Local Symmetry and Compositional Verification

Kedar S. Namjoshi[1] and Richard J. Trefler[2⋆]

[1] Bell Laboratories, Alcatel-Lucent kedar@research.bell-labs.com
[2] University of Waterloo, trefler@cs.uwaterloo.ca

Abstract. This work considers concurrent programs formed of processes connected by an underlying network. The symmetries of the network may be used to reduce the state space of the program, by grouping together similar global states. This can result in an exponential reduction for highly symmetric networks, but it is much less effective for many networks, such as rings, which have limited global symmetry. We focus instead on the *local symmetries* in a network and show that they can be used to significantly reduce the complexity of *compositional reasoning*. Local symmetries are represented by a *symmetry groupoid*, a generalization of a symmetry group. Certain sub-groupoids induce quotient networks which are equivalent to the original for the purposes of compositional reasoning. We formulate a compositional reasoning principle for safety properties of process networks and define symmetry groupoids and the quotient construction. Moreover, we show how symmetry and local reasoning can be expoited to provide parameterized proofs of correctness.

"*Whenever you have to do with a structure-endowed entity try to determine its group of automorphisms*"
Hermann Weyl, ***Symmetry***, 1952

"*... there are plenty of objects which exhibit what we clearly recognize as symmetry, but which admit few or no nontrivial automorphisms. It turns out that the symmetry, and hence much of the structure, of such objects can be characterized algebraically if we use groupoids and not just groups.*"
Alan Weinstein, ***Groupoids: Unifying Internal and External Symmetry – A Tour through Some Examples***, Notices of the AMS, 1996.

1 Introduction

State-space explosion is the main obstacle to the scalability of model checking. In this work, we consider proofs of safety for programs structured as a network of processes, executing concurrently and asynchronously. The network is used to represent how state is shared between groups of processes. The model is

⋆ Supported by a Natural Sciences and Engineering Research Council of Canada grant.

expressive, allowing refined statements of sharing relationships, such as read-only, read-write and write-only. As an example, globally shared memory may be represented by a hub-and-spoke network, with the memory at the hub and processes at the spokes; a dining philosophers network has processes arranged in a ring, with adjacent philosophers given read-write access to their shared fork.

A natural question is whether network symmetries can be exploited to reduce the complexity of model checking. Indeed, it is known that for networks which are highly symmetric, reducing the global state space by collapsing together symmetric states results in exponential savings [18, 6, 14]. On the other hand, many networks, such as rings, have limited global symmetry so this reduction is much less effective for those networks. We consider instead the *local symmetries* of a network and show that they can be used to significantly reduce the complexity of *compositional reasoning* methods.

The essence of compositional methods lies in using *local* reasoning as a substitute for *global* reasoning: each process of a concurrent program is analyzed separately along with an abstraction of its neighboring processes. The benefit is that local methods work in time polynomial in the number of processes, in contrast with the PSPACE-hardness of the model checking question. Efficiency comes, however, at the cost of incompleteness. (It is possible to overcome incompleteness by adding auxiliary state, at the cost of making the analysis less compositional.)

The intuition behind our results is that compositional methods, being local in their scope, benefit from purely local symmetries. Networks with little global symmetry can have significant local symmetry: in a ring network, for instance, any two nodes are locally similar, as they have identical left and right neighbors.

To illustrate these issues, consider a uniform ring network with N nodes. A program on this network may have a state space whose size is exponential in N – this is the case, for instance, of a simple token-based mutual exclusion protocol. The global symmetry group of the ring has N elements (the rotations), so the global state space can be reduced only by a factor of N. (The state space of a program could exhibit more symmetry than that of its underlying network, but that is not the case here.) We show that it is possible to automatically construct a compositional invariant which is strong enough to prove mutual exclusion, in time polynomial in N. Making use of the local symmetries of a ring, this calculation can be reduced to one on a fixed set of representative nodes, making the time complexity for computing the compositional invariant *independent* of N! Moreover, it is sometimes possible to pick the same set of representatives for all networks in a family. In such a case, the compositional invariant computed for a small instance forms a parameterized invariant which holds for all members of the family.

Technically, local symmetries are described by a *symmetry groupoid*, a generalization of a symmetry group (cf. the quotations at the start of this section). The main question tackled in this work is to determine precisely how the local symmetries of the network influence the symmetry of a compositional inductive invariant which is computed to prove safety properties. In the following sketch

of the main results, a local invariant has the shape $(\forall i : \theta_i)$, where the quantification in i is over the nodes of the network and θ_i is an assertion which depends only on the neighborhood of node i.

1. Given an "balance" relation B on the network (a form of bisimulation defining local symmetries), if $(i, j) \in B$ then θ_i and θ_j are similar. I.e., the local symmetry of the network is reflected in the symmetry of the computed compositional invariant.
2. The orbit relation of the group of global automorphisms of a network forms a balance relation. I.e., global symmetries induce local symmetries.
3. A groupoid balance relation induces a quotient network which is equivalent to the original for the purpose of local reasoning.
4. If there is a single quotient for a family of networks, the compositional invariant computed for this quotient generalizes to a parameterized invariant which holds for all networks in the family.

The results point to deep connections between local symmetry, compositional methods and parameterized reasoning.

2 Networks and Their Symmetry Groupoids

A *network* is given by a pair (N, E) where N is a set of *nodes* and E is a set of *edges*. Each node is assigned a *color* by a function $\xi : N \to C$, with C a set of colors. Associated with each edge is a color, given by a function $\xi : E \to C$ (we use the same color set for simplicity); a set of input nodes, given by $ins : E \to \mathcal{P}(N)$; and a set of output nodes, given by $outs : E \to \mathcal{P}(N)$, where $\mathcal{P}(N)$ represents the power-set of N. The input and output sets of an edge may overlap.

There are several derived notions. The incoming edges for a node are given by a function $In : N \to \mathcal{P}(E)$, defined by $In(n) = \{e \mid n \in outs(e)\}$. The outgoing set of edges for a node is similarly defined by a function $Out : N \to \mathcal{P}(E)$, given by $Out(n) = \{e \mid n \in ins(e)\}$. The set of edges incident to a node is defined by the function $InOut : N \to \mathcal{P}(E)$, given by $InOut(n) = In(n) \cup Out(n)$.

Definition 1 (Points-To) *A node m points-to node n, denoted $m \in pt(n)$, if $m \neq n$ and $Out(m) \cap InOut(n)$ is non-empty.*

Informally, two nodes are locally similar if their immediate neighborhoods are identical up to a re-mapping.

Definition 2 (IO-Similarity) *Nodes m and n are (locally) similar, written $m \simeq_{IO} n$, if (1) the nodes have the same color, i.e., $\xi(m) = \xi(n)$, and (2) there is a correspondence between respective sets of incident edges, which preserves color and in/out status. I.e., there is a function $\beta : InOut(m) \to InOut(n)$ which is a bijection between $In(m)$ and $In(n)$, a bijection between $Out(m)$ and $Out(n)$ and for every e, $\xi(e) = \xi(\beta(e))$.*

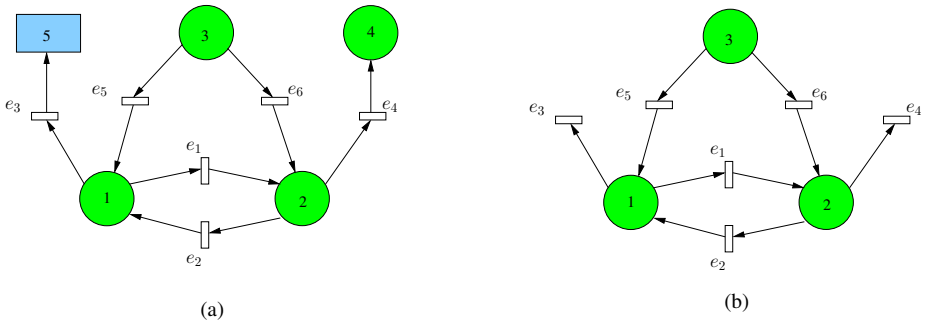

Fig. 1. Trivial Global Symmetry, Non-trivial Local Symmetry.

Figure 1(a) an example network, based on one from [16]. Colors are also marked by shapes: node 5 has a different color from node 4. This network has only the identity as a global automorphism; however, nodes 1 and 2 are locally similar, as their neighborhoods – shown in Figure 1(b) – are related by the bijection β which maps input edges $e_2 \mapsto e_1; e_5 \mapsto e_6$ and output edges $e_1 \mapsto e_2; e_3 \mapsto e_4$. Nodes 3 and 2 point to node 1, while nodes 3 and 1 point to node 2.

2.1 Local Symmetry Groupoids

The set of tuples of the form (m, β, n) where β is a witnessing bijection for $m \simeq_{IO} n$ forms a *groupoid*. Following [16], we call this the symmetry groupoid of the network and denote it by $\mathcal{G}_{IO}$. A groupoid (cf. [3, 26]) is (roughly) a group with a *partial* composition operation. It is defined by specifying a set of elements, E, a set of objects O, source and target functions $src : E \to O$ and $tgt : E \to O$ and an identity function $id : O \to E$. These must satisfy group-like conditions.

1. The composition ab of elements a, b is defined only if $tgt(a) = src(b)$, with $src(ab) = src(a)$ and $tgt(ab) = tgt(b)$
2. Composition is associative. If one of $a(bc)$ or $(ab)c$ is defined, so is the other, and they are equal
3. For every element a, the element $\lambda_a = id(src(a))$ is a left identity, i.e., $\lambda_a a = a$, and $\rho_a = id(tgt(a))$ is a right identity, i.e., $a\rho_a = a$
4. Every element a has an inverse (a^{-1}), such that $aa^{-1} = \lambda_a$ and $a^{-1}a = \rho_a$

A groupoid can be pictured as a directed graph: the nodes are the objects, there is a directed edge labeled by element e from its source to its target object. Identities form self-loops. (A group is a groupoid with a single base object.)

In the network symmetry groupoid, the objects are the nodes of the network and the elements are all triples (m, β, n) where β is a bijection defining the similarity between nodes m and n. The identities are defined by $id(n) = (n, \iota, n)$, where ι

is the identity map. For an element (m, β, n), $src(m, \beta, n) = m$, $tgt(m, \beta, n) = n$ and its inverse is (n, β^{-1}, m). The composition of (m, β, n) and (n, γ, o) is $(m, \gamma\beta, o)$.

A groupoid induces an *orbit relation*: objects a, b are related if there is a groupoid element e connecting them, i.e., if $src(e) = a$ and $tgt(e) = b$. From the groupoid properties, this is an equivalence relation. The orbit relation for the symmetry groupoid is just $\simeq_{IO}$.

3 Local Reasoning on a Process Network

In this section we develop an assume-guarantee rule for proving safety properties of process networks. It is similar to the rules from [15, 23] which apply to the globally shared memory model. Each node in a network is assigned a process, with locally similar nodes being assigned similar processes. The proof rule results in an inductive invariant of the form $(\forall i : \theta_i)$ where θ_i is an assertion on the neighborhood of the process assigned to node i. We show that there is a strongest invariant of this form and that it can be computed as a simultaneous fixpoint.

3.1 Assignment of Variables and Processes

Given a network (N, E), we associate a variable l_n with each node n and a variable v_e with every edge e. The type of the variable is the color of the node or edge. Let $X_n = \{v_e \mid e \in In(n)\}$ be the set of input variables for node n; similarly, let $Y_n = \{v_e \mid e \in Out(n)\}$ be the set of output variables for n and let $L_n = \{l_n\}$. Let $V_n = X_n \cup Y_n \cup L_n$. Thus, V_n defines the variables in the immediate neighborhood of node n. The set V is defined as $(\cup\, n : V_n)$.

With each node n is associated a *transition condition* $T_n(V_n, V_n')$ which is constrained so that it leaves the variables in $X_n \backslash Y_n$ unchanged. The network as a whole has an *initial condition*, $\mathcal{I}$. The variables assigned to nodes and edges, the network initial condition and the transition conditions for each node define an *assignment* of processes to the network. We denote the process for node n as P_n. Let $\tilde{I}_n = (\exists V \backslash V_n : \mathcal{I})$ be the projection of the initial condition on to V_n.

Definition 3 (Valid Assignment) *An assignment is* valid *for $B \subseteq \mathcal{G}_{IO}$ if it respects the local symmetries in B: i.e., for every $(m, \beta, n) \in B$, it should hold that $[T_n \equiv \beta(T_m)]$ and $[\tilde{I}_n \equiv \beta(\tilde{I}_m)]$.*

In this definition, $\beta(f)$ is a predicate which holds for a valuation b over V if f holds for a valuation a over V where for every $e \in InOut(m)$, $a(v_e) = b(v_{\beta(e)})$ and $a(v_e') = b(v_{\beta(e)}')$, and $a(l_m) = b(l_n)$ and $a(l_m') = b(l_n')$. Informally, $\beta(f)$ is the predicate obtained by substituting in f variables from the neighborhood of m (i.e., those in V_m) with the variables which correspond to them by β.

The semantics of a valid program assignment is defined as the *asynchronous, interleaving* composition of the processes associated with each node. The initial condition is $\mathcal{I}$. The interleaved transition relation, $\mathcal{T}$, is defined as a choice between local transitions, $(\exists n : \tilde{T}_n)$, where $\tilde{T}_n$ is the transition relation which extends T_n so that all variables of nodes other than n are unchanged. Formally, $\tilde{T}_n \equiv T_n \wedge unch(V \backslash V_n)$.

3.2 Local Proof Rules

A rely-guarantee proof rule based on the Owicki-Gries method is given in [23] for shared-memory programs. We generalize this formulation to networks. The rely-guarantee conditions are expressed over vectors of the form $\theta = (\theta_1, \theta_2, \ldots, \theta_n)$, where each component, θ_i, is a state assertion local to process P_i. The proof conditions ensure that the conjunction $(\forall i : \theta_i)$ is a *global inductive invariant*.

Let $\theta_n(V_n)$ be a predicate on the neighborhood variables of node n. For $(\forall n : \theta_n)$ to be a globally inductive invariant, θ_n must include the initial states of P_n, it must be closed under transitions of P_n and it must be closed under interference from the nodes which point to n. We next express these conditions precisely. We use two convenient notational conventions, taken from the book by Dijkstra and Scholten [12]. The notation $[\varphi]$ expresses that φ is valid. The notation $(\exists X : r : \varphi)$, where $X = \{x_1, \ldots, x_k\}$ is a finite set of variables, is a shorthand for $(\exists x_1, \ldots, x_k : r \wedge \varphi)$. The predicate r constrains the type or the range of variables in X. If X is empty, the quantified expression is equivalent to *false*.

The first condition is expressed as

$$[\tilde{I}_n \Rightarrow \theta_n] \tag{1}$$

The second condition is expressed as follows, where SP_n is the strongest post-condition operator for node n. ($SP(T, \psi)$ is the set of successors of states satisfying ψ by transitions satisfying T. $SP_n(T, \psi)$ is the projection of states in $SP(T, \psi)$ on to V_n.)

$$[SP_n(T_n, \theta_n) \Rightarrow \theta_n] \tag{2}$$

Closure under interference is expressed as follows

$$[SP_n(intf^{\theta}_{mn}, \theta_n) \Rightarrow \theta_n] \text{ for every } m \in pt(n) \tag{3}$$

The transition term, $intf^{\theta}_{mn}$ (read *intf* as "interference") represents the effect of transitions at a node m on the values of variables in the neighborhood of node n. This is defined as

$$intf^{\theta}_{mn} \equiv (\exists V \backslash V_n, V' \backslash V'_n : \tilde{T}_m \wedge \theta_m) \tag{4}$$

The interference term is a function of (V_n, V'_n), and is thus a general transition term. The definition of $\tilde{T}_m$ implies, however, that the interference leaves all variables not in $V_n \cap Y_m$ unchanged.

The three implications can be gathered together to form a simultaneous system of implications $[F_n(\theta) \Rightarrow \theta_n]$, with F_n defined by

$$F_n(\theta) \equiv \tilde{I}_n \vee SP_n(T_n, \theta_n) \vee (\vee m : m \in pt(n) : SP_n(intf^{\theta}_{mn}, \theta_n)) \quad (5)$$

This is in pre-fixpoint form as $F_n(\theta)$ is monotone in the vector θ, ordered component-wise by implication. By the Knaster-Tarski theorem, this system has a least fixpoint. For finite-state systems, the fixpoint can be computed as the limit, say θ^*, of the iteration sequence given by $\theta^0_m = \tilde{I}_n; \theta^{k+1}_m = \theta^k_m \vee SP_n(T_n, \theta^k_n) \vee (\vee m : m \in pt(n) : SP_n(intf^{\theta^k}_{mn}, \theta^k_n))$. For infinite-state systems, the limit may be trans-finite. Component θ^*_n is defined over V_n, as can be seen by its equivalence to $F_n(\theta^*)$ and the definition of F_n.

Theorem 1 *(Soundness) The proof rules (1)-(3) imply that $\theta = (\forall n : \theta_n)$ is a globally inductive invariant.*

Proof: The base case, that $[\mathcal{I} \Rightarrow \theta_n]$, follows for all n by (1), as $\mathcal{I}$ is stronger than $\tilde{I}_n = (\exists V \backslash V_n : \mathcal{I})$. To show inductiveness, consider any state s satisfying θ and a transition by process P_m from state s to state t. As θ_m holds of s, the transition satisfies both T_m and $intf^{\theta}_{mn}$. By (2), θ_m holds of t. Now consider any other node n. If m points to n, as θ_n holds of s by assumption, it follows by (3) that θ_n holds of t. If m does not point to n, the transition does not change the values of any variables in the neighborhood of n, so that θ_n continues to hold. **EndProof.**

Complexity Let L be the number of local states per process – i.e., the number of valuations to V_n, assuming all V_n's are identical. Let $|N|$ be the number of nodes in the network, which is also the number of components of the θ vector. Then, (1) the number of fixpoint rounds is at most $|N| * L$, as each round must strictly increase the set of states in at least one component; (2) the number of updates per round is $|N|$, as each component of θ is updated. The work for an update to θ_n is typically dominated by the interference term. Consider round k. For each m which points to n, this requires computing successors for all states in θ^k_n with respect to the transition relation $intf^{\theta^k}_{mn}$. For a state in θ^k, its successors can be found by looking up its association list in a table storing $intf^{\theta^k}_{mn}$. The cost of the successor computation is, therefore, bounded by $L * L$. The total cost is bounded by $(|N| * L) * |N| * (L^2 * D)$ where D is the maximum over all n of the size of $pt(n)$. This simplifies to $|N|^2 * L^3 * D$, which is polynomial in all parameters, whereas global model-checking is PSPACE-complete in $|N|$ which, in practice, implies time-complexity exponential in $|N|$.

Completeness Owicki-Gries [25] and Lamport [20] recognized that local assertions may not always suffice to represent the global constraints needed for a valid proof. The resolution is to expose local state through auxiliary or history

variables, a process which can be automated [8, 9, 17]. It was observed in [8] that for many protocols, constructing $(\forall ij)$ local invariants – described below – is a good alternative to adding auxiliary variables. We consider purely compositional proofs: auxiliary variables modify network symmetries in ways that will be explored in future work.

Pair-Indexed Properties A similar simultaneous fixpoint scheme can be constructed for multi-indexed properties, such as $(\forall m, n : m \neq n : \theta_{mn})$. The proof rules for a pair (m, n) are as follows. The term $\tilde{I}_{mn}$ is defined as $(\exists V \backslash (V_n \cup V_m) : \mathcal{I})$ and SP_{mn} is the projection of SP on to variables $V_m \cup V_n$. We abbreviate $(\forall m, n : m \neq n : \theta_{mn})$ by θ.

$$[\tilde{I}_{mn} \;\Rightarrow\; \theta_{mn}] \tag{6}$$

$$[SP_{mn}(T_m \;\wedge\; unch(V_n \backslash V_m), \theta) \;\Rightarrow\; \theta_{mn}] \tag{7}$$

$$[SP_{mn}(T_n \;\wedge\; unch(V_m \backslash V_n), \theta) \;\Rightarrow\; \theta_{mn}] \tag{8}$$

$$[SP_{mn}(intf^{\theta}_{kmn}, \theta) \;\Rightarrow\; \theta_{mn}], \text{ for } k \in pt(m, n) \text{ where} \tag{9}$$

$$intf^{\theta}_{kmn} \;\equiv\; unch((V_n \cup V_m) \backslash V_k) \;\wedge\; (\exists V'_k \backslash (V'_m \cup V'_n) : T_k) \text{ and} \tag{10}$$

$$k \in pt(m, n) \text{ if } k \notin \{m, n\} \text{ and } Out(k) \cap (InOut(m) \cup InOut(n)) \text{ is non-empty} \tag{11}$$

For the rest of the paper we focus first on the simpler case of singly-indexed properties, returning to pair-indexed properties at the end.

4 Symmetry and Quotients

The equivalence $\simeq_{IO}$ induced by the local symmetry groupoid $\mathcal{G}_{IO}$ is not enough in itself to obtain the symmetry reduction results. While it ensures that nodes m, n related by $\simeq_{IO}$ have a similar neighborhood, it does not ensure that the nodes which point into m and n correspond in any way. Some correspondence is needed, as the processes on $pt(m)$ affect θ_m and those on $pt(n)$ affect θ_n. We define a bisimulation-like relationship which builds on (strengthens) the basic local symmetry relation. We call relations satisfying the stronger conditions "balance" relations, following [16], where a similar notion is defined.

Definition 4 (Balance) *A* balance *relation B is a subset of $\mathcal{G}_{IO}$ satisfying the following properties. For any (m, β, n) in B: (1) (n, β^{-1}, m) is in B, and (2) for any j in $pt(m)$, there must be k in $pt(n)$ and δ such that (2a) (j, δ, k) is in B and (2b) for every edge f in $InOut(j) \cap InOut(m)$, $\delta(f) = \beta(f)$.*

Condition (2a) ensures that any node which points to m has an equivalent node which points into n. Condition (2b) ensures that β and δ agree on edges that are common to m, j. The theorem below summarizes properties of balance relations.

Theorem 2 *(Balance Properties) For any network:*

1. *The union of two balance relations is a balance relation*
2. *The composition of two balance relations is a balance relation*
3. *There is a largest balance relation, which we denote by* B^*
4. B^* *is a greatest fixpoint*
5. B^* *is a sub-groupoid of* $\simeq_{IO}$

The final balance property implies that the orbit relation for B^* is an equivalence. The fixpoint property induces a partition-refinement algorithm for computing the orbit relation of B^* which is polynomial in the size of the network.

4.1 Automorphisms and Balance

Informally, an automorphism of a network is a permutation of the edge and node set which leaves the network structure unchanged. Formally, for a network (N, E), an *automorphism* is given by a function π which is a bijection from N to N and a bijection from E to E such that

1. (Color Preservation) For any node n, $\xi(n) = \xi(\pi(n))$, and for any edge e, $\xi(e) = \xi(\pi(e))$
2. (Link Preservation) For any node n and edge e, $n \in \mathit{ins}(e)$ holds iff $\pi(n) \in \mathit{ins}(\pi(e))$ and $n \in \mathit{outs}(e)$ holds iff $\pi(n) \in \mathit{outs}(\pi(e))$

The global symmetry of the network is defined by its set of automorphisms, which forms a group under function composition. Given an automorphism group, G, of the network, define $\mathit{Local}(G)$ as the set of triples (m, β, n) where, for some $\pi \in G$, $\pi(m) = n$ and β is π restricted to $\mathit{InOut}(m)$. The following theorem shows that global automorphisms induce balance relations.

Theorem 3 *For any automorphism group* G *of a network,* $\mathit{Local}(G)$ *is both a sub-groupoid of* $\mathcal{G}_{IO}$ *and a balance relation.*

The network of Figure 1 has a balance relation connecting 1 and 2 through the bijection β, even though the only automorphism is the identity.

4.2 Balance and Symmetry

The following theorem shows how a balance relation influences the symmetry of the computed invariant. We say that a vector θ *respects* a balance relation B if for all (m, β, n) in B, $[\theta_n \equiv \beta(\theta_m)]$.

Lemma 1 *Let* B *be a balance relation. Consider a program assignment which is valid for* B*. For any* $(m, \beta, n) \in B$ *and any transition condition* $t(V_m, V'_m)$ *and any predicate* $p(V)$*, it is the case that* $[\beta(SP_m(t, p)) \equiv SP_n(\beta(t), \beta(p))]$.

Lemma 2 *Let B be a balance relation. Consider a program assignment which is valid for B. For all $(m, \beta, n) \in B$, any θ which respects B, and $j \in pt(m), k \in pt(n)$ which correspond for (m, β, n) by B, $[\beta(intf^{\theta}_{jm}) \equiv intf^{\theta}_{kn}]$ holds.*

Theorem 4 *(Symmetry Reduction) Let B be a balance relation. For a program assignment which is valid for B, the computed local invariant θ^* respects B.*

Proof: The proof is by transfinite induction on the fixpoint stages. The inductive assumption at stage λ is that θ^S respects B for all stages S which precede λ.

(Basis) The initial values $\theta^0_m = \tilde{I}_m$ and $\theta^0_n = \tilde{I}_n$ are related as claimed by the validity of the assignment.

(Step ordinal) Suppose that the hypothesis is true at stage S. The definition of θ^{S+1}_m is $\theta^S_m \ \vee \ SP_m(T_m, \theta^S_m) \ \vee \ (\vee j : j \in pt(m) : SP_m(intf^{\theta^S}_{jm}, \theta^S_m))$. Applying β, which distributes over $\vee$, we get

$$\beta(\theta^S_m) \ \vee \ \beta(SP_m(T_m, \theta^S_m)) \ \vee \ (\vee j : j \in pt(m) : \beta(SP_m(intf^{\theta^S}_{jm}, \theta^S_m))) \quad (12)$$

The SP terms satisfy the conditions of Lemma 1. By the inductive hypothesis and Lemma 1, we get

$$\theta^S_n \ \vee \ SP_n(\beta(T_m), \theta^S_n) \ \vee \ (\vee j : j \in pt(m) : SP_n(\beta(intf^{\theta^S}_{jm}), \theta^S_n)) \quad (13)$$

By valid program assignment and Lemma 2, this is equivalent to

$$\theta^S_n \ \vee \ SP_n(T_n, \theta^S_n) \ \vee \ (\vee k : k \in pt(n) : SP_n(intf^{\theta^S}_{kn}, \theta^S_n)) \quad (14)$$

There is a slight subtlety in the last step. By the definition of B, every j has a corresponding $k \in pt(n)$. As B is closed under inverse, all k in $pt(n)$ are related to some $j \in pt(m)$. Hence, the interference terms for m map exactly to the interference terms of n. The final expression is just the definition of θ^{S+1}_n.

(Limit ordinal) Suppose that the hypothesis is true for all stages S below a limit ordinal λ. As β distributes over arbitrary unions, we obtain the chain of equivalences $\beta(\theta^\lambda_m) \ \equiv \ \beta(\vee S : S \prec \lambda : \theta^S_m) \ \equiv \ (\vee S : S \prec \lambda : \beta(\theta^S_m)) \ \equiv \ (\vee S : S \prec \lambda : \theta^S_n) \ \equiv \ \theta^\lambda_n$. **EndProof.**

4.3 Symmetry-Reduced Local Invariant Computation

The main symmetry theorem gives rise to the following symmetry-reduced fixpoint computation for the local invariant.

1. Fix a balance relation B which is a sub-groupoid of $\mathcal{G}_{IO}$. (B^* is one such relation.) Let $\simeq_B$ be its orbit relation; this is an equivalence.
2. Pick a representative from each equivalence class of $\simeq_B$. For a node n, let $rep(n)$ denote its representative.

3. For each non-representative node n fix a bijection β_n such that $(rep(n), \beta_n, n)$ is a triple in B. For a representative node r, fix β_r to be the identity.
4. Compute the fixpoint over the set of representatives. The fixpoint vector has a component θ_r for each representative r. To compute the update for representative r, use the formula for $F_r(\theta)$, except that the term θ_n for a node n which is not a representative node is replaced with $\beta_n(\theta_{rep(n)})$.

By induction on the fixpoint stages, we get the theorem below. The complexity of the symmetry-reduced calculation is given by the formula derived previously, with $|N|$ replaced by the number of representatives.

Theorem 5 *The symmetry-reduced computation computes the same least fixpoint as the original.*

4.4 Equivalent Networks and the Quotient Construction

A balance relation which is a groupoid (for instance, B^*) induces an orbit relation, which is an equivalence on the nodes. This partitions nodes into equivalence classes. We use the classes to define a quotient network, and show that it suffices to compute the local invariant on the quotient.

A quotient is an instance of the more general concept of an *equivalent network*. For networks W_1 and W_2 with valid assignments, W_2 is equivalent to W_1 via the relation $R \subset N_1 \times N_2$ if, for all $(i, j) \in R$, $[\theta_1^*(i) \equiv \theta_2^*(j)]$. Every network is equivalent to itself through the identity relation. A quotient construction produces a smaller assigned network which is equivalent to the original.

Given a network $W = (N, E)$ and a groupoid balance relation B, a *quotient* $\overline{W}$ is defined as follows.

1. The nodes of $\overline{W}$ are the equivalence classes of $\simeq_B$. Each class C has a defined representative, denoted $rep(C)$, chosen arbitrarily. We write the class for node n as $\overline{n}$. The color of a class is the (common) color of all nodes in it.
2. For a class C with representative r, there is an edge $\overline{e}$ for each edge e in $InOut(r)$. The edge $\overline{e}$ connects equivalence classes of nodes connected by e. In more detail, $m \in ins(e)$ iff $\overline{m} \in ins(\overline{e})$, and $m \in outs(e)$ iff $\overline{m} \in outs(\overline{e})$. The color of a quotient edge is the color of the edge which generates it.
3. A class C with representative r is assigned similarly to r; i.e., such that $[\tilde{I}_C \equiv \beta_r(\tilde{I}_r)]$ and $[T_C \equiv \beta_r(T_r)]$, where β is the bijection which relates each e in $InOut(r)$ to its corresponding edge $\overline{e}$.

The quotient is not unique, except under stronger conditions on the balance groupoid. (Non-uniqueness arises as the balance definition allows representatives x and y for a class C to have corresponding edges e and f such that e and f have inequivalent *outs* sets. This does not, however, influence the invariant computation, as only *ins* sets are relevant for the points-to definition.) The

theorem below shows that local invariant computed on a quotient is identical to that on the original network for the representative nodes. Values for non-representative nodes are obtained by the transformation given in Theorem 4.

Theorem 6 *Any quotient $\overline{W}$ is equivalent to W via $R = \{(r, C) \mid r = rep(C)\}$.*

5 Pairwise Symmetry and Balance Relations

In this section, we turn to symmetry reduction for invariants of the form $(\forall i, j : i \neq j : \theta_{ij})$. The definitions of pairwise local symmetry and balance, given below, are analogues of the previous definitions for singly-indexed invariants.

For a pair of nodes (i, j), let $In(i, j) = In(i) \cup In(j)$ and let $Out(i, j) = Out(i) \cup Out(j)$. A node k is in $pt(i, j)$ if $k \notin \{m, n\}$ and $Out(k) \cap InOut(i, j)$ is non-empty. A *pairwise local symmetry* between (i, j) and (m, n) is possible if $\xi(i) = \xi(m)$ and $\xi(j) = \xi(n)$, and there is a a function β such that (i, β, m) and (j, β, n) are local symmetries. The set of all pairwise local symmetries $((i, j), \beta, (m, n))$ forms the pairwise symmetry groupoid of the network.

A *pairwise balance relation* B is a subset of the pairwise symmetry groupoid of the network which is closed under inverse, and such that for all $((i, j), \beta, (m, n))$ in B, and every k in $pt(i, j)$, there is l in $pt(m, n)$ such that

1. There is a function δ so that (k, δ, l) is a local symmetry, and
2. For all edges e in $InOut(k) \cap InOut(i, j)$, it is the case that $\beta(e) = \delta(e)$

A program assignment is valid for B if for any $((i, j), \beta, (m, n))$ in B, $[\beta(T_i) \equiv T_m]$, $[\beta(T_j) \equiv T_n]$ and $[\beta(\tilde{I}_{ij}) \equiv \tilde{I}_{mn}]$. With a proof strategy similar to that for singly-indexed properties, we have the following analogue of Theorem 4.

Theorem 7 *(Pairwise Symmetry Reduction) Let B be a pairwise balance relation. For any program assignment valid for B: for any $((i, j), \beta, (m, n))$ in B, in the computed local pairwise invariant, θ^*, it is the case that $[\theta^*_{mn} \equiv \beta(\theta^*_{ij})]$.*

From a global automorphism group G, define $Local_2(G)$ as the set of triples $((i, j), \beta, (m, n))$ such that for a permutation π in G, $\pi(i) = m, \pi(j) = n$ and β is π restricted to $InOut(i, j)$.

Theorem 8 *For any automorphism group G of a network, $Local_2(G)$ is a pairwise groupoid balance relation.*

6 Consequences

Consider a simple token-passing protocol on a unidirectional ring network. Each process is in one of three states: thinking (T), hungry (H), and eating (E). It moves from T to H on its own; from H to E by removing a token from its left edge; and from E to T on its own, placing the token on its right edge. The predicate t_i expresses the presence of a token on the edge to the left of node i.

The singly-indexed local invariant for a ring is too weak to conclude safety (mutual exclusion). However, the pairwise local invariant suffices. It is given by $(\forall i, j : i \neq j : (t_i \Rightarrow \neg t_j) \wedge (E_i \Rightarrow \neg E_j \wedge \neg t_i \wedge \neg t_j))$.

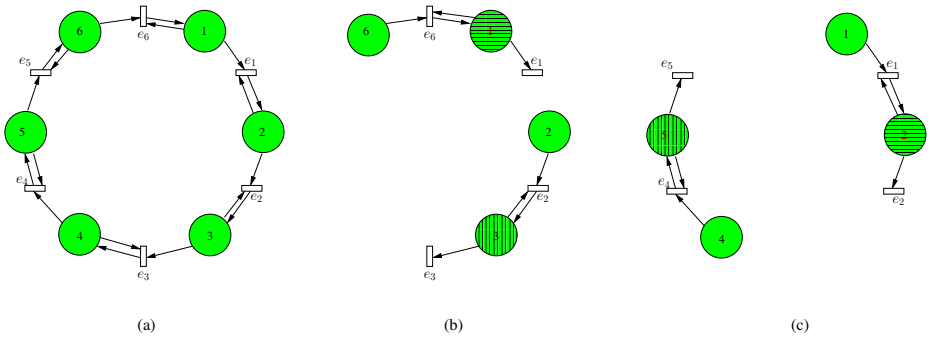

Fig. 2. Token-Ring Network and Neighborhood Views for $(1, 3)$ and $(2, 5)$

By Theorem 8, $Local_2(G)$ is a pairwise balance relation. Pairs (i, j) and (m, n) are related if the nodes in the pairs are the same distance apart (clockwise) on the ring. Thus, $(1, 2)$ is a representative for spacing 1, and $(1, 3)$ is a representative for spacing 2. It turns out that $(1, 3)$ is also a representative for any larger spacing, as the relation between $(1, 3)$ and (m, n) with spacing at least 2 is a balance relation. Figure 2 shows, for example, the similar local neighborhoods of $(1, 3)$ and $(2, 5)$ in a ring of size 6.

It suffices, therefore, to compute the pairwise invariant over the representative pairs $(1, 2)$ and $(1, 3)$ for a fixed ring network of size at least 3. Moreover, for a family of ring networks, each instance has the same pair of representatives. The following theorem establishes conditions under which a pairwise invariant generalizes to an invariant for any larger instance.

Theorem 9 *For a uniform ring network family where the processes and node and edge data types are independent of the size of the ring, the pairwise invariant computed for a ring of size 3 holds (by extending the range of node indices) for all larger ring sizes.*

For a dining philosophers protocol, mutual exclusion is required only between neighboring processes. For an abstract dining philosophers protocol, the singly-indexed invariant $(\forall i : E_i \;\Rightarrow\; fork_{i-1} = R \;\wedge\; fork_i = L)$ holds, where L, R represent left and right directions. Thus, the symmetry-reduced structure is a single node, which also proves that the invariant holds in a parameterized sense.

7 Related Work, Conclusions and Open Questions

There is a large body of work on compositional methods for verification of concurrent programs. Much of this work, the early examples of which are the Owicki-Gries method [25] and proof rules used by Lamport [20] and Jones [19], applies to a memory model where all processes share a common memory. The assume-guarantee method of Misra and Chandy [4] is based on a network model with processes communicating on unbounded queues. Compositional methods for CCS and CSP are described in [11, 21]. Proof rules based on CCS/CSP synchronization have been automated using learning techniques [7]. In [24] and [1], for example, compositional proof rules are given that are sound (and semantically complete) for the full range of Linear Temporal Logic properties, thus including safety properties, liveness properties and fairness properties. Local reasoning has also been applied to synchronous computation [5, 22].

Our network model is based on atomic actions and shared memory rather than CCS/CSP style synchronization or message queues. Using it, it is possible to represent, for instance, the sharing of forks among dining philosophers. The proof rules are assertion-based. The key idea of reduction with local symmetries should, we believe, carry over to other models of process communication.

Earlier work [23, 10] on symmetry reduction for compositional reasoning applies to programs with a common shared memory. This paper significantly generalizes the scope of symmetry reduction to arbitrary networks of processes and fine-grained sharing relationships.

These results have been strongly influenced by the work of Golubitsky and Stewart on local symmetry in networks [16], but there are crucial differences in both the problem domain and the questions being addressed. The networks in [16] are clocked synchronous networks where the "program" at each node is given by an ordinary differential equation. The authors show that the local symmetries of the network influence the emergence of computations in which a group of nodes have completely synchronized (or phase-shifted) values. (In temporal logic terms, the network satisfies properties such as $\mathsf{EG}(x_1 = x_2)$ or $\mathsf{EG}(x_1' = x_2)$.) They identify balance as a necessary (and, in a sense, sufficient) condition for this behavior. Our results, on the other hand, are about interleaved process execution and universal rather than existential safety properties. We do make use of and adapt the groupoid formulation defined in their paper to describe local symmetries.

The results on parameterized verification build on the idea of generalizing from proofs of small instances that was explored in the work on "invisible invariants" [2]. This method was connected to compositional reasoning in [23]. The

earlier papers used a globally shared memory model; the network model results in a strengthening of the results, especially for ring networks. The token-ring example from Section 6 falls into the decidable class from [13] but the result here is both more general in that it applies also to non-token-passing protocols and yet limited in that it applies only to inductive invariants. There is, of course, a variety of other methods for parameterized verification; these are, in general, incomparable. Our results do point to intriguing connections between local symmetry, compositional invariants and parameterized verification.

This work shows that striking reductions can be obtained by considering the combination of local symmetries with compositional reasoning. There are several intriguing open questions: the proper treatment of auxiliary variables, deriving similar results for CCS/CSP-style synchronization, extending the symmetry reduction theorems from safety to liveness properties and exploring the role of local symmetries in proofs of parameterized properties of irregular networks.

References

1. N. Amla, E. A. Emerson, K. S. Namjoshi, and R. J. Trefler. Visual specifications for modular reasoning about asynchronous systems. In *Proceedings of the 22nd IFIP WG 6.1 International Conference Houston on Formal Techniques for Networked and Distributed Systems*, FORTE '02, pages 226–242, London, UK, UK, 2002. Springer-Verlag.
2. T. Arons, A. Pnueli, S. Ruah, J. Xu, and L. D. Zuck. Parameterized verification with automatically computed inductive assertions. In *CAV*, volume 2102 of *LNCS*, pages 221–234, 2001.
3. R. Brown. From groups to groupoids: A brief survey. *Bull. London Math. Society*, 19:113–134, 1987.
4. K. Chandy and J. Misra. Proofs of networks of processes. *IEEE Transactions on Software Engineering*, 7(4), 1981.
5. H. Cho, G. D. Hachtel, E. Macii, B. Plessier, and F. Somenzi. Algorithms for approximate FSM traversal based on state space decomposition. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 15(12):1465–1478, 1996.
6. E. M. Clarke, T. Filkorn, and S. Jha. Exploiting symmetry in temporal logic model checking. In *CAV*, volume 697 of *LNCS*, 1993.
7. J. M. Cobleigh, D. Giannakopoulou, and C. S. Pasareanu. Learning assumptions for compositional verification. In *TACAS*, volume 2619 of *LNCS*, pages 331–346. Springer, 2003.
8. A. Cohen and K. S. Namjoshi. Local proofs for global safety properties. In *CAV*, volume 4590 of *LNCS*, pages 55–67. Springer, 2007.
9. A. Cohen and K. S. Namjoshi. Local proofs for linear-time properties of concurrent programs. In *CAV*, volume 5123 of *LNCS*, pages 149–161. Springer, 2008.
10. A. Cohen and K. S. Namjoshi. Local proofs for global safety properties. *Formal Methods in System Design*, 34(2):104–125, 2009.
11. W.-P. de Roever, F. de Boer, U. Hannemann, J. Hooman, Y. Lakhnech, M. Poel, and J. Zwiers. *Concurrency Verification: Introduction to Compositional and Noncompositional Proof Methods*. Cambridge University Press, 2001.

12. E. Dijkstra and C. Scholten. *Predicate Calculus and Program Semantics*. Springer Verlag, 1990.
13. E. Emerson and K. Namjoshi. Reasoning about rings. In *ACM Symposium on Principles of Programming Languages*, 1995.
14. E. Emerson and A. Sistla. Symmetry and model checking. In *CAV*, volume 697 of *LNCS*, 1993.
15. C. Flanagan and S. Qadeer. Thread-modular model checking. In *SPIN*, volume 2648 of *LNCS*, pages 213–224, 2003.
16. M. Golubitsky and I. Stewart. Nonlinear dynamics of networks: the groupoid formalism. *Bull. Amer. Math. Soc.*, 43:305–364, 2006.
17. A. Gupta, C. Popeea, and A. Rybalchenko. Predicate abstraction and refinement for verifying multi-threaded programs. In *POPL*. ACM, 2011.
18. C. Ip and D. Dill. Better verification through symmetry. *Formal Methods in System Design*, 9(1/2), 1996.
19. C. Jones. Tentative steps toward a development method for interfering programs. *ACM Trans. on Programming Languages and Systems (TOPLAS)*, 1983.
20. L. Lamport. Proving the correctness of multiprocess programs. *IEEE Trans. Software Eng.*, 3(2), 1977.
21. L. Lamport and F. B. Schneider. The "Hoare Logic" of CSP, and All That. *ACM Trans. Program. Lang. Syst.*, 6(2):281–296, 1984.
22. I.-H. Moon, J. H. Kukula, T. R. Shiple, and F. Somenzi. Least fixpoint approximations for reachability analysis. In *ICCAD*, pages 41–44, 1999.
23. K. S. Namjoshi. Symmetry and completeness in the analysis of parameterized systems. In *VMCAI*, volume 4349 of *LNCS*, 2007.
24. K. S. Namjoshi and R. J. Trefler. On the completeness of compositional reasoning methods. *ACM Trans. Comput. Logic*, 11:16:1–16:22, May 2010.
25. S. S. Owicki and D. Gries. Verifying properties of parallel programs: An axiomatic approach. *Commun. ACM*, 19(5):279–285, 1976.
26. A. Weinstein. Groupoids: Unifying internal and external symmetry-a tour through some examples. *Notices of the AMS*, 1996.