Uncovering Symmetries in Irregular Process Networks

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. In this work, we consider distributed protocols that operate on arbitrary networks. The analysis of such protocols is challenging, as an arbitrarily chosen network may have limited global symmetry. We describe a methodology that uncovers significant *local* symmetries by appropriately abstracting node neighborhoods in a network. The local symmetries give rise to uniform compositional proofs of correctness. As an illustration of these ideas, we show how to obtain a uniform compositional invariance proof for a Dining Philosophers protocol operating on a fixed-size, arbitrary network. An interesting and somewhat unexpected consequence is that this proof generalizes easily to a parametric proof, which holds on any network regardless of size or structure.

1 Introduction

A distributed protocol may be viewed as a collection of processes communicating over an underlying interconnection network. In many protocols, the processes are similar, while the network may be arbitrary. Examples are networking protocols such as TCP/IP and BGP, application-level protocols such as termination detection and global snapshot, and protocols for sensor and ad-hoc networks. The verification questions are (1) the analysis of arbitrary, fixed-size instances and (2) showing correctness in the parameterized sense; i.e., over an unbounded set of network instances.

These analysis questions are challenging, in large part because standard symmetry arguments do not apply to networks with irregular structure. On the other hand, proofs carried out by hand (e.g., those in [3]) make few distinctions between nodes; the typical inductive invariant has the uniform shape "for every node m, ...". This observation motivates our work. We conjecture that many distributed protocols can be analyzed uniformly, even if the underlying networks are irregular. Furthermore, we also conjecture that, once discovered, the uniformity can guide the construction of a parameterized proof. The parameterized model checking question is undecidable in general [2].

* Supported in part by grants from the Natural Sciences and Engineering Research Council of Canada.

To make progress on this conjecture, we look to a combination of abstraction and compositional reasoning. The components of our analysis are as follows.

1. Uncover local similarities in a network by abstracting node neighborhoods.
2. Perform a compositional analysis on the abstracted network to fully exploit the newly uncovered local symmetries; the result is an inductive invariant.
3. Check whether the reductions due to local symmetries are powerful enough for the invariant to be parametric.

Compositional analysis, by its nature, is less sensitive to global irregularities in network structure. This is because the analysis is carried out for each node individually, taking into account interference only from neighboring nodes. In recent work [19], we showed that the limited sensitivity makes it possible for compositional methods to take advantage of local symmetries in a network. As an example, consider a ring network of N nodes. The global symmetry group of the ring has size $O(N)$. Hence, standard symmetry reduction methods have limited effect: a state space of size potentially exponential in N can be reduced only by a linear factor. On the other hand, any two nodes are locally similar, as their immediate neighborhoods are identical. Using this local symmetry, a compositional invariant can be computed on a *single* representative node. This reduction also enables a parametric proof, as the representative may be chosen to be the same for all ring networks.

These earlier results, however, find their best application to networks with a regular structure, such as star, ring, mesh, and complete networks. In an irregular network, two obstacles arise. The first is that nodes may have different numbers of neighbors; this suffices to make them locally dissimilar. Even if all nodes have the same degree, irregular connectivity may limit the degree of *recursive* local similarity, called “balance”, which is needed for the most effective symmetry reduction. To obtain a uniform analysis for irregular networks, it is necessary, therefore, to redefine local symmetry in a more general form. We do so in this work, which makes the following contributions.

– We formulate a notion of *local symmetry up to abstraction*. This generalizes the structural definition of local symmetry from [19] to a semantic one.
– We show that nodes that are “balanced” (i.e., recursively locally similar) have similar components in the strongest compositional invariant.
– Hence, an compositional invariant can be calculated using only a single representative from each equivalence class of balanced nodes.
– We show completeness: for any compositional invariant, it is always possible to derive it through a network abstraction based on a small set of local predicates, one that creates a highly locally-symmetric abstract network.
– We illustrate these ideas by showing how local symmetries may be used to calculate a parametric invariant for a Dining Philosophers protocol.

2 Abstraction Uncovers Symmetry

It is well understood that standard symmetry reduction [4, 12] is a form of abstraction: symmetric states are bisimular, and the reduction abstracts a state to its bisimulation equivalence class. This work illustrates a converse principle: that abstraction may help uncover hidden symmetries. We demonstrate this with an example based on global symmetries. The subsequent sections work out this principle for local symmetries.

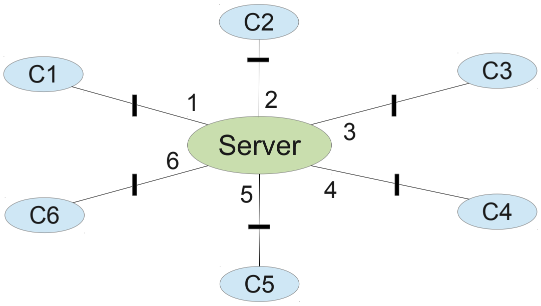

Fig. 1. A client-server network.

The example is a client-server protocol with N identical clients. A 6-client instance is shown in Figure 1. The server controls access to a shared resource. Each client may request access to the resource, in which case it waits for a grant and eventually returns the resource to the server. To ensure fairness, the server cycles through its adjacent edges in clockwise order: if the last grant was given to the client on edge number c and the resource is available, the next grant is given to the first requesting client encountered when examining edges in clockwise order – i.e., in the sequence $c + 1, c + 2, ..., c + N$, where addition is modulo N.

This system has an exponential reachable state space, of size at least 2^N, as the subset of clients that have an outstanding request could be arbitrary. Although the picture suggests that any pair of clients can be interchanged, the operation of the server restricts the group of isomorphisms of the system to that of a ring. Hence, the degree of reduction that is possible is only $O(N)$.

An important safety property is mutual exclusion: at any time, at most one client should have a grant. This can be established with a simpler server process: if the resource is available, the server chooses a requesting client non-deterministically and grants its request. Formally, the abstract server simulates the original; hence, the abstract system as a whole simulates the original system. Moreover, the abstract system satisfies mutual exclusion. The abstract system has an exponential number of states as well. However, its automorphism group is the full symmetry group on the clients, so its state space can be reduced by an exponential factor.

3 Background: System Model, Compositional Invariants

This section introduces the system model, compositional invariants, and the fixpoint computation which produces the strongest such invariant. The material is largely a summary of [18, 7, 19].

System Model A network is given as a graph. The graph has nodes and edges as objects, which are each given a color. Every node is connected to a set of edges; this set is ordered at the node according to an arbitrarily defined numbering. With respect to a node, a connected edge is either an input edge, an output edge, or both an input and an output edge. The network shown in Figure 1 has this form; edges are represented by rectangles and the numbering of edges on the server side is shown. Each edge is both an input and an output edge for its adjacent client and server node.

An *assignment* maps a set of processes to a network. The assignment also defines a state type for each node and each edge; types correspond to the coloring of the nodes and edges. The process assigned to a node m is denoted P_m. The *internal state* of this process is given by a value of the node state type. A *local state* of this process is given by a tuple of the form $(i, v_1, v_2, \ldots, v_k)$, where i is an internal state, and $v_1, v_2, \ldots, v_k$ is a valuation to the states of its adjacent edges, ordered by the numbering assigned to the edges at node m. For convenience, we associate symbolic variable names to all nodes and edges. The set of all node and edge variables is denoted V. The set of variables for process P_m is denoted V_m; these variables represent its internal state and the state of its adjacent edges. Hence, a local state is a valuation to V_m. The transition relation for P_m is denoted T_m. It relates local states of m with the constraint that if (s, t) is in T_m then the values for non-output edges of node m must be identical in s and t.

A *global* state is a valuation to all variables. Equivalently, a global state can be viewed as a set of local states where the local states for any two nodes agree on the value assigned to any common edge. The set of *global initial states* is denoted I. The projection of I on m, the set of *local initial states* for P_m, is denoted by I_m. Symbolically, I_m may be written as $(\exists V \backslash V_m : I)$. The quantification over all variables not in V_m projects I onto V_m. The global transition graph is induced by interleaving transitions from individual processes, starting from an initial state. There is a transition by process m from global state s to global state t if $T_m(s[V_m], t[V_m])$ holds ($s[V_m]$ is the local state of m in s) and for every variable x not in V_m, $s[x] = t[x]$.

Inductive and Compositional Invariants An *inductive invariant* for a transition system is a set of global states which (a) includes all initial states and (b) is closed under all transitions. Formally, for an invariant ξ, condition (a) is denoted as $[I \Rightarrow \xi]$ and condition (b) as $[SP(T_i, \xi) \Rightarrow \xi]$, for all i. SP is the strongest post-condition operator (also known as the successor function, or as *post*).

A *compositional invariant* (called a "split" invariant in [18]) is an inductive invariant of a special shape: it is formed from the conjunction of a number of local invariants, one for each process. Hence, it can be represented as a vector, $\theta = (\theta_1, \theta_2, \ldots, \theta_N)$, where each θ_i is defined over V_i and is itself an inductive invariant for process P_i. Equivalently, the constraints defining a compositional invariant are as follows.

- (Initiality) θ_i includes all initial states of P_i; formally, $[I_i \;\Rightarrow\; \theta_i]$
- (Step) θ_i is closed under transitions of P_i; formally, $[SP_i(T_i, \theta_i) \;\Rightarrow\; \theta_i]$, and
- (Non-interference) θ_i is closed under actions of neighboring processes. Formally, $[SP_i(\mathit{intf}^{\theta}_{ki}, \theta_i) \;\Rightarrow\; \theta_i]$, for any process k which points to i.

The predicate transformer SP_i is the strongest post-condition operator for node i. Node k *points to* node i if an output edge of k is adjacent to i. A transition of k may modify a local state of node i only if k points to i.

The term $\mathit{intf}^{\theta}_{ki}$ (read as "interference by k on i") is a transition condition. It describes how the local state of m may be changed due to moves by process k from states in its local invariant. Formally, $\mathit{intf}^{\theta}_{ki}$ is the projection of T_k, under θ_k, on to variables shared with m. This can be written symbolically as $\mathit{intf}^{\theta}_{ki} = (\exists V \backslash V_i, V' \backslash V'_i : T_k \wedge \theta_k)$, where the primed variables denote next-state values.

The Strongest Compositional Invariant as a Least Fixpoint Grouping together the initiality, step, and non-interference constraints gives a set of simultaneous implications of the shape $[F_i(\theta) \;\Rightarrow\; \theta_i]$. Here, $F_i(\theta)$ is the disjunction of the terms appearing on the left-hand side of the constraints for θ_i: namely, I_i, $SP_i(T_i, \theta_i)$, and $SP_i(\mathit{intf}^{\theta}_{ki}, \theta_i)$ for all k pointing to i. As F_i is monotonic in θ for all i (vectors are ordered by point-wise implication), the set of constraints has a least vector solution by the Knaster-Tarski theorem. The least solution, denoted by θ^*, forms the strongest solution to the constraints, and is therefore the strongest compositional invariant.

The least fixpoint is calculated in the standard manner by a process of successive approximation. The initial approximation, θ^0_i, is the empty set for all i. The approximation $\theta^{(K+1)}_i$ for stage $(K+1)$ is defined as $F_i(\theta^K)$. Standard methods, such as widening, may be used to ensure convergence for infinite-state systems. This is a synchronized computation. However, by the chaotic iteration theorem of [9], the simultaneous least fixpoint may be computed in an asynchronous manner, following any "fair" schedule (one in which each component is eventually given a turn). In Figure 2 we show one possible implementation of the computation.

```
var $\theta, \theta'$: vector
initially, for all $i$: $\theta_i = \emptyset, \theta'_i = I_i$
while $(\theta \neq \theta')$ do
   forall $i$: $\theta_i := \theta'_i$
   forall $i$: $\theta'_i := \theta_i \ \vee \ SP_i(T_i, \theta_i) \ \vee \ ( \vee\, k : k \text{ points-to } i : SP_i(\mathit{intf}^{\theta}_{ki}, \theta_i))$
done
```

Fig. 2. Computing the Compositional Fixpoint.

4 Informal Analysis of A Dining Philosophers Protocol

In this section, we describe a protocol for the Dining Philosophers problem and outline an analysis which performs local abstraction to extract symmetry. This is done in an informal manner; the justification for the soundness of these steps is laid out in the following sections.

The protocol We model a Dining Philosophers protocol (abbreviated by DP) as follows. The protocol consists of a number of similar processes operating on an arbitrary network. Every edge on the network models a shared "fork"; the variable for the edge between nodes i and j is called f_{ij}. Its domain is $\{i, j, \bot\}$. Node i is said to *own* the fork f_{ij} if $f_{ij} = i$; node j owns this fork if $f_{ij} = j$; and the fork is available if $f_{ij} = \bot$.

The process at node i goes through the following internal states: T (thinking); H (hungry); E (eating); and R (release). Each state s is really a "super-state" with a sub-state s_X for every subset X of adjacent forks, but we omit this detail for simplicity. Let *nbr* be the neighbor relation between processes. The state transitions for a process are as follows.

- A transition from T to H is always enabled.
- In state H, the process acquires forks, but may also choose to release them
    - (acquire fork) if $nbr(i, j)$ and $f_{ij} = \bot$, set $f_{ij} \ := \ i$,
    - (release fork) if $nbr(i, j)$ and $f_{ij} = i$, set $f_{ij} \ := \ \bot$, and
    - (to-eat) if $(\forall j : nbr(i, j) : f_{ij} = i)$ holds, change state to E.
- A transition from E to R is always enabled.
- In state R, the process releases its owned forks.
    - (release fork) if $nbr(i, j)$ and $f_{ij} = i$, set $f_{ij} \ := \ \bot$
    - (to-think) if $(\forall j : nbr(i, j) : f_{ij} \neq i)$, change state to T

The initial state of the system is one where all processes are in internal state $T_{\emptyset}$ and all forks are available (i.e., have value $\bot$).

Correctness Properties The desired safety property is that there is no reachable global state where two neighboring processes are in the eating state (E). The protocol given above is safe. It is also free of deadlock, as a process may always release a fork to its neighbor. It is, however, not free of livelock. Our focus is on a proof of the safety property of mutual exclusion between neighbors.

Abstract DP model The simplest abstraction is to have just the four abstract states: T, H, E, R, corresponding to the four super-states. The abstract transitions derived from standard existential abstraction are $T \to H, H \to H, H \to E, E \to R, R \to R, R \to T$. However, this is too coarse an abstraction for compositional analysis. By the Step rule (Section 3) all four states, being reachable from the initial abstract state T, must belong to the final invariant, θ_i^*, for every process i. This abstract compositional invariant contains a global state where neighbors i, j are in state E, which violates the desired property of mutual exclusion between neighbors.

To tighten up the abstraction, we define a predicate A that is true for a node i if it "owns all adjacent forks" (i.e., if for every j adjacent to i, the fork variable on the edge (i, j) has value i). Note that this predicate occurs in the protocol, guarding the transition from state H to state E. The reachable abstract transitions at a node with at least one adjacent edge are shown in Figure 3(a). For an isolated node A is vacuously true as it has no adjacent forks; the transitions for such a node are shown in Figure 3(b).

The standard existential abstraction is used to compute these transitions. The concrete domain for node m is the set of local states of m, L_m. The abstract domain is the set $\{T, H, E, R\} \times \{\mathsf{A}, -\mathsf{A}\}$. The abstraction function, α_m, maps a local state s to an abstract state based on the super-state in s and the value of A in s. This induces a Galois connection (α_m, γ_m) connecting the two domains. There is an abstract transition from (abstract) state a to (abstract) state b if there exist local states x, y of node m such that $x \in \gamma_m(a)$, $y \in \gamma_m(b)$, and $T_m(x, y)$ holds.

Abstract Interference Transitions Figure 3 shows the abstract states reachable through step transitions. For the compositional analysis, we also have to consider how interference by nodes adjacent to node m affects the abstract states of node m. The concrete interference due to node k was defined in Section 3. Expanding the definition of $\mathit{intf}_{km}^{\theta}$, one gets that (u, v) is an interference transition for node m caused by node k, under a vector of assertions θ, if the following holds.

$$(\exists s, t : u = s[V_m] \ \wedge \ v = t[V_m] \ \wedge \ T_k(s, t) \ \wedge \ \theta_k(s[V_k])) \tag{1}$$

Here, states s, t are *joint* states of nodes m and k, representing an assignment of values to the variables in $V_m \cup V_k$.

By analogy, the interference of node k on the abstract state of m is defined as follows. An abstract transition (a, b) for node m is the result of interference by

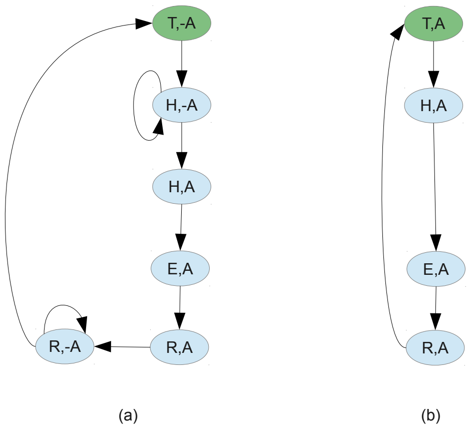

Fig. 3. Abstract State Transitions (a) for non-isolated nodes and (b) for an isolated node. The notation "$-\mathsf{A}$" indicates the negation of A. Green/dark states are initial.

process P_k under a vector of *abstract* assertions ξ if the following holds.

$$(\exists s, t : a = \alpha_m(s[V_m]) \ \wedge \ b = \alpha_m(t[V_m]) \ \wedge \ T_k(s,t) \ \wedge \ \xi_k(\alpha_k(s[V_k]))) \qquad (2)$$

Informally, this expression says that every abstract interference transition is witnessed by a concrete interference transition.

Computing Interference Surprisingly, *there is no non-trivial interference in the abstract DP protocol*! Informally, this is due to the following reason. The predicate A_m refers to the set of forks owned by m. In the concrete protocol, an adjacent node cannot take away ownership of forks from node m, nor can it grant ownership of forks to m. Hence, the value of A_m is unchanged by transitions at neighboring nodes. Those transitions cannot change the value of the internal state of node m, either. Formally, the maximum interference from k on to m, obtained by setting ξ_k to *true* in the defining expression (2), shows that the abstract state of m is unchanged by transitions of k.

Abstract Compositional Invariants We have just established that in the abstract domain, all interference is trivial. Hence, the compositional invariance calculation produces, for each process, only the set of states reachable via abstract step transitions. This is just the set of states shown in Figure 3. From it, one can read off the following invariant: for all nodes m, if E_m is true, then A_m is true. The corresponding concrete invariant is that for all nodes m, if E_m is true, then $\gamma_m(\mathsf{A}_m)$ is true. There cannot be a global state meeting this invariant where adjacent nodes m and n are each in state E. Otherwise, $\gamma_m(\mathsf{A}_m)$ and $\gamma_n(\mathsf{A}_n)$ will be true simultaneously, which is impossible – recall that A_m states that m owns *all* adjacent forks. Hence, the invariant suffices to show exclusion between neighbors.

Symmetry Reduction The abstract transition graphs are identical for all non-isolated nodes, and identical for all isolated nodes. It suffices, therefore, to have a single representative node for each class. Hence, the analysis of an arbitrary network, however irregular, can be reduced to the analysis (under abstraction) of *two* representative nodes. As this holds for all networks, the (abstract) compositional invariant calculated for a small representative network, one with each kind of node, defines a *parameterized proof* of safety.

Next Steps In the following sections, we build up the foundations required to show the soundness of this informal analysis. We give a definition of abstract symmetry, specialize it to the case of predicate abstraction, and show the consequences for symmetry reduction. We also show a completeness result generalizing the observation made for DP that its abstract transition graph is interference-free. We show that there is always an abstraction for which this is true if there exists a parameterized conjunctive invariant for the protocol.

A similar analysis to the one carried out here applies also to another Dining Philosophers protocol where there is always a distinguished node. In the case of a ring network, the process at the distinguished node, say P_0, chooses its forks in an order that is the reverse of that taken by other processes, for instance, in the order left;right instead of right;left. (This ensures the absence of deadlock.) The irregularity introduced by the distinguished process implies that the only structural balance relation for this ring is the trivial identity relation. However,in the semantic balance relation after abstraction, any two nodes are equivalent.

5 Local Similarity up to Abstraction

In this section, we develop the theory combining local abstraction and symmetry. In [19], we introduced the notion of a local similarity between nodes of a network. We refer to that as *structural* similarity here, to distinguish it from the semantic similarity notion to follow.

5.1 Structural Similarity and Balance

Two nodes in a network are structurally similar if they have the same color and there is a bijection between their neighborhoods. The neighborhood of a node is the set of its adjacent edges. The bijection should respect the edge color and the type of the node-edge connection. I.e., input edges should be mapped to input edges and output edges to output edges. The similarity between nodes is represented by a triple, (m, β, n), where m, n are nodes and β is the witnessing bijection. The set of all similarity triples forms a *groupoid*, a group-like structure with a partial composition operation.

A *structural balance* relation is a subset of this groupoid that induces a recursive similarity. The definition has a co-inductive form, rather like that of a bisimulation. If (m, β, n) is in the balance relation, then for every node k that points to m, there is a node l that points to n and a map δ, such that (k, δ, l) is in the balance relation, and β and δ agree on the edges common to m and k. We have the following theorem.

Theorem 1 ([19]) *Consider a structural balance relation B. For any program that is a valid assignment relative to B, the strongest compositional invariant θ^* is such that for any (m, β, n) in B, $[\theta^*_n \equiv \beta(\theta^*_m)]$.*

A program is a valid assignment relative to B if the assignment respects the symmetries in B: if (m, β, n) is in B, the transition relations and initial conditions for P_m and P_n must be related by β. The conclusion of the theorem says that for any state x in θ^*_m, the state y obtained from x by permuting edges according to β belongs to θ^*_n. The permutation is given by setting $y(\beta(e)) = x(e)$ for every edge variable e. Informally, this theorem says that the strongest local invariants for any pair of structurally similar nodes are themselves similar. Hence, it suffices to compute θ^* for a representative from each class of balance-equivalent nodes, as shown in [19].

5.2 Semantic Similarity and Balance

Structural balance is defined solely on the network structure. This can be limiting, as irregular networks have only trivial structural balance relations. The semantic notion of balance mixes together program and network structure. It requires balanced nodes to have similar transition relations and similar interference from other nodes.

The analysis in Section 4 relies on abstracting the local state of each node. After this is done one cannot, in general, define interference between nodes in terms of shared state. Therefore, it is necessary to abstract the definition of interference. It is convenient to think of "interference" as a primitive notion: for node k that points to m, there is a postulated interference relation intf^X_{km} that is a transition relation on the states of m, parameterized by a set X of states of k. This relation is used as usual in the fixpoint computation (Figure 2). We require that intf^X_{km} is monotonic in X to ensure that the least fixpoint is defined.

A *semantic balance* relation consists of triples (m, β, n) where m and n are nodes and β is a relation on the local state sets L_m and L_n. (Recall that in a structural balance relation, β is a bijection on edges.) As with structural balance, there is a clause that propagates local symmetry between m and n to corresponding neighbors. In the following, we use the notation $\langle\beta\rangle Y$ for the set $\{x \mid (\exists y : (x, y) \in \beta \wedge y \in Y)\}$ of predecessors of Y according to β.

Definition 1 ***(Semantic One-sided Balance)*** *A one-sided balance relation is a set of triples such that for every (m, β, n) in the balance relation*

1. *(initial-similarity) initial states are related by β: formally, $[I_m \Rightarrow \langle\beta\rangle I_n]$*
2. *(step-similarity) β is a safety-simulation from T_m to T_n*
3. *(interference-similarity) For every k that points to m, there is l that points to n and δ for which*
   (a) *(successive-balance) (k, δ, l) is in the balance relation, and*
   (b) *(agreement) For state sets X, Y such that $[X \Rightarrow \langle\delta\rangle Y]$, β is a safety-simulation from $intf^X_{km}$ to $intf^Y_{ln}$.*

A safety simulation (cf. [21]) R from T_1 to T_2 is defined as follows. For any $(s, t) \in R$ and any transition $(s, s') \in T_1$, there must be state t' such that $(t, t') \in T_2^*$ and $(s', t') \in R$. It is a form of simulation, weakened by the use of the reflexive transitive closure T_2^*.

Definition 2 ***(Two-sided Balance)*** *A two-sided balance relation is a one-sided balance relation that is closed under inverse; i.e., if (m, β, n) is in the relation, so is (n, β^{-1}, m).*

Theorem 2 *Every structural balance relation together with a valid assignment induces a two-sided balance relation.*

5.3 Symmetry Reduction

We consider the compositional fixpoint computation to be carried out in stages. The set of local states computed for node m at stage S is denoted θ^S_m. At the initial stage ($S = 0$), $\theta^S_m = I_m$. In a "step" stage, step transitions are applied for all nodes until no new states are generated. In an "interference" stage, interference transitions are applied for all nodes until no new states are generated. The main symmetry theorem below shows that for a balance triple (m, β, n), at every stage, the local states generated at m are related through β to the local states generated by n.

Theorem 3 *(Main Symmetry Theorem) Given a semantic balance relation, at every fixpoint stage S in the compositional invariance calculation, for all (m, β, n) in the balance relation, $[\theta^S_m \Rightarrow \langle\beta\rangle\theta^S_n]$ holds.*

Proof. The proof is by induction on stages.

Consider the initial stage. Here, the θ-values are the initial state sets for each process. The claim follows from the initial-similarity condition.

Suppose, inductively, that the claim holds for stage S for all triples in the balance relation. Consider stage $S+1$ and the triple (m, β, n). For node m, let s be a state generated at stage $S+1$ that is an immediate successor of a state u in θ_m^S. From the induction hypothesis applied to u, there is a state v in θ_n^S related by β to u. There are two cases, based on the type of stage $S+1$.

(1) Suppose that $S+1$ is a step stage, so that s is a successor by T_m. By step-similarity, there is a state t reachable by a sequence of T_n moves from v that is β-related to s. As the stage $(S+1)$ calculation closes-off under step successors, t belongs to θ_n^{S+1}.

(2) Suppose, instead, that $S+1$ is an interference stage, and that s arises from an interference transition by node k that points to m. By balance, there is l that points to n and δ such that (k, δ, l) is in the balance relation.

Let $X = \theta_k^S$ and $Y = \theta_l^S$. By the inductive claim applied to k and l, $[X \;\Rightarrow\; \langle\delta\rangle Y]$. From the agreement condition, β is a safety-simulation between intf_{km}^X and intf_{ln}^Y. Thus, for the transition from u to s, which is in intf_{km}^X, there is a matching sequence of intf_{ln}^Y-transitions from v to a state t that is related by β to s. As the stage $(S+1)$ calculation closes-off under interference successors, t belongs to θ_n^{S+1}.

We have considered only the important case, where s is an immediate successor of a state in the previous stage. The case of a non-immediate successor within the same stage may be shown by induction within a stage based on the length of its derivation path within that stage. □

Corollary 1 *Consider a two-sided semantic balance relation where every triple is based on a one-to-one relation. For any (m, β, n) in the balance relation and every fixpoint stage S, $[\theta_m^S \;\equiv\; \langle\beta\rangle\theta_n^S]$.*

Proof. The direction from left-to-right follows from Theorem 3. As the balance relation is two-sided, it includes the triple (n, β^{-1}, m). By Theorem 3 for this triple, $[\theta_n^S \;\Rightarrow\; \langle\beta^{-1}\rangle\theta_m^S]$. Hence, $[\langle\beta\rangle\theta_n^S \;\Rightarrow\; \langle\beta\rangle\langle\beta^{-1}\rangle\theta_m^S]$. As β is one-to-one, this implies that $[\langle\beta\rangle\theta_n^S \;\Rightarrow\; \theta_m^S]$. □

Based on Theorem 3 and Corollary 1, one may symmetry-reduce the fixpoint calculation as follows, using a procedure defined in [19]. Consider a balance relation that is closed under inverse and composition. This defines a symmetry groupoid. The orbit of the groupoid, i.e., the set of pairs (m, n) such that there is a β for which (m, β, n) is in the balance relation, is an equivalence relation. For each balance-equivalence class, one chooses a representative node. The fixpoint calculation is carried out only for the representative nodes. The value of θ_i^S for a non-representative node i, which may be needed to calculate interference, can be computed by Corollary 1 as $\langle\beta\rangle\theta_r^S)$, where r is the representative for node i and (i, β, r) is a triple linking i to r.

6 Local Predicate Abstraction and Symmetry Reduction

This section connects the general concepts from Section 5 with the predicate abstractions used in the informal treatment in Section 4.

6.1 Local Domain Abstraction

We consider the effect of a general domain abstraction before specializing to predicate abstraction. The effect of a domain abstraction at each node is to construct an abstract network, which has the same structure as the original one, but with a different state space at each node. We refer to the abstract network for network N as $\overline{N}$ and refer to the abstract counterpart of node m as $\overline{m}$. In the following, we show how to connect the two networks using a balance relation. This relation induces a connection between the compositional invariants computed on the concrete and abstract networks.

A *local abstraction* is given by specifying, for each node m an abstract domain, D_m, and a total abstraction function, $\alpha_m : L_m \to D_m$. This induces a Galois connection on subsets, which we also refer to as (α_m, γ_m): $\alpha_m(X) = \{\alpha_m(x) \mid x \in X\}$, and $\gamma_m(A) = \{x \mid \alpha_m(x) \in A\}$.

The abstract set of initial states, $\overline{I}_m$ is given by $\alpha_m(I_m)$. (For a simpler notation, we denote this set as $\overline{I}_m$ rather than $I_{\overline{m}}$.) The abstract step transition, $\overline{T}_m$, is obtained by existential abstraction: there is a transition from (abstract) state a to (abstract) state b if there exist x, y in L_m such that $\alpha_m(x) = a$, $\alpha_m(y) = b$, and $T_m(x, y)$ holds. An abstract transition (a, b) is the result of interference by node k from state set Y, that is, $(a, b) \in \overline{intf}^{Y}_{km}$, if the following holds.

$$(\exists s, t : \alpha_m(s[m]) = a \ \wedge\ \alpha_m(t[m]) = b \ \wedge\ T_k(s, t) \ \wedge\ \alpha_k(s[k]) \in Y) \qquad (3)$$

Theorem 4 *The set of triples of the form $(m, \alpha_m, \overline{m})$ is a one-sided semantic balance relation connecting the concrete network N to the abstract network $\overline{N}$.*

Proof. We have to check the conditions for one-sided balance. Initial-similarity follows by the definition of $\overline{I}_m$. Step-similarity holds as the abstract transition relation is the standard existential abstraction. For a node k that points to m, we use $\overline{k}$ as its corresponding node, which points to $\overline{m}$, and let $\delta = \alpha_k$. Then, $(k, \delta, \overline{k})$ is also in the balance relation. The agreement condition follows from the analogy between the concrete and abstract interference conditions. Specifically, if (x, y) is a transition in $intf^{X}_{km}$, there exist joint (concrete) states s, t such that $s[k] \in X$, $s[m] = x$, $t[m] = y$ and $T_k(s, t)$ all hold. Thus, we get that $\alpha_k(s[k]) \in \alpha_k(X)$, $\alpha_m(s[m]) = \alpha_m(x)$, $\alpha_m(t[m]) = \alpha_m(y)$ and $T_k(s, t)$ all hold. Let Y be such that $[X \ \Rightarrow\ \langle \alpha_k \rangle Y]$. Then we get that $\alpha_k(s[k]) \in Y$. Hence, from definition (3) above, the pair $(\alpha_m(x), \alpha_m(y))$ is in $\overline{intf}^{Y}_{km}$, which establishes the agreement condition. □

As a consequence, from Theorem 3, we obtain the following corollary. This corollary shows that the strongest compositional invariant obtained on $\overline{N}$ can be concretized to a compositional invariant for the network N. Abstraction may lose some precision, but this comes at the potential gain of local symmetry.

Corollary 2 *Let θ^* and ξ^* be the strongest compositional invariants for the concrete and abstract networks, respectively. Then (1) for every m, $[\theta^*_m \Rightarrow \gamma_m(\xi^*_{\overline{m}})]$ and (2) the vector $(m : \gamma_m(\xi^*_{\overline{m}}))$ is a compositional invariant for the concrete network.*

Proof (sketch). The first conclusion follows from Theorem 3 applied to the (disjoint) union of N and $\overline{N}$, and the Galois connection between α_m and γ_m.

The second conclusion follows by reasoning with the conditions of compositional invariance and the Galois connection. For simplicity, we suppose that the abstract domain is flat (i.e., ordering is equality), so that the condition $x \in \gamma_k(Y)$ is equivalent to $\alpha_k(x) \in Y$. (Non-flat domains may be handled by taking downward closures in the computation of ξ^*, a complication we omit here.)

Define $Z_m = \gamma_m(\xi^*_m)$ for all m. We show that Z_m satisfies the compositional invariance conditions for node m by considering them in turn. For $x \in I_m$, $\alpha_m(x)$ is in $\overline{I}_m$. By initiality, $\alpha_m(x)$ is in ξ^*_m, so that $x \in Z_m$. For $x \in SP_m(T_m, Z_m)$, there is $y \in Z_m$ such that $T_m(x, y)$. Hence, $(\alpha_m(x), \alpha_m(y))$ belongs to $\overline{T}_m$ and $\alpha_m(y)$ is in ξ^*_m. The step condition for $\overline{T}_m$ ensures that $\alpha_m(x)$ is in ξ^*_m; hence, $x \in Z_m$. Similar reasoning proves the case for interference transitions. □

6.2 Local Predicate Abstraction

Local predicate abstraction maps the local state of a node to the valuation of a set of predicates on the local state. For simplicity, we fix a set of predicates, $\mathcal{P}$, which can be interpreted over all nodes.

There is a natural Galois connection, denoted (α_m, γ_m), established by $\mathcal{P}$ over a node m. For a local state s of node m, the abstraction function $\alpha_m(s)$ maps s to a set of literals giving the valuation of the predicates in $\mathcal{P}$ on s. This is the set $\{(p \equiv p(s)) \mid p \in \mathcal{P}\}$. The abstraction is extended naturally to sets of concrete states. The concretization function, γ_m, maps a set X of of sets of literals to the set of local states given by $\{s \mid \alpha_m(s) \in X\}$. Given this abstraction, the abstract forms of the transition relation and interference are as defined in Section 6.1.

Corollary 2 establishes that the compositional invariant calculated using the abstract initial states, abstract step and abstract interference transitions defined over $\mathcal{P}$ is, as interpreted through the γ_m functions, a valid compositional invariant for the concrete system.

We now show that, with the right choice of predicates, the induced abstract network is (a) fully locally symmetric, (b) free of interference, and (c) adequate.

This result is similar in spirit to a completeness theorem of Kesten and Pnueli [17] for abstraction over the global state space. The key difference in the following theorem is in its treatment of compositionality and symmetry, which are not covered by the Kesten-Pnueli result.

Theorem 5 *For a fixed-size process network, any inductive invariant of the form $(\forall i : \theta_i)$, where each θ_i is local to process P_i, can be established by compositional reasoning over a uniform abstract Boolean network.*

Proof (sketch). By a result in [18], the strongest compositional invariant, θ^*, is such that $[\theta_i^* \;\Rightarrow\; \theta_i]$, for all i.

The single predicate symbol is B. (A helpful mnemonic is to read B as "black" and $(\neg B)$ as "white".) The abstraction function α_i maps a local state x of L_i to B, if $\theta_i^*(x)$ holds, and to $(\neg B)$ otherwise. Let ξ^* be the strongest compositional invariant computed for the abstract network. By Corollary 2, $[\theta_i^* \;\Rightarrow\; \gamma_i(\xi_i^*)]$. This implies that the abstract state $(B = \mathit{true})$ is in ξ_i^* for all i.

We show that the implication is, in fact, an equality; i.e., that the other abstract state, $(B = \mathit{false})$, does not belong to ξ_i^* for any i. As $[I_m \;\Rightarrow\; \theta_m^*]$, it follows that $[\alpha_m(I_m) \;\Rightarrow\; \alpha_m(\theta_m^*)]$, i.e., that $[\overline{I}_m \;\Rightarrow\; B]$. Hence, initially, only the state B is in ξ_m^*. Suppose, inductively, that this is the case at the S'th stage. Consider an abstract step transition that introduces the state $(\neg B)$. This must have a concrete step transition $T_m(x, y)$ as a witness where $\theta_m^*(x)$ holds but $\theta_m^*(y)$ does not. This is impossible by the step constraint for θ^*. Similarly, one can establish the impossibility of an abstract interference transition of this kind, so that the only interference is the trivial (B, B) transition. Hence, $\xi_m^* = \{B\}$ for all m, so that the concrete invariant induced by ξ^* is just θ^*. This establishes adequacy: any property implied by the original invariant can be shown with the concretized abstract invariant.

We now show that, in the abstract network, any two isolated nodes and any two non-isolated nodes are balanced. The balance relation consists of triples (m, β, n) where β is the partial bijection $B \mapsto B$. Initial-similarity holds as the only initial state is B. Step similarity holds as, by the reasoning above, the only abstract reachable step transition is (B, B). The reasoning so far is sufficient to show that any two isolated nodes are balanced; the rest of the proof applies to the case where m and n are not isolated nodes. For a node k that points to m, choose its corresponding node l arbitrarily from the nodes pointing to n – there is at least one such node as n is not isolated. Then k and l are non-isolated and (k, δ, l) is in the balance relation with the bijection δ which maps B to B. Consider X, Y as in the agreement condition. It suffices to consider $Y = \delta(X)$, by the monotonicity of interference relations. As δ is defined for all elements of X it cannot include $(\neg B)$; therefore, X must be either $\emptyset$ or $\{B\}$. Thus, $Y = \delta(X)$ is correspondingly $\emptyset$ or $\{B\}$. Expanding the definition of abstract interference, it follows from the non-interference condition for θ^* that β is a safety simulation between intf_{km}^X and intf_{ln}^Y. □

A consequence of Corollary 2 is that if all members of a parameterized family of networks can be reduced to a finite set of abstract networks (over a fixed set of predicates), the compositional invariants computed for the abstract networks concretize to a parametric compositional invariant for the entire family. The following theorem shows that there is always a "right" choice of predicate for which this is true.

Theorem 6 *For a parameterized family of process networks, any compositional invariant of the form $(\forall i : \theta_i)$, where each θ_i is local to process P_i, can be established by compositional reasoning over a small uniform abstract Boolean network.*

Proof (sketch). The difference between the statement of this theorem and Theorem 5 is that the assumed invariant is compositional. This implies that the compositionality conditions hold for the given θ_i for every instance. Hence, the predicate symbol B in the proof of Theorem 5 has an interpretation that is uniform across all instances.

The proof of Theorem 5 shows that in the abstract network, there are at most two classes of nodes: those that are isolated and those that are not. Hence, there is a cutoff instance N whose abstract network $\overline{N}$ (over B) exhibits all the classes of nodes that can arise. By Theorem 3, the concretized form of the compositional abstract invariant for $\overline{N}$ is a compositional invariant for every larger instance of the family. □

6.3 Reviewing the Dining Philosophers Analysis

We can now review the informal analysis of Section 4 in terms of these theorems. The per-node abstraction with predicate A is an instance of the local predicate abstraction discussed in Section 6.2. From Corollary 2, the abstract compositional invariant, when concretized, is a compositional invariant for the concrete system. This establishes the correctness of $(\forall m : E_m \Rightarrow \gamma_m(\mathsf{A}_m))$ as a concrete invariant.

For the abstract network, the candidate balance relation is $\{(m, id, n)\}$ where id is the identity relation, and m, n are both isolated nodes or both non-isolated nodes. Using the definitions of abstract transitions and interference, one can check that this meets the conditions for a two-sided balance relation (Definitions 1 and 2). The orbit of this balance relation has just two classes so, from Corollary 1, it suffices to consider two representative nodes for the analysis. Since the calculation for the representative node is identical across networks, we may conclude that the computed invariant applies to all networks.

The completeness theorems show that the phenomenon observed in the informal analysis of the Dining Philosophers protocol (Section 4) is not an isolated case. Any compositional invariant of a parameterized family can be obtained through a local predicate abstraction that induces complete local symmetry and only trivial interference in the abstract network.

7 Summary and Related Work

The seminal work on symmetry reduction in model checking [12, 4, 16] and its many refinements base the theory on the *global* symmetries of a Kripke structure, expressed as a group of automorphisms. For a distributed program, these symmetries (as shown in [12]) are lower-bounded by the symmetries of the process interconnection network. In fact, for most interesting protocols, the symmetries are also upper-bounded by the group of symmetries of the process network. This implies that symmetry reduction works well for networks with a complete interconnection or a star shape. (This is typically a client-server structure, although, as the example in Section 2 shows, not all client-server protocols fit the assumptions made in [12].) For other networks, most notably ring, mesh and torus networks, the global automorphism group has size at most linear in the number of nodes of the network; hence, the available symmetry reduction is also at most a linear factor. This is not particularly helpful if, as is often the case, the Kripke structure is exponential in the size of the network.

Motivated by this problem, we introduced in [19] the notion of local symmetries. Regular networks, such as the complete, star, ring, mesh and torus networks have a high degree of local similarity: intuitively, the network "looks the same" from nearly every node. We show that this results in symmetry reduction for compositional methods. Although the compositional invariance calculation is polynomial, requiring time $O(N^3)$ for a network of N nodes, local symmetry does have a significant effect, in two ways. First, for many regular networks, the calculation time becomes independent of N after symmetry reduction. Second, it is possible to derive parametric invariants if local symmetry reduces each member of a family of networks to a fixed set of representatives.

As discussed in the introduction, this symmetry notion is, however, not applicable to irregular networks. In this work, we show that it is possible in many cases to overlay a local similarity structure on an irregular network, by using an appropriate abstraction over node neighborhoods. The theoretical drawback is that using abstraction generally results in weaker invariants. On the other hand, we show that for the Dining Philosophers protocol, the invariant calculated by abstraction suffices to prove mutual exclusion. Moreover, the completeness result ensures that, for any compositional invariant, there is always an appropriate predicate abstraction. Hence, we conjecture that abstraction will suffice for most practical analysis problems.

Other compositional reasoning methods, such as those based on alternative assume-guarantee rules [15] or on automaton learning [14, 6] should also benefit from local symmetry reduction; working out those connections is a subject for future work. It should be noted that there are other techniques (e.g., [13]) which enhance global symmetry in certain cases where the original protocol is only minimally globally symmetric. In the current work, we have instead applied local, rather than global, symmetry reduction techniques; local symmetries appear to be more widely applicable. A particularly intriguing outcome of the analysis of the Dining Philosophers protocol is that one can show a parametric invariant, one which holds for all networks. Parameterized safety analysis is undecidable in general [2]. There is a large variety of analysis methods, such as those based on well-quasi-orders (e.g., [1]) or on iterating transducers (e.g., [10]), each of which works well on a class of problems.

We discuss two methods which are the closest to the point of view taken here. The first is the "network grammar" method from [20]. A family of networks is described by a context-free network grammar. A choice of abstract process is made for each non-terminal in the grammar. This results in a set of model-checking constraints, which, if solvable, give a parametric proof of correctness. This technique is applicable to regular networks (rings, trees) that have a compact grammar description. The second method is that of environment abstraction [5]. This method chooses the point of view of a single process, abstracting the rest of the network. There is a certain similarity between the generic process used for environment abstraction and the single representative process used in our work. However, there is a difference in how the network abstraction is defined (non-compositionally for environment abstraction) and the method has not been applied to irregular networks.

The connections made in [19] between local symmetry, compositionality and parametric verification are extended in here to irregular networks. The crucial observation is that local abstraction can make an irregular network appear regular, facilitating symmetry reduction. The application to versions of the Dining Philosophers protocol and the completeness results suggest that these connections are worth further study. In ongoing work, we are examining how well abstraction works for other protocols. An interesting question is whether appropriate abstraction predicates, such as the predicate A from Section 4, can be discovered automatically. It is possible that automatic methods that discover auxiliary predicates to address incompleteness (e.g., [7, 8]) can be adapted to discover predicates for abstraction. A particularly interesting question for future work is to investigate parametric proofs of protocols on dynamic networks, i.e., networks where links and nodes can fail or appear, a domain that is interesting because of its connections to fault tolerance and ad-hoc networking (cf. [11]).

References

1. P. A. Abdulla, K. Cerans, B. Jonsson, and Y.-K. Tsay. General decidability theorems for infinite-state systems. In *LICS*, pages 313–321. IEEE Computer Society, 1996.
2. K. R. Apt and D. Kozen. Limits for automatic verification of finite-state concurrent systems. *Inf. Process. Lett.*, 22(6):307–309, 1986.
3. K. M. Chandy and J. Misra. *Parallel Program Design: A Foundation.* Addison-Wesley, 1988.
4. E. M. Clarke, T. Filkorn, and S. Jha. Exploiting symmetry in temporal logic model checking. In *CAV*, volume 697 of *LNCS*, pages 450–462, 1993.

5. E. M. Clarke, M. Talupur, and H. Veith. Environment abstraction for parameterized verification. In *VMCAI*, volume 3855 of *LNCS*, pages 126–141, 2006.
6. 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.
7. A. Cohen and K. S. Namjoshi. Local proofs for global safety properties. In *CAV*, volume 4590 of *LNCS*, pages 55–67. Springer, 2007.
8. 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.
9. P. Cousot and R. Cousot. Automatic synthesis of optimal invariant assertions: mathematical foundations. In *ACM Symposium on Artificial Intelligence & Programming Languages*, Rochester, NY, ACM SIGPLAN Not. 12(8):1–12, Aug. 1977.
10. D. Dams, Y. Lakhnech, and M. Steffen. Iterating transducers. *J. Log. Algebr. Program.*, 52-53:109–127, 2002.
11. G. Delzanno, A. Sangnier, and G. Zavattaro. Verification of ad hoc networks with node and communication failures. In *FMOODS/FORTE*, volume 7273 of *LNCS*, pages 235–250. Springer, 2012.
12. E. Emerson and A. Sistla. Symmetry and model checking. In *CAV*, volume 697 of *LNCS*, pages 463–478, 1993.
13. E. A. Emerson, J. Havlicek, and R. J. Trefler. Virtual symmetry reduction. In *LICS*, pages 121–131. IEEE Computer Society, 2000.
14. D. Giannakopoulou, C. S. Pasareanu, and H. Barringer. Assumption generation for software component verification. In *ASE*, pages 3–12. IEEE Computer Society, 2002.
15. A. Gupta, C. Popeea, and A. Rybalchenko. Predicate abstraction and refinement for verifying multi-threaded programs. In *POPL*, pages 331–344. ACM, 2011.
16. C. Ip and D. Dill. Better verification through symmetry. *Formal Methods in System Design*, 9(1/2):41–75, 1996.
17. Y. Kesten and A. Pnueli. Verification by augmented finitary abstraction. *Information and Computation*, 163(1):203–243, 2000.
18. K. S. Namjoshi. Symmetry and completeness in the analysis of parameterized systems. In *VMCAI*, volume 4349 of *LNCS*, pages 299–313, 2007.
19. K. S. Namjoshi and R. J. Trefler. Local symmetry and compositional verification. In *VMCAI*, volume 7148 of *LNCS*, pages 348–362, 2012.
20. Z. Shtadler and O. Grumberg. Network grammars, communication behaviors and automatic verification. In *Automatic Verification Methods for Finite State Systems (1989)*, volume 407 of *LNCS*, pages 151–165, 1990.
21. R. J. Trefler and T. Wahl. Extending symmetry reduction by exploiting system architecture. In *VMCAI*, LNCS, pages 320–334. Springer, 2009.