, we recall some specific concepts and fix the notation.

Structures and Simulations. A structure M is a pair (S, R) where S is a finite set of states and $R \subseteq S \times S$ is the transition relation. The domain of R is denoted by $Dom(R) \triangleq \{s \in S \mid \exists t \in S \cdot (s,t) \in R\}$. We use $s \to t$ and (s,t) interchangeably to denote a transition in R.

Let $M_1 = (S_1, R_1)$ and $M_2 = (S_2, R_2)$ be two structures. Then, M_2 *simulates* M_1 with respect to a relation $\rho \subseteq S_1 \times S_2$, denoted by $M_1 \preceq_\rho M_2$, if and only if for $(s_1, s_2) \in \rho$, the following condition holds:

$$\forall t_1 \in S_1 \cdot (s_1, t_1) \in R_1 \Rightarrow \exists t_2 \in S_2 \cdot (s_2, t_2) \in R_2 \wedge (t_1, t_2) \in \rho$$

Furthermore, M_2 is *bisimilar* to M_1 with respect to ρ, denoted by $M_1 \equiv_\rho M_2$, if both $M_1 \preceq_\rho M_2$ and $M_2 \preceq_{\rho^{-1}} M_1$.

Symmetry Reduction. Let $M = (S, R)$ be a structure and G be a permutation group on S. The group G induces an equivalence partition on S. The equivalence class of a state s is called the *orbit* of s under G, defined by $\theta_G(s) \triangleq \{s' \in S \mid \exists \sigma \in G \cdot \sigma(s) = s'\}$. We use $\theta(s)$ to denote the orbit of s when G is clear from the context. The extension of θ to a set of states $Q \subseteq S$ is defined by $\theta(Q) \triangleq \bigcup_{s \in Q} \theta(s)$.

The *quotient structure* of M induced by G is $M^G = (S^G, R^G)$ where $S^G = \theta(S)$, and $\forall s,t \in S \cdot (\theta(s), \theta(t)) \in R^G \Leftrightarrow \exists s' \in \theta(s) \cdot \exists t' \in \theta(t) \cdot (s', t') \in R$. A permutation group G is called an *automorphism group* for M if it preserves the transition relation R, i.e., $\forall s,t \in S \cdot (s,t) \in R \Rightarrow \forall \sigma \in G \cdot (\sigma(s), \sigma(t)) \in R$. A structure M is called *symmetric* with respect to a permutation group G, if G is its automorphism group. In this case, M is bisimilar to its symmetry-reduced quotient structure.

Theorem 1. [7, 10] *Let $M = (S, R)$ be a structure, G be a permutation group acting on S, and $\rho_G \triangleq \{(s, \theta(s)) | s \in S\}$. Then, $M \equiv_{\rho_G} M^G$ if G is an automorphism group for M.*

Note that temporal logics such that CTL* and modal μ-calculus are invariant under bisimulation [5]. Therefore, model checking a temporal logic formula φ on M can be reduced to model checking φ on M^G, provided that the atomic propositions of φ are preserved by ρ_G.

Compositional Structures. Symmetry reduction is often applied to a parallel composition of similar processes. Such a composition is modeled by a structure whose statespace is assignments of local states to each process.

Let $I = [1..n]$ be the index set of n processes which have the same set of local states $\mathcal{L}$. The composition of the processes is modeled by a *compositional structure* $M = (S, R)$, where $S = \mathcal{L}^n$. Then a global state s in S is an n-tuple $(l_1, \ldots, l_n) \in \mathcal{L}^n$. For each $i \in I$, we use $s(i)$ to denote the value of l_i, i.e., the current local state of the ith process, P_i, at s. Let $K \subseteq I$ be a set of processes. The *group counter* of a local state L with respect to K is a function $\#L[K] : \mathcal{L}^n \to [0..n]$ such that for any global state s, $\#L[K](s) = |\{i \in K \mid s(i) = L\}|$. That is, $\#L[K](s)$ is the number of process in K whose current state at s is L. In particular, if $K = I$, we use $\#L$ to denote $\#L[I]$, and call $\#L$ the *total counter* of L.

The *full symmetry group* of I, i.e., the group of all permutations acting on I, is denoted by $Sym(I)$. A permutation $\sigma \in Sym(I)$ is extended to act on a state s of a compositional structure M as follows: $\forall i,j \in I \cdot \sigma(s)(i) = s(j) \Leftrightarrow \sigma(i) = j$. In the rest of the paper, we do not distinguish between a permutation group on S or I. A structure M is called *fully* symmetric if M is symmetric with respect to $Sym(I)$.

3 Abstraction and Virtual Symmetry

In this section, we formalize the connection between symmetry reduction and abstraction. We then show how this connection can be used to establish a necessary and sufficient condition for the application of symmetry reduction. This condition, referred to by Emerson et al. as *virtual symmetry* [9], generalizes the notion of automorphism-based symmetry [7, 10] (see Theorem 1) and increases the applicability of symmetry reduction.

Given a structure $M = (S, R)$ and a set of *abstract* states S_α, an *abstraction* $\alpha : S \to S_\alpha$ is a total function that maps each state $s \in S$ to a state $a \in S_\alpha$. S and S_α are the *concrete* and the *abstract* statespaces, respectively. We define $\gamma : S_\alpha \to 2^S$ to be a *concretization function* that maps each abstract state s_α to a set of concrete states corresponding to it, i.e., $\gamma(a) \triangleq \{s \in S \mid \alpha(s) = a\}$. Following [8], we extend α to the transition relation as follows. A relation $R_\alpha^{\exists\exists} \subseteq S_\alpha \times S_\alpha$ is an *existential abstraction* of R where $(a,b) \in R_\alpha$ if and only if R has a transition between *some* concretizations

Abstraction	Symmetry Reduction
abstract statespace : S_α	orbits induced by G: $\theta_G(S)$
abstraction function : α	membership determined by orbit relation: $\alpha_G(s) = \theta(s)$
concretization function : γ	identity function: $\gamma_G(\theta(s)) = \theta(s)$
existential abstraction of R : $R_\alpha^{\exists\exists}$	quotient of R with respect to G: R^G
abstract equivalence: $\alpha(s) = \alpha(t)$	orbit equivalence: $\theta(s) = \theta(t) \Leftrightarrow \exists \sigma \in G \cdot s = \sigma(t)$

Table 1. A mapping between abstraction and symmetry reduction.

of a and b; $R_\alpha^{\forall\exists}$ is a *universal abstraction* where $(a, b) \in R_\alpha$ if and only if R has a transition from *every* concretization of a to *some* concretization of b:

$$R_\alpha^{\exists\exists} \triangleq \{(a,b) \mid \exists s \in \gamma(a) \cdot \exists t \in \gamma(b) \cdot R(s,t)\} \qquad \text{(existential abstraction)}$$
$$R_\alpha^{\forall\exists} \triangleq \{(a,b) \mid \forall s \in \gamma(a) \cdot \exists t \in \gamma(b) \cdot R(s,t)\} \qquad \text{(universal abstraction)}$$

Accordingly, we define $M_\alpha^{\exists\exists} = (S_\alpha, R_\alpha^{\exists\exists})$ and $M_\alpha^{\forall\exists} = (S_\alpha, R_\alpha^{\forall\exists})$ to be the existential and the universal abstractions of M, respectively.

Theorem 2. *Let $\rho \subseteq S \times S_\alpha$ be a relation defined as $\rho \triangleq \{(s, a) \mid \alpha(s) = a\}$. Then M is ρ-bisimilar to $M_\alpha^{\exists\exists}$ if and only if $M_\alpha^{\exists\exists}$ is isomorphic to $M_\alpha^{\forall\exists}$: $M_\alpha^{\exists\exists} \equiv_\rho M \Leftrightarrow M_\alpha^{\exists\exists} = M_\alpha^{\forall\exists}$.*

Symmetry reduction of a structure $M = (S, R)$ with respect to a permutation group G can be seen as a form of abstraction. Formally, let $\mathcal{O} \triangleq \{\theta(s) \mid s \in S\}$, the set of orbits of S, be the abstract statespace, and let an abstraction $\alpha_G : S \to \mathcal{O}$ map each state to its orbit, i.e., $\alpha_G(s) \triangleq \theta(s)$. Under this interpretation, the quotient M^G of M is equivalent to the existential abstraction of M. A mapping between key concepts in abstraction and symmetry reduction is summarized in Table 1.

Using this connection between symmetry and abstraction, we reinterpret Theorem 2 as a necessary and sufficient condition for bisimilarity between M and its quotient M^G. Note that

$$R_\alpha^{\exists\exists} = R_\alpha^{\forall\exists} \text{ if and only if } (s,t) \in R \Rightarrow \forall s' \in \gamma(\alpha(s)) \cdot \exists t' \in \gamma(\alpha(t)) \cdot (s', t') \in R$$

In the context of symmetry reduction, $\gamma(\alpha(s))$, the abstract equivalence class of s, is simply its orbit $\theta(s)$. Furthermore, s and s' share an orbit, i.e., $s' \in \theta(s)$ if and only if there exists a permutation $\sigma \in G$ such that $s' = \sigma(s)$. Combining the above, we obtain the following theorem.

Theorem 3. *Let $M = (S, R)$ be a structure, G be a permutation group acting on S, and $\rho_G \triangleq \{(s, \theta(s)) \mid s \in S\}$. Then, $M \equiv_{\rho_G} M^G$ if and only if*

$$\forall s, t \in S \cdot (s,t) \in R \Rightarrow \forall \sigma \in G \cdot \exists \sigma' \in G \cdot (\sigma(s), \sigma'(t)) \in R \qquad (1)$$

Note that Theorem 3 is a generalization of Theorem 1 since G is no longer required to be an automorphism group for M, and thus M is not necessarily symmetric with respect to G.

Definition 1. *A structure M is* virtually *symmetric with respect to a permutation group G if and only if $M \equiv_{\rho_G} M^G$.*

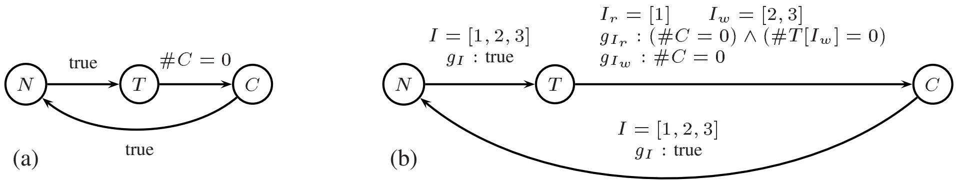

Fig. 1. (a) Synchronization Skeleton for MUTEX. (b) GSST for R&W.

The problem of establishing a necessary and sufficient condition for a quotient M^G to be bisimilar to M has also been addressed by Emerson et al. [9]. Unlike us, they do not use abstraction, but proceed directly to show that M is virtually symmetric with respect to G if and only if it can be "completed" to a structure M' such that M' is both symmetric with respect to G and bisimilar to M. Thus, Theorem 3 provides an alternative (and, in our opinion, much simpler) characterization of virtual symmetry. In the rest of the paper, we show how this new characterization leads to an efficient symbolic model-checking algorithm for a large class of asymmetric systems.

4 Full Virtual Symmetry Identification using Constraints

In this section, we address the problem of identifying full virtual symmetry. Notice that we cannot simply use Condition (1) of Theorem 3 since it requires building the transition relation of $M(P)$, which may not be feasible. We begin by reviewing existing modeling languages for specifying fully symmetric systems in Section 4.1 and then extend them to asymmetric systems in Section 4.2. In Section 4.3, we discuss conditions that ensure that the specified system is fully virtually symmetric, and show how to decide these conditions using constraints derived from the problem description in Section 4.4.

4.1 Modeling Symmetric Systems

Consider an asynchronous composition of n processes $\{P_1, \ldots, P_n\}$ executing a common concurrent program. Each process is specified using a finite directed graph, called a *synchronization skeleton* [6]. Nodes in the graph represent states of the process, and edges, labeled with boolean expressions called *guards*, represent guarded transitions. For example, a synchronization skeleton of a process participating in MUTEX is shown in Figure 1(a). A MUTEX process has 3 states: Non-critical (N), Trying (T), and Critical (C); it can enter states N and T freely, but can only enter the state C if no other process is currently in state C.

When all processes have identical synchronization skeletons, their asynchronous composition can be specified using a single skeleton P. This skeleton can be seen as a template from which skeletons of each individual process are instantiated. Thus, Figure 1(a) is also a synchronization skeleton *template* for MUTEX.

A synchronization skeleton template P defines a compositional structure $M(P)$ in which a (global) transition results from a local transition of some process. For example, in the three-process MUTEX, $M(P)$ has a transition from (N, N, T) to (N, N, C) because the third process, P_3, can move from T to C.

Note that when each transition guard in P is invariant under a permutation of process indices, the structure $M(P)$ is unchanged by any permutation of process indices,

Basic Elements	Predicates on Total Counters
$\forall i \cdot l_i = L, \forall i \cdot l_i \neq L$	$\#L = n, \#L = 0$
$\exists i \cdot l_i = L, \exists i \cdot l_i \neq L$	$\#L \geq 1, \#L \leq n-1$
$\exists i \neq j \cdot l_i = L \wedge l_j = L$	$\#L \geq 2$

Table 2. Basic guard elements for ensuring full symmetry.

i.e., it is fully symmetric [11]. For example, the three-process MUTEX is fully symmetric since if the guard $(\#C = 0)$ is true in a state s, it is also true in a state $\sigma(s)$ for any permutation $\sigma \in Sym([1, 2, 3])$. Symmetry reduction of a fully symmetric system can yield an exponential reduction in the number of states. In practice, full symmetry of a synchronization skeleton is ensured by restricting basic elements of the guards to the ones shown in the left column of Table 2, where $l_i = L$ is true in a state s if the ith process is in a state L, i.e., $s(i) = L$. The basic elements can be equivalently expressed using total counters, as shown in the right column of Table 2 [11].

4.2 Modeling Asymmetric Systems

In this paper, we are interested in applying symmetry reduction to asymmetric systems composed of many similar, but not identical processes, such as R&W mentioned in Section 1. In this case, since the condition for entering the critical section is different between the two groups of processes (writers have a higher priority than readers), the system cannot be modeled by a single synchronization skeleton. Thus, for such asymmetric systems, we need both a more general modeling formalism, and an approach to identify whether the system is fully virtually symmetric. To address the first problem, we define a *generalized synchronization skeleton template*.

Definition 2. *A* generalized synchronization skeleton template *(GSST) for an asynchronous system with n processes is a tuple $P = (\mathcal{L}, \mathcal{R}, I, \tau)$, where $\mathcal{L}$ is a finite set of (local) states, $\mathcal{R} \subseteq \mathcal{L} \times \mathcal{L}$ is a (local) transition relation, $I = [1..n]$ is the index set, and $\tau : \mathcal{R} \to [I \to G]$ is a labeling function that labels each transition with a guard for each process. Here, $G : \mathcal{L}^n \to \{true, false\}$ is a set of transition guards.*

We assume that for any local transition $u \to v \in \mathcal{R}$, $u \neq v$, i.e., no self-loops are allowed in a GSST.

Definition 3. *A GSST $P = (\mathcal{L}, \mathcal{R}, I, \tau)$ defines an asynchronous structure $M(P) = (S, R)$, where $S = \mathcal{L}^{|I|}$ is the global statespace, and $R \subseteq S \times S$ is the global transition relation defined as follows:*

(a) *for any local transition $u \to v \in \mathcal{R}$,*
$R_{u\to v}(s, t) \triangleq \exists i \in I \cdot (s(i) = u \wedge t(i) = v \wedge s \models \tau(u \to v)(i) \wedge \forall j \neq i \cdot s(j) = t(j))$

(b) $R \triangleq \bigcup_{r \in \mathcal{R}} R_r$.

Intuitively, $R_{u\to v}$ is the set of all global transitions resulting from some process changing its state from u to v. We say that $s \to t \in R$ is a result of firing a local transition $u \to v$ if $s \to t$ is in $R_{u\to v}$.

For a local transition $r \in \mathcal{R}$, the labeling function $\tau : \mathcal{R} \to [I \to G]$ can be seen as: (a) a partition $\Pi_r = \{I_1, \ldots, I_d\}$ of processes into process groups, (b) an index mapping function $\pi : I \to \Pi_r$, and (c) a function $\eta : \Pi_r \to G$ assigning a guard to each process group, i.e., for any $i \in I$, $\tau(r)(i) = \eta(\pi(i))$. For example, in the GSST for the

1Reader-2Writers R&W shown in Figure 1(b), the guards for the local transition $T \rightarrow C$ are described by partitioning the processes into two groups: $I_r = \{P_1\}$ (readers) and $I_w = \{P_2, P_3\}$ (writers). Readers have the guard $g_{I_r} : (\#C = 0) \wedge (\#T[I_w] = 0)$, and writers $g_{I_w} : \#C = 0$. Note that this allows us to model not only the static process partitioning, i.e., $\forall r, r' \in \mathcal{R} \cdot \Pi_r = \Pi_{r'}$, but a dynamic one as well, that is, processes can be divided into different groups at different local transitions.

Motivated by R&W, we restrict our attention to a counter-based syntax of guards. Formally, a guard for a transition $u \rightarrow v$ is a boolean combination of *group counter constraints* on the local state u, i.e., $\#u[I_k] \bowtie b$, or a *total counter constraints* on any local states, i.e., $(\sum_i \#L_i) \bowtie b$, where b is a positive integer, and $\bowtie$ is one of $\{\leq, \geq, =\}$. For example, in Figure 1(b), $\#C = 0$ means no process is currently in the local state C, whereas $\#T[I_w] = 0$ means that no *writer* process is currently in T.

4.3 Full Virtual Symmetry in Asynchronous Structure

In this section, we show how to identify whether a system specified by a GSST is fully virtually symmetric.

Let P be a GSST and r be a transition in P. If all processes at r belong to the same group, i.e., $|\Pi_r| = 1$, then the transition guard is defined on total counters and is independent of any permutation of process indices. Furthermore, if this is the case for all transitions in P, then P is just a synchronization skeleton, and the underlying structure $M(P)$ is fully symmetric (see Section 4.1). In general, when P contains a transition r with $|\Pi_r| > 1$, even restricting guards to just total counter constraints is not sufficient to ensure that $M(P)$ is fully virtually symmetric. For example, consider the GSST shown in Figure 1(b) and assume that we change the guard g_{I_r} of the transition $T \rightarrow C$ to $(\#C = 0) \wedge (\#T = 2)$. In this case, $M(P)$ contains a global transition from $s = (N, N, T)$ to $t = (N, N, C)$ corresponding to the process P_3 entering state C. Let $\sigma \in Sym(I)$ be a permutation that switches process indices 1 and 3. Then, the only two states reachable from $\sigma(s) = (T, N, N)$ are $t_1 = (T, T, N)$ and $t_2 = (T, N, T)$. Since neither t_1 nor t_2 can be obtained by applying a permutation $\sigma' \in Sym(I)$ to t, transitions in the form $\sigma(s) \rightarrow \sigma'(t)$ are *not* in $M(P)$ for any permutation σ'; hence, $M(P)$ is not fully virtually symmetric.

As illustrated by the example above, lack of regularity in asymmetric systems makes it difficult to ensure full virtual symmetry through a syntactic restriction. Therefore, we seek an algorithmic way to identify symmetry. As mentioned before, we cannot simply use Condition (1) of Theorem 3 since it requires building the transition relation of $M(P)$.

Notice that in our example, full virtual symmetry is broken at a global transition resulting from firing a local transition where the processes are partitioned into several groups. We generalize from this example and show that virtual symmetry of a structure is equivalent to virtual symmetry of each transition relation subset defined by a local transition. This allows us to decompose the problem of identifying virtual symmetry of a system along *local* transitions. Formally, we establish the following theorem.

Theorem 4. *Given a GSST $P = (\mathcal{L}, \mathcal{R}, I, \tau)$ and a permutation group $G \subseteq Sym(I)$, the structure $M(P) = (S, R)$ is virtually symmetric with respect to G if and only* *if each transition relation subset R_r is virtually symmetric with respect to G, where $R \triangleq \bigcup_{r \in \mathcal{R}} R_r$.*

The "if" direction of Theorem 4 is trivial: a union of virtually symmetric transition relations is virtually symmetric. For the "only if" direction, by Theorem 2, we know that R is virtually symmetric with respect to G if and only if $R^{\exists\exists}_{\alpha_G} = R^{\forall\exists}_{\alpha_G}$. With the aid of Theorem 5 given below and obtained from the perspective of abstraction, we show that $(R_r)^{\exists\exists}_{\alpha_G} = (R_r)^{\forall\exists}_{\alpha_G}$ for each R_r, i.e., R_r is virtually symmetric with respect to G.

Let $M' = (S', R')$ be a structure, and $\alpha : S' \rightarrow S_\alpha$ be an abstraction function. We define a restriction of R' to a pair of abstract states (a, b) as

$$R'_{|(a,b)} \triangleq \{(s,t) \in R' \mid s \in \gamma(a) \wedge t \in \gamma(b)\}$$

Note that $R' = \bigcup_{a,b \in S_\alpha} R'_{|(a,b)}$, and the universal and the existential abstractions of R' coincide if and only if they coincide for each $R'_{|(a,b)}$. The following theorem generalizes this observation.

Theorem 5. *Let $M' = (S', R')$ be a structure, $\alpha : S' \rightarrow S_\alpha$ be an abstraction function, and $R' = \bigcup_{i \in [1..k]} R'_i$ such that $\forall i \in [1..k] \cdot \exists D \subseteq S' \times S' \cdot R'_i = \bigcup_{(s,t) \in D} R'_{|(\alpha(s),\alpha(t))}$. Then, $(R')^{\forall\exists}_\alpha = (R')^{\exists\exists}_\alpha \Leftrightarrow \forall i \in [1..k] \cdot (R'_i)^{\forall\exists}_\alpha = (R'_i)^{\exists\exists}_\alpha$.*

Recall that in the context of symmetry reduction, $\alpha_G(s)$ is equivalent to $\theta(s)$ (see Table 1). We claim that each R_r satisfies the precondition of Theorem 5 by showing that $R_r = \bigcup_{(s,t) \in R_r} R_{|(\theta(s),\theta(t))}$. That is, we need to show that if a transition $s \rightarrow t$ is a result of firing a local transition r, then for any permutations $\sigma, \sigma' \in G$, a transition $\sigma(s) \rightarrow \sigma(t)$ is a result of firing r as well. This holds from the following observations: (a) two states s_1 and s_2 share an orbit only if they agree on total counters, and (b) a global transition $s \rightarrow t$ is a result of firing a local transition $u \rightarrow v$ if and only if $\#u$ at s is one more than that at t, $\#v$ at s is one less than that at t, and the total counters of other local states at s and t are the same. For example, consider two global transitions $s \rightarrow t$ and $s' \rightarrow t'$ such that $s' \in \theta(s)$, and $t' \in \theta(t)$. Since s and t agree with s' and t', respectively, on total counters, then if $s \rightarrow t$ is in R_r, $s' \rightarrow t'$ must be in R_r as well. Therefore, virtual symmetry of R implies virtual symmetry of each R_r. This concludes the proof of Theorem 4.

When G is the full symmetry group $Sym(I)$, Theorem 4 can be simplified further since here two states share an orbit *if and only if* they agree on total counters. Note that if R_r is fully virtually symmetric, i.e., $(R_r)^{\forall\exists}_{\alpha_G} = (R_r)^{\exists\exists}_{\alpha_G}$, then $Dom(R_r)$ contains its orbit $\theta(Dom(R_r))$, which follows from the definitions of existential and universal abstractions. On the other hand, if $Dom(R_r)$ contains $\theta(Dom(R_r))$, then for any pair of states s and s' in the same orbit, if $s \rightarrow t$ is in R_r for some state t, then there exists a state t' such that $s' \rightarrow t'$ is in R_r. Furthermore, t and t' agree on total counters, and thus belong to the same orbit. Hence, by Theorem 3, R_r is fully virtually symmetric. Since $\theta(Dom(R_r))$ always contains $Dom(R_r)$, we obtain the following theorem.

Theorem 6. *Given a GSST $P = (\mathcal{L}, \mathcal{R}, I, \tau)$, the structure $M(P) = (S, R)$ is fully virtually symmetric if and only if $\forall r \in \mathcal{R} \cdot \theta(Dom(R_r)) = Dom(R_r)$.*

Thus, we have reduced the problem of checking virtual symmetry of R, a global property of the entire system, to a local property of each transition subset R_r.

Constraints	Meaning
$0 \leq \#N[I_r]$ $0 \leq \#T[I_r]$ $0 \leq \#C[I_r]$ $0 \leq \#N[I_w]$ $0 \leq \#T[I_w]$ $0 \leq \#C[I_w]$	each group counter is a positive integer
$\#N[I_r] + \#T[I_r] + \#C[I_r] = 1$ $\#N[I_w] + \#T[I_w] + \#C[I_w] = 2$	there is one reader process and two writer processes

Table 3. Invariant for the 1Reader-2Writers R&W.

4.4 Constraint-Based Identification of Full Virtual Symmetry

In this section, we present a technique for identifying full virtual symmetry based on Theorem 6. Specifically, we construct Presburger formulas representing sets of states directly from the description of the GSST.

By Theorem 4, checking whether a structure $M(P)$ is fully virtually symmetric is equivalent to checking whether R_r is fully virtually symmetric for each local transition r of the GSST P. Note that if all processes belong to the same group at a local transition r, i.e., $|\Pi_r| = 1$, then R_r is fully symmetric and no check is required. Otherwise, when $|\Pi_r| > 1$, by Theorem 6, we need to check whether the domain of R_r, $Dom(R_r)$, is equal to its orbit, $\theta(Dom(R_r))$. In this section, we show that both $Dom(R_r)$ and $\theta(Dom(R_r))$ can be represented by Presburger formulas and their equivalence can be reduced to checking satisfiability of a Quantifier Free Presburger (QFP) formula.

We illustrate the procedure on the $T \rightarrow C$ transition of the R&W problem whose GSST is shown in Figure 1(b). The counter-based syntax of the guards provides a compact representation of a set of states in the structure $M(P)$ using Presburger formulas on group counters. The formula $\varphi_{T\rightarrow C}$ representing $Dom(R_{T\rightarrow C})$ is constructed based on the transition guards in the GSST as follows. According to the interleaving semantics, a state s is in $Dom(R_{T\rightarrow C})$ if and only if either a reader or a writer process can move from T to C at s. In the first case, s must satisfy the guard g_{I_r}, and since the current local state of the reader process is T, s satisfies $g_{I_r} \wedge \#T[I_r] \geq 1$; similarly, in the second case, s satisfies $g_{I_w} \wedge \#T[I_w] \geq 1$. Therefore, $Dom(R_{T\rightarrow C})$ can be represented by the formula $\varphi_{T\rightarrow C} = \varphi_{T\rightarrow C,I_r} \vee \varphi_{T\rightarrow C,I_w}$, where

$$\varphi_{T\rightarrow C,I_r} \triangleq g_{I_r} \wedge \#T[I_r] \geq 1 \wedge inv_{T\rightarrow C} \quad \varphi_{T\rightarrow C,I_w} \triangleq g_{I_w} \wedge \#T[I_w] \geq 1 \wedge inv_{T\rightarrow C}$$

and the invariant $inv_{T\rightarrow C}$, defined as the conjunction of the constraints in the left column of Table 3, represents the statespace of the system. Note that $\varphi_{T\rightarrow C}$ is still defined only on group counters since $\#C$ is equivalent to $\#C[I_r] + \#C[I_w]$. In general, for a local transition r, the formula φ_r representing $Dom(R_r)$ is a disjunction of formulas representing subsets of $Dom(R_r)$ with respect to each process group.

We now show how to derive a formula $\tilde{\varphi}_r$ representing $\theta(Dom(R_r))$ from φ_r. For simplicity, assume that P contains only two local states, X and Y, and the processes are partitioned into two groups. Let $Dom(R_r)$ and the invariant of the statespace be represented by $\varphi_r(X_1, X_2, Y_1, Y_2)$ and $inv_r(X_1, X_2, Y_1, Y_2)$, respectively. Then $\tilde{\varphi}_r$ representing $\theta(Dom(R_{T\rightarrow C}))$ is defined as

$$\begin{aligned}\tilde{\varphi}_r(X_1, X_2, Y_1, Y_2) \triangleq \exists X_1', X_2', Y_1', Y_2' \cdot\ & (inv_r(X_1, X_2, Y_1, Y_2) \wedge \varphi_r(X_1', X_2', Y_1', Y_2') \\ & \wedge X_1 + X_2 = X_1' + X_2' \wedge Y_1 + Y_2 = Y_1' + Y_2')\end{aligned}$$

That is, a state s satisfies $\tilde{\varphi}_r$ if and only if there exists a state s' satisfying φ_r ($s' \in Dom(R_r)$) and s and s' agree on total counters, i.e., they are in the same orbit. Since $Dom(R_r)$ is a subset of $\theta(Dom(R_r))$, $Dom(R_r) = \theta(Dom(R_r))$ if and only if the sentence $\psi = \exists X_1, X_2, Y_1, Y_2 \cdot (\tilde{\varphi}_r \wedge \neg\varphi_r)$ is unsatisfiable. Since ψ contains only existential quantifiers, this is equivalent to unsatisfiability of a QFP formula obtained from ψ by removing all quantifiers, which can be checked using any existing decision procedure for QFP [3, 16, 17].

Note that while the satisfiability problem of a Presburger formula has a worst-case super-exponential complexity, satisfiability of a QFP formula is NP-complete [14]. Furthermore, the number of local transitions in a GSST that need to be checked is expected to be small, since we are interested in asynchronous systems in which processes are relatively similar to one another. Indeed, if the processes differ significantly, it does not seem appropriate to consider full virtual symmetry at all. In practice, the structure of the guards often leads to further optimizations of the decision procedure. As illustrated by experiments in Section 6, full virtual symmetry can be identified efficiently when the guards are defined on a small number of local states.

5 Counter Abstraction for Full Virtual Symmetry

The naive way of constructing a symmetry-reduced quotient structure requires a representative function for choosing a state as the unique representative from each orbit [7, 5]. The abstract transition relation is then defined on the set of representatives. For symbolic model-checking, computation of the representative function requires building an orbit relation which, for many groups, including the full symmetry group, has a BDD representation that is exponential in the minimum of the number of processes and the number of local states in each process [7], decreasing the effectiveness of symbolic model-checking.

An alternative is to use *generic representatives* (or a counter abstraction) technique proposed by Emerson at el. [11, 12], which avoids building the orbit relation. As we have seen before, under full symmetry group, states in the same orbit agree on all total counters. Thus, each orbit can be uniquely represented by values of these counters. For example, in the three-process MUTEX, the orbit $\{(N,T,T), (T,N,T), (T,T,N)\}$ is represented by a tuple $(1, 2, 0)$ which corresponds to the counters of states N, T and C. In this section, we extend the counter-based abstraction technique to handle fully virtually symmetric structure specified by a GSST. The key idea is that instead of using the orbit relation, a structure isomorphic to the quotient structure is constructed on the statespace of total counters directly from the GSST.

For the rest of this section, let $P = (\mathcal{L}, \mathcal{R}, I, \tau)$ be a GSST of a fully virtually symmetric system with local states $\mathcal{L} = \{L_1, \ldots, L_m\}$ and process indices $I = [1..n]$. A *counter abstraction* $\alpha : S \to S_\alpha$ on the structure $M(P) = (S, R)$ is constructed using a set of assignments to a vector $\mathbf{x} = (x_1, \ldots, x_m)$ of m counter variables ranging over $[0..n]$. Each variable x_i corresponds to a total counter $\#L_i$ of a local state L_i. Since there are n processes, the sum of the values of $\mathbf{x}$ must always equal n. Therefore,

$$S_\alpha \triangleq \{(c_1, \ldots, c_m) \in [0..n]^m \mid \textstyle\sum_{i=1}^{m} c_i = n\}$$

The abstraction function $\alpha : S \to S_\alpha$ maps a state $s \in S$ to an abstract state $a \in S_\alpha$ if and only if for each $i \in I$, $a(i)$ equals $\#L_i(s)$. The concretization function $\gamma : S_\alpha \to 2^S$ maps an abstract state a to an orbit θ where states in θ agree with a on total counters. In what follows, let R_α denote the existential abstraction of R with respect to α.

Theorem 7. *Given a GSST P and a counter abstraction α, the abstract structure $M(P)_\alpha = (S_\alpha\ , R_\alpha)$ is isomorphic to the quotient structure $M(P)^{Sym(I)} = (S^{Sym(I)}, R^{Sym(I)})$ via a bijection $h : S_\alpha \to S^{Sym(I)}$, where $\forall s \in S \cdot h(\alpha(s)) \triangleq \theta(s)$.*

The above definition of $M(P)_\alpha$ guarantees that the abstract transition relation R_α can be constructed directly from P for a fully virtually symmetric system. Since existential abstraction distributes over union, and $R = \bigcup_{r \in \mathcal{R}} R_r$ by Definition 3, it follows that $R_\alpha = \bigcup_{r \in \mathcal{R}} (R_r)_\alpha$. Therefore, we only need to show how to construct $(R_r)_\alpha$ for a local transition r.

We start by illustrating the construction in the case of an unguarded local transition r. If r is of the form $L_i \to L_j$, then r can be fired from a global state s if and only if s contains a process whose current state is L_i; in other words, $Dom(R_r)$ is $\#L_i \geq 1$. Furthermore, if $s \to t$ is in R_r, then the counters $\#L_i$ and $\#L_j$ at t are one less and one more than those at s, respectively. From the definition of existential abstraction, for any abstract states a and b, a transition $a \to b$ is in $(R_r)_\alpha$ if and only if $s \to t \in R_r$ for some $s \in \gamma(a)$ and $t \in \gamma(b)$. Therefore,

$$(R_r)_\alpha \ \equiv\ x_i \geq 1 \wedge (x_i := x_i - 1;\ x_j := x_j + 1)$$

which is a formula over counter variables. Generalizing from this example, we obtain that for every local transition r of the form $L_i \to L_j$,

$$(R_r)_\alpha \ \equiv\ g_r \wedge (x_i := x_i - 1;\ x_j := x_j + 1)$$

where g_r is a formula defined over counter variables $\mathbf{x}$ representing the "existential" abstraction of $Dom(R_r)$. Specifically,

$$a \models g_r \ \Leftrightarrow\ \exists s \in \gamma(a) \cdot s \in Dom(R_r)$$

Since $M(P)_\alpha$ is isomorphic to the quotient structure, the above construction allows us to combine symmetry reduction and symbolic model-checking without building the orbit relation. The only remaining problem is the construction of the formula g_r for an arbitrary local transition r, and in the rest of this section, we show how to do this for cases where r is guarded by (a) a single guard on total counters, (b) multiple guards on total counters, and (c) multiple guards on group counters of the source state of r and arbitrary total counters.

Case (a). Let r be a local transition $L_i \to L_j$. Suppose r is guarded by a single guard g, i.e., $|\Pi_r| = 1$. Then $Dom(R_r)$ can be represented by $\psi_r = (\#L_i \geq 1 \wedge g)$, i.e., $s \in Dom(R_r)$ if there is at least one process at s in local state L_i and s satisfies g. Let $sub(\psi_r)$ denote a formula obtained from ψ_r by replacing each occurrence of a total counter with its corresponding counter variable. For example, $sub(\#L_i \geq 0) = (x_i \geq 0)$ and $sub(\#L_i \geq 1 \wedge \#L_j \leq 3) = (x_i \geq 1 \wedge x_j \leq 3)$. Since g contains only total counter constraints, we define $g_r \triangleq sub(\#L_i \geq 1 \wedge g)$. Note that this is procedure constructs a counter abstraction for a fully symmetric synchronization skeleton, and is effectively equivalent to the *generic representatives* approach of Emerson and Trefler [11].

Case (b). Suppose that r is guarded by multiple guards, i.e., $|\Pi_r| = d > 1$, but each guard is expressed using only total counters. In this case, $Dom(R_r)$ is represented by $\psi_r = \bigvee_{k \in [1..d]} (\#L_i[I_k] \geq 1 \wedge g_{I_k})$, where g_{I_k} is the guard for the process group

I_k. Since ψ_r depends on group counters, we cannot simply define g_r to be $sub(\psi_r)$. However, R_r is fully virtually symmetric, so $Dom(R_r) = \theta(Dom(R_r))$ by Theorem 6, and $\theta(Dom(R_r))$ is representable by $\tilde{\psi_r} = (\#L_i \geq 1 \wedge (\bigvee_{k\in[1..d]} g_{I_k}))$. Thus, we define $g_r \triangleq sub(\tilde{\psi_r})$.

Case (c). Finally, we look at the case where the guards of r depend on group counters. In this case, $\tilde{\psi_r}$ defined above still contains group counters. However, this problem can be solved for cases where group counters in guards for a transition $r : L_i \to L_j$ are defined only over L_i.

First, let $Q \subseteq S$ be some non-empty set of states given by some formula ψ defined only on group counters of L_i. That is, $\psi = \bigwedge_{k\in[1..d]}(min_k \leq \#L_i[I_k] \leq max_k)$ where $\{min_k\}$ and $\{max_k\}$ are positive integers. Then the orbit $\theta(Q)$ under $Sym(I)$ is given by the formula

$\tilde{\psi} = (min \leq \#L_i \leq max)$ where $min \triangleq \sum_{k\in[1..d]} min_k$ and $max \triangleq \sum_{k\in[1..d]} max_k$

For example, suppose there are only two local states, L_1 and L_2, $d = 2$, and Q is given by $\psi = (1 \leq \#L_1[I_1] \leq 4) \wedge (1 \leq \#L_1[I_2] \leq 4)$. Then $\theta(Q)$ is $\tilde{\psi} = (2 \leq \#L_1 \leq 8)$ since for any state s in S satisfying $\tilde{\psi}$ there exists a state s' in S satisfying ψ such that s and s' agree on total counters of L_1 and L_2, i.e., they are in the same orbit. Furthermore, if Q is encoded by a conjunction $\psi^t \wedge \psi^g$, where ψ^t and ψ^g are defined only on total and group counters, respectively, then the orbit of Q is given by $\psi^t \wedge \tilde{\psi^g}$.

Second, suppose a guard g_{I_k} contains group counter constraints. Let $Dom(R_r)_{I_k}$ denote the subset of $Dom(R_r)$ containing states in which the local transition r of some process in the group I_k can be fired. If the formula ψ_{r,I_k} representing $Dom(R_r)_{I_k}$ can be decomposed as $\psi_{r,I_k} = \psi^t_{r,I_k} \wedge \psi^g_{r,I_k}$, then a total counter formula representing $\theta(Dom(R_r)_{I_k})$ is computed as described above. Otherwise, ψ_{r,I_k} can be converted to a DNF, and formulas corresponding to the orbit of each clause computed as above. Since $Dom(R_r) = \bigcup_{k\in[1..d]} Dom(R_r)_{I_k}$, and θ distributes over union, i.e., $\theta(Q_1 \cup Q_2) = \theta(Q_1) \cup \theta(Q_2)$, we can define $\tilde{\psi_r}$ representing $\theta(Dom(R_r))$ as a disjunction of the clause formulas. Finally, $\tilde{\psi_r}$ depends only on total counters; thus, we define g_r to be $sub(\tilde{\psi_r})$.

For example, the domain of the transition $T \to C$ of the R&W shown in Figure 1(b), is the union of the domain for the readers and that of the writers. For readers,

$$\begin{aligned} Dom(R_{T\to C})_{I_r} &\equiv \#T[I_r] >= 1 \wedge \#T[I_w] = 0 \wedge \#C = 0 \\ &\equiv \#T[I_r] = 1 \wedge \#T[I_w] = 0 \wedge \#C = 0 \end{aligned}$$

since there is only one reader. Using only total counters, the orbit $\theta(Dom(R_{T\to C})_{I_r})$ is represented by $\tilde{\psi_r} = (\#T = 1 \wedge \#C = 0)$. Similarly, for the writers,

$$Dom(R_{T\to C})_{I_w} \equiv \#T[I_w] \geq 1 \wedge \#C = 0$$

and the orbit $\theta(Dom(R_{T\to C})_{I_w})$ is represented by $\tilde{\psi_w} = (\#T \geq 1 \wedge \#C = 0)$. Finally, $g_{T\to C}$ is defined by $sub(\tilde{\psi_r} \vee \tilde{\psi_w}) = (\#T \geq 1 \wedge \#C = 0)$.

6 Experiments

In this section, we report on experiments of identifying full virtual symmetry and performing counter abstraction-based symbolic model checking on two examples: generalized R&W (GR&W) and asymmetric sharing of resources (ASR) [9]. We used the

Omega library [16] to check for full virtual symmetry as described in Section 4, and NuSMV [4] as the model-checker for both the direct and the counter abstraction-based analysis.

In GR&W, we assumed that each process has m local states $\{L_1, \ldots, L_m\}$, where L_m represents the critical section. Each process can move from L_i to L_{i+1} ($i \in [1..m\text{-}2]$) and return from L_m to L_1 freely. The processes are partitioned into d groups, each of size q, based on their priorities: a process cannot access the critical section if another process with higher priority is waiting for it. The property we verified was $AG(\#L_m \leq 1)$. The second example, ASR, is motivated by the drinking philosophers problem [9]. It exhibits full virtual symmetry induced by the asymmetric sharing of resources, where n processes have different permissions to access r critical resources, and the number of processes that can be waiting for each resource and using it is bounded. We checked whether it is possible for all critical resources to be used at the same time, i.e., $EF\big(\bigwedge_{i \in [1..r]}(\#C_i > 0)\big)$. The experiments were performed on a Sun Fire V440 server (4@1.3GHz, USPARC3i, 16384M). The results of the direct (*NuSMV*) and the counter abstraction-based (*Symmetry Reduction with Counter Abstraction*) analysis are summarized in Table 4, where dashes indicate that verification did not complete either due to memory or time limits. Where appropriate, we separate the checking time into identifying symmetry (*CkSym*) and checking the resulting reduced model (*ModelCk*). For ASR, we also reported the results of computing the set of reachable states first before evaluating the property (the "-f" option of NuSMV).

The experiments show that counter abstraction provides a significant reduction in both memory and CPU usage. Memory usage grows slowly with the number of processes, which indicates that the method is applicable for systems comprised of a large number of processes.

In these examples, the time it took to identify full virtual symmetry was relatively small. One reason is that the guards depend only on a small number of process groups and local states. Otherwise, more specialized solvers may be useful. For example, identifying symmetry of GR&W with d = 100 and q = 20 took us many hours with the Omega library and only 17 seconds with the pseudo-Boolean solver (PBS) [1].

7 Conclusion and Related Work

The problem of exploiting symmetry reduction in model checking has been studied by many researchers, e.g., [2, 7, 10, 13]. To extend symmetry reduction to asymmetric systems, Emerson and his colleagues first proposed "looser" notions of *near* symmetry and *rough* symmetry [11], and finally virtual symmetry [9] which subsumes the previous two. In this paper, we give an alternative (and simpler) characterization of virtual symmetry from the perspective of abstraction.

The problem of identifying full symmetry has been avoided by imposing restrictions on the specification language [11–13]. However, lack of regularity in asymmetric systems makes it difficult to enforce full virtual symmetry through such a syntactic restriction. Emerson et al. proposed a combinatorial condition for checking virtual symmetry based on counting the missing transitions [9], which seems to require the construction of the transition relation. With our characterization of virtual symmetry, we avoid this problem by checking satisfiability of a QFP formula built from the system description.

	Parameter	NuSMV			Symmetry Reduction with Counter Abstraction				
		BDD Nodes Allocated	Mem. (MB)	Time (sec.)	BDD Nodes Allocated	Mem. (MB)	Time (sec.) CkSym	ModelCk	Total
Generalized R&W	d (q=20, m=10)								
	5	51,778,281	931	241	25,146	7	0.07	0.27	0.34
	10	-	-	-	31,772	8	0.83	0.53	1.36
	15	-	-	-	38,927	8	5.09	1.26	6.35
	m (d=5, q=20)								
	10	51,778,281	931	241	25,146	7	0.07	0.27	0.34
	20	121,392,365	2,041	837	130,891	10	0.07	0.59	0.66
	30	-	-	-	379,336	14	0.07	1.35	1.42
	q (d=10, m=20)								
	10	121,408,515	2,040	742	131,010	10	0.80	0.58	1.38
	30	-	-	-	187,469	12	0.81	24.14	24.95
	50	-	-	-	195,653	13	0.75	67.21	67.96
Asymmetric Sharing of Resources	n (r=2)								
	20	597,911	18	2.11	77,885	8	0.10	0.78	0.88
	30	2,443,114	51	8.19	179,389	10	0.10	1.74	1.84
	40	8,151,508	151	30.74	427,075	14	0.10	4.35	4.45
	80	57,163,279	1,001	2928.81	289,566	18	0.10	36.83	36.93
	n (r=3)								
	20	1,896,771	43	10.39	182,799	10	0.15	1.55	1.70
	30	11,503,014	216	78.46	403,628	14	0.15	3.64	3.79
	40	44,877,253	782	43108.92	390,715	17	0.15	9.68	9.83
	80	-	-	-	420,347	20	0.15	80.61	80.76
	n (r=5)								
	40	-	-	-	67,060	19	0.30	28.31	28.61
	80	-	-	-	342,060	39	0.30	279.89	280.19
	n (r=10)								
	40	-	-	-	484,260	48	3.00	251.87	254.87
	80	-	-	-	671,318	153	3.00	1409.53	1412.53
Asym. Sharing of Resources (reachable states)	n (r=2)								
	20	635,791	19	2.24	5,575	6.9	0.10	0.13	0.23
	30	2,557,272	53	8.91	6,589	6.9	0.10	0.14	0.24
	40	8,543,329	159	34.47	10,165	7	0.10	0.15	0.25
	80	57,375,594	1,006	528.25	18,611	7.2	0.10	0.25	0.35
	n (r=3)								
	20	1,927,302	43	8.07	11,634	7	0.15	0.15	0.30
	30	11,591,335	220	61.14	14,616	7.1	0.15	0.18	0.33
	40	42,633,638	805	1614.32	21,647	7.3	0.15	0.21	0.36
	80	-	-	-	38,913	7.7	0.15	0.39	0.54
	n (r=5)								
	40	-	-	-	71,925	8.2	0.30	0.49	0.79
	80	-	-	-	133,034	9.5	0.30	1.03	1.33
	n (r=10)								
	40	-	-	-	394,722	14	3.00	2.55	5.55
	80	-	-	-	404,477	18	3.00	6.13	9.13

Table 4. Experimental results for generalized R&W and asymmetric sharing of resources.

To combine full symmetry reduction and symbolic model-checking, Emerson et al. [11] proposed a *generic representatives* technique, also known as a counter abstraction [15]. In this paper, we have extended this technique to fully virtually symmetric systems. The generic representatives technique was later applied to fully symmetric systems on processes communicating via shared variables [12], and the experiments show that it is superior to other methods, such as multiple representatives [7]. We plan to do the same for fully virtually symmetric systems in future.

We believe that our techniques have a potential to significantly increase the scope of systems to which symmetry reduction can be effectively applied. Note that our work assumed that group counters occurring in a guard are defined only on the source state (see Section 5). While this did not pose a problem for examples we have tried, we do not know what the consequences of this restriction are, and would like to explore these further.

References

1. F. Aloul, A. Ramani, I. Markov, and K. Sakallah. “PBS: A Backtrack Search Pseudo-Boolean Solver”. In *SAT’02*, pp. 346–353, 2002.
2. S. Barner and O. Grumberg. Combining symmetry reduction and under-approximation for symbolic model checking. In *CAV’02*, vol. 2404 of *LNCS*, pp. 93–106, 2002.
3. C. Barrett and S. Berezin. “CVC Lite: A New Implementation of the Cooperating Validity Checker”. In *CAV’04*, vol. 3114 of *LNCS*, pp. 515–518, 2004.
4. A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri. “NuSMV: a new Symbolic Model Verifier”. In *CAV’99*, vol. 1633 of *LNCS*, pp. 495–499, 1999.
5. E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.
6. E.M. Clarke and E.A. Emerson. “Design and Synthesis of Synchronization Skeletons for Branching Time Temporal Logic”. In *Logic of Programs*, vol. 131 of *LNCS*, 1981.
7. E. Clarke, R. Enders, T. Filkorn, , and S. Jha. “Exploiting Symmetry in Temporal Logic Model Checking”. *FMSD*, 9(1-2):77–104, 1996.
8. D. Dams, R. Gerth, and O. Grumberg. “Abstract Interpretation of Reactive Systems”. *ACM TOPLAS*, 2(19):253–291, 1997.
9. E. Emerson, J. Havlicek, and R. Trefler. “Virtual Symmetry Reduction”. In *LICS’00*, pp. 121–131, 2000.
10. E. Emerson and A. Sistla. “Symmetry and Model Checking”. *FMSD*, 9(1-2):105–131, 1996.
11. E. Emerson and R. Trefler. “From Asymmetry to Full Symmetry: New Techniques for Symmetry Reduction in Model Checking”. In *CHARME’99*, vol. 1703 of *LNCS*, pp. 142–157, 1999.
12. E. Emerson and T. Wahl. “On Combining Symmetry Reduction and Symbolic Representation for Efficient Model Checking”. In *CHARME’03*, vol. 2860 of *LNCS*, pp. 216–230, 2003.
13. C. Ip and D. Dill. “Better Verification Through Symmetry”. *FMSD*, 9(1-2):41–75, 1996.
14. C. Papadimitriou. “On the Complexity of Integer Programming”. *J. ACM*, 28(4):765–768, 1981.
15. A. Pnueli, J. Xu, and L. Zuck. Liveness with (0, 1, ∞)-counter abstraction. In *CAV’02*, vol. 2404 of *LNCS*, pp. 107–122, 2002.
16. W. Pugh. “The Omega Test: A Fast and Practical Integer Programming Algorithm for Dependence Analysis”. *Comm. of the ACM*, August 1992.
17. P. Wolper and B. Boigelot. “An Automata-Theoretic Approach to Presburger Arithmetic Constraints”. In *SAS’95*, vol. 1785 of *LNCS*, pp. 21–32, 1995.