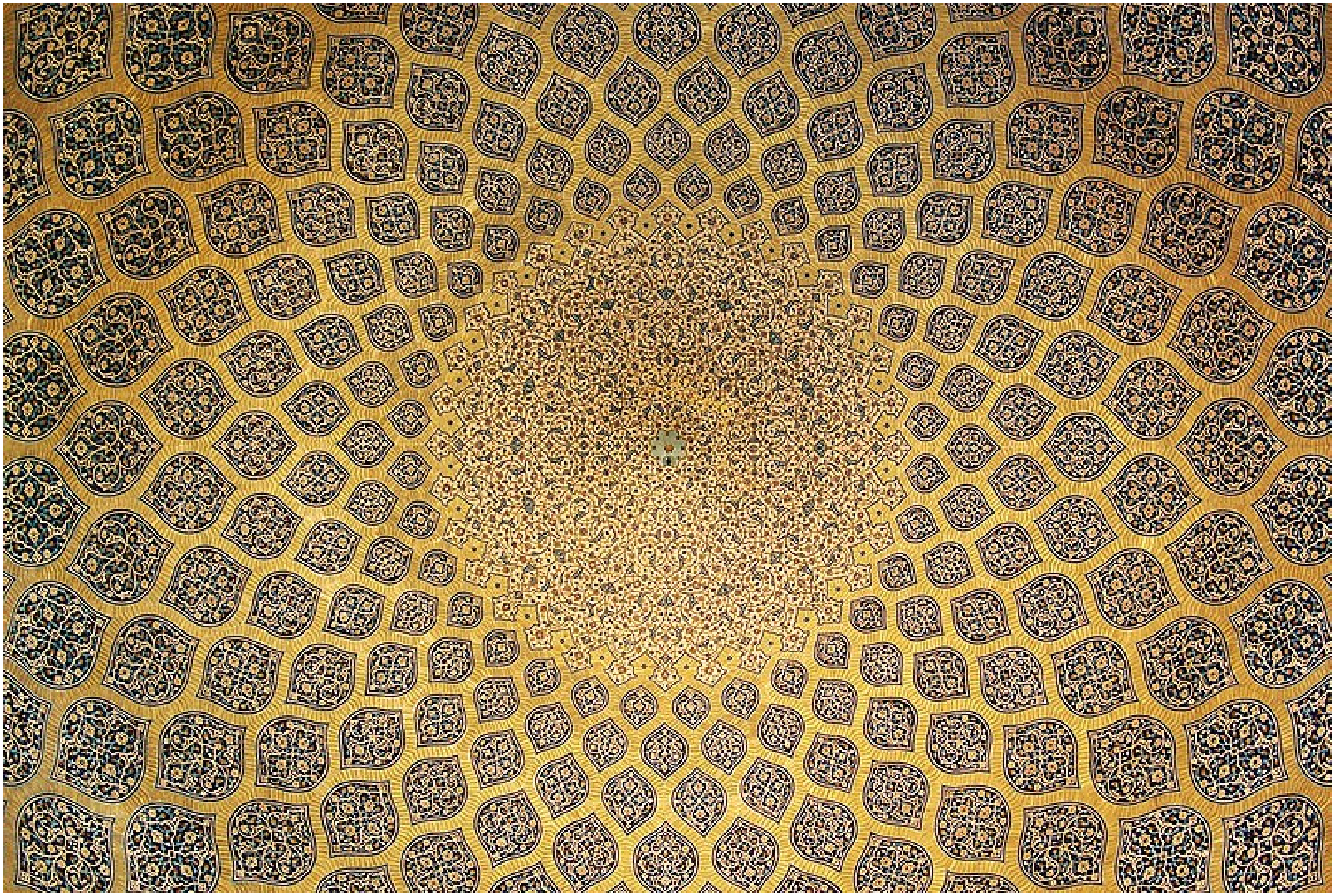


Symmetry Reduction For Dynamic Process Networks

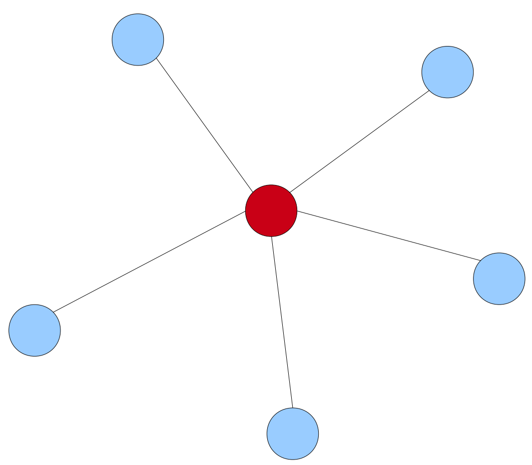
Kedar Namjoshi (Bell Labs) and Richard Trefler (University of Waterloo)

FRIDA 2015

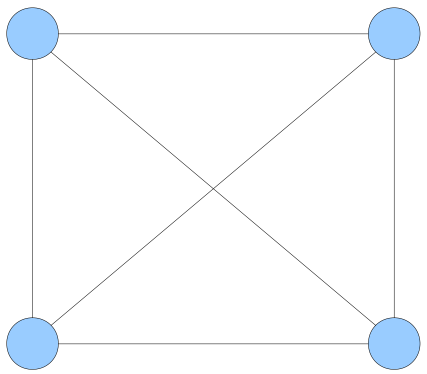
(Photo: Mosque in Isfahan, Iran. By Phillip Maiwald (Nikopol))

Network Topology and Symmetry

- A concurrent program often exhibits symmetries in its state space
- Symmetry: a transformation that leaves *global* structure unchanged
- E.g. any onto function on the star network that maps the centre to the centre.



N+1 nodes; |Aut(G)| = N!



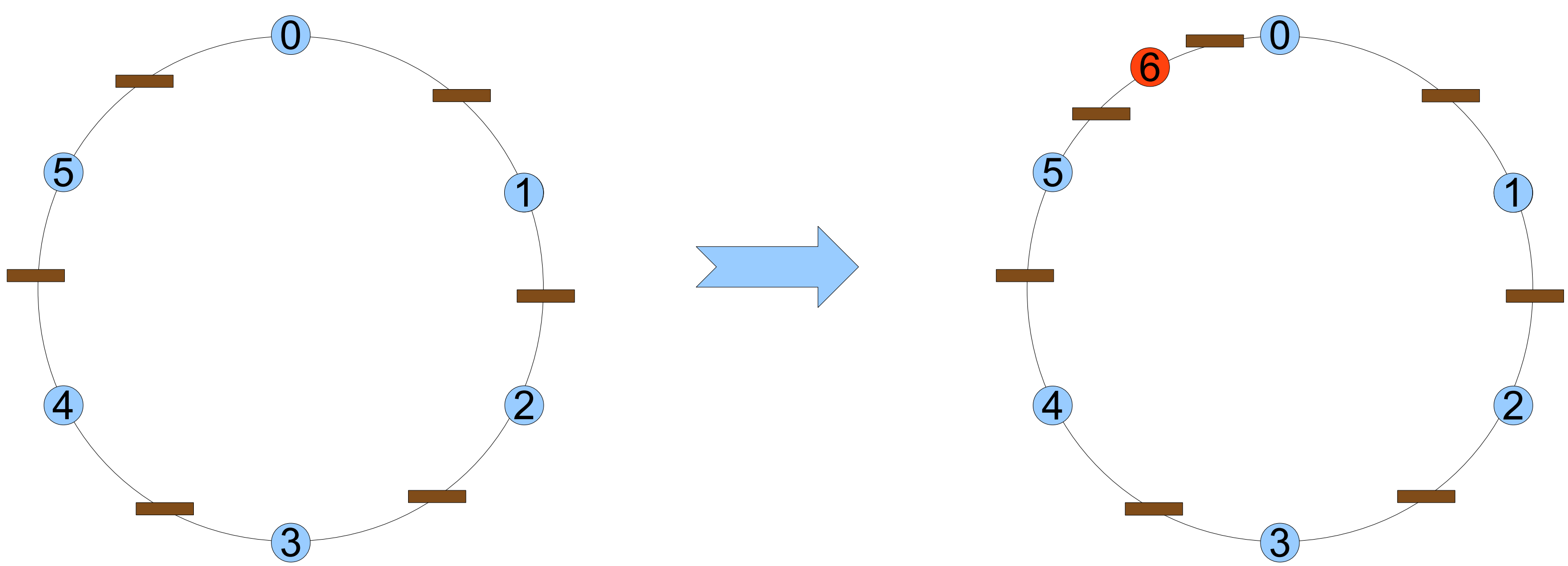
N nodes; |Aut(G)| = N!

Symmetry Reduction in Program Analysis

- Process network symmetries induce state-space symmetries
- Key idea: a network/system symmetry maps a network state to an `equivalent' state
- Network symmetries induce equivalence classes of network states
- Analyze only one representative of each equivalence class
- Symmetry reduction results in a smaller, equivalent state space
- The best case (exponential reduction) is that of star and complete graphs

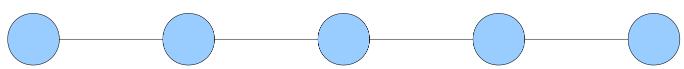
Dynamic Networks

- Nodes (and the processes associated with them) and edges may be inserted or deleted *during* process execution
- E.g., sensor networks, communication protocol, routing protocols may be modeled as dynamic networks
- IP routers may fail and be (re)activated during protocol execution

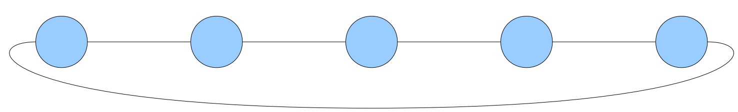


Symmetry and Protocol Analysis

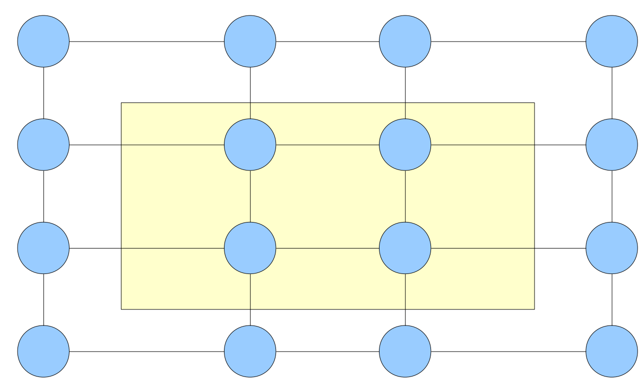
- Global symmetry definitions do not apply to dynamic networks. E.g. a ring of six nodes is not globally symmetric to a ring of 5 nodes
- Many fixed, regular topologies have little global symmetry
- The state-space reduction can be at most polynomial, or even constant
- Examples: pipeline, ring, mesh, torus, hypercube, etc.



N nodes; |Aut(G)| = 2



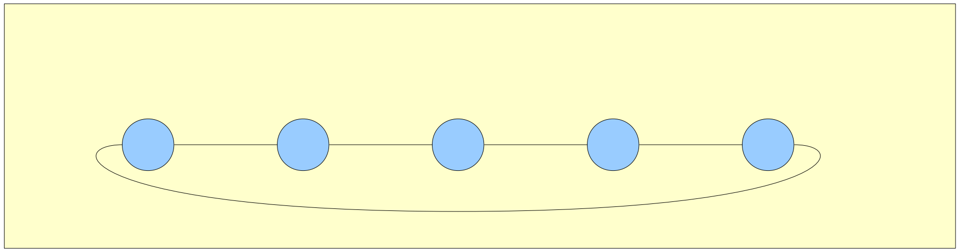
N nodes; |Aut(G)| = N



N*N nodes; |Aut(G)| = 8

Local Symmetry in Graphs

- Two nodes are locally symmetric if they have similar neighborhoods, and the neighborhoods are also symmetric
- Individual nodes in the ring of size 5 are locally symmetric to each other and to the nodes in the ring of size 6



Symmetry in Dynamic Models: Key Ideas

- Nodes are locally symmetric if their neighborhoods are similar and their (local) processes are similar
- Compute the inductive-compositional-invariant for a representative process --- the local reachable state space
- The invariants are isomorphic for locally symmetric processes

-

Compositional Safety Analysis

- Given: a dynamic process network
- To compute: a global inductive invariant: (forall G, n: Θ(G, n))
- Where quantification is over dynamic network change
- G is a network graph
- n is a network node in G
- θ(G, n) is an invariant (the reachable local states) of process n and its neighborhood in network G

Inductive Invariant: θ(G, n)

fix point calculation: program transitions

- (Initially) All initial states of the process at (G, n) are in Θ(G, n)
- (Step) Θ(G, n) is closed under transitions of process P(n)
- (Non-interference) if node m points to node n in G, then the set of joint states local to m and n are closed under transitions of m

  So Θ(G, n) is closed under transitions of m from states in Θ(G, m)

Dynamic Network Changes

Adversarial Actions: topology changes include

- Addition/removal of a node
- Addition/removal of an edge
- Addition/removal of a link between a node and an edge

Plus program response to adversarial change

Inductive Invariant

Fix point calculation: network disruptions
Link introduction

- Let a be an assignment to the neighborhood of n in G, and let v be a valuation to the edge e in G
- Here a is a valuation for P(n)
- Transition link(n, e) changes G to G' by adding a connection between node n and edge e
- If a is in Θ(G, n) and link(n,e) is a transition from the joint state (a, v) resulting in (a', v') then (a', v') is in Θ(G', n)
- Similar rules for link removal, node addition, node removal, edge addition, edge removal

Compositional Safety

- **Theorem:** If the compositional constraints hold, the assertion (forall G, n: Θ(G, n)) is an inductive invariant of the dynamic network
- The constraints, non-dynamic and dynamic, form a set of simultaneous implications, all monotone in Θ
- By monotonicity there is a least solution, the strongest compositional invariant, Θ*
- The strongest non-dynamic compositional invariant (calculated without dynamic transitions) is a subset of the dynamic compositional invariant

Local Reasoning in a Nutshell

- Move from global to local analysis
- Analyze each component only within the context of its neighborhood
- Symmetry is recursive similarity among local neighborhoods
- For parametrized families of locally symmetric protocols, similarities are preserved across dynamic network change
- Local symmetries are described by `balance' relation – B.

Local Symmetry

- Local symmetries defined by balance relation B with entries (m, β, n)
- Structural properties of B:

  identity: (m, β, m) is in B for all nodes m

  (inverse): if (m, β, n) is in B then so is (n, β', m)

  (transitivity) if (m, β, k) and (k, γ, n) are in B then so is (m, (γ β), n)

Balance Relations

- A balance relation ensures recursive similarity: it is like a strong bisimulation
- For every triple (m,β,n) in a balance relation B:
  - Nodes m and n are locally similar
  - Neighbors of m are matched by neighbors of n
  - Relation is preserved recursively through corresponding neighbors
- Structural symmetry used to define recursive, computational symmetry
- (Symmetry defined by automorphism encodes local symmetry)

Local Symmetry

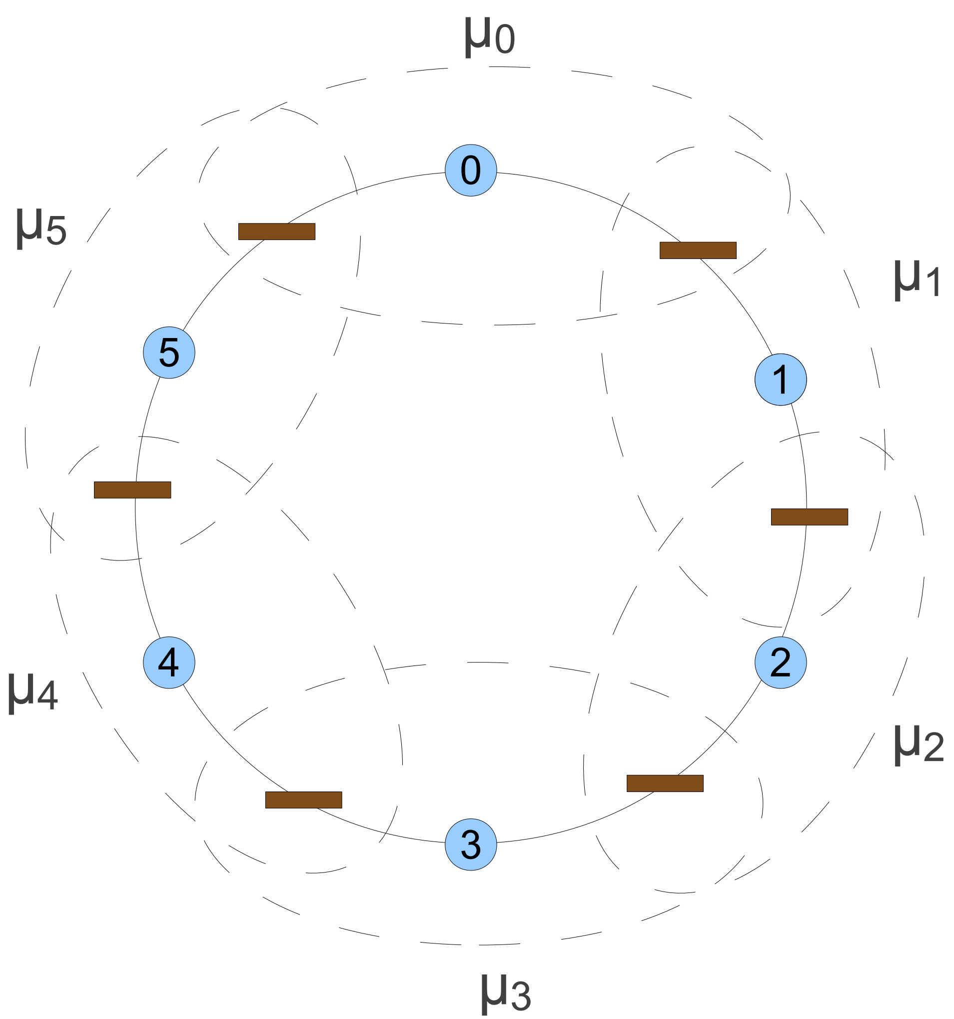
- Computational properties:

  (initial states) if (m, β, n) is in B then every initial state of P_m is related to an initial state of P_m (and vice versa)

  (transition) if (m, β, n) is in B then every transition of P_m is matched either by a transition of P_n or by a local interference transition of a neighbor of P_n (and vice versa)

  (interference) if (m, β, n) is in B and the local state of P_m changes due to interference from a neighbor P_q, then P_n has a matching neighbor P_r and either P_r has a matching interference transition or P_n has a matching transition

Balancing the Ring



Non-Dynamic Symmetry Theorem

- **Theorem:** For m and n related by local symmetry B, the local states of n, reachable in G, are locally symmetric to the local states of m, reachable in G
- Corollary: if the reachable states of m satisfy an invariant property Prop, and Prop on m is symmetric to Prop on n, then the reachable states of n satisfy Prop

Dynamic Symmetry Theorem

- **Theorem:** For (G, m) and (H, n) related by local symmetry B, the local states of n, reachable in H, are locally symmetric to the local states of m, reachable in G.
- Corollary: if the reachable states of m satisfy an invariant property Prop, and Prop on m is symmetric to Prop on n, then the reachable states of n satisfy Prop

Safety in Dynamic Action

- **React Assumption:** any local reaction to dynamic change preserves the non-dynamic invariance (e.g., the reaction "reboots" to initial state)
- Σ^* --- the strongest non-dynamic compositional invariant
- From any local state (s, m) in $\Sigma^*(G, m)$, if (s', m) results from dynamic change at (s, m), then (s', m) is in $\Sigma^*(G', m)$
- **Theorem:** Under the React Assumption, the strongest compositional invariant for the non-dynamic and the dynamic systems are identical

Application to dynamic network protocols

1. Define symmetry B for the network family with finitely many equivalence classes
2. Find representative network instance, R, whose nodes cover all equivalence classes
3. Compute the strongest non-dynamic invariant on R
4. Check the React assumption on the network family. (Any dynamic change from a reachable state results in a reachable state represented by the representative instance.)

Local Symmetry Example: Dynamic Dining Philosophers

- Each Philosopher: Thinking, Hungry, Eating, Release
- Neighbors m and n share fork var with value: {\bot, m, n}
- Thinking – does not acquire forks, may transit to Hungry at any time
- Hungry – acquire/release forks, transit to Eating only if all forks owned
- Eating – does not release forks, transit to Release at any time
- Release – sets each owned fork to \bot, transit to Thinking when all forks are unowned
- Safety property: If a Philosopher is eating the philosopher owns all its forks

Dining Philosophers Dynamic Changes

- Potential problem: Adding an edge between two Eating Philosophers would violate the key safety property
- Link addition: if Philosopher is Eating and a new link to the Philosopher is added, then the Philosopher responds by moving to Hungry
- Isolated nodes are added in local state Thinking
- Link removal, edge addition and removal, and node removal are straightforward

Example: Dynamic Dining Philosophers

- Dining philosophers model where a philosopher may voluntarily surrender a fork (implies no deadlock)
- ((G, m), β, (H, n)): local state (T, H, E, R) in m is the same in n; and node m owns all its forks, if and only if, node n owns all its forks
- Local states in (G, m) are stuttering similar to local states in (H, n)
- Representative network instances with 2 nodes
- Assertion: for every reachable local state of m, if m is in state E, then m owns all its forks
- Symmetry theorem: assertion holds for entire network family

AODVv2: routing in a dynamic network

- Establish routes from O to T in a network of nodes/edges
- Intermediate nodes maintain routes to O and routes to T
- Nodes/edges may come and go
- Neighbor connections monitored by each node
- Route discovery from O associated with seq_num established by O --- each new route increases the seq_num at O

AODVv2: routing in a dynamic network

- Node n prefers route back to O through m, rather than through m', if the route through m has a higher seq_num than the route at m', or if the seq_num at m is equal to the seq_num at m' but the cost (distance) of the route at m is less than the cost (distance) at m'

- Key safety property: in any reachable global state, the combined routing tables of the nodes do not form a routing loop: to send a message to O, node G send the message to H, and node H send the message to G

Automating Analysis of AODVv2 --- challenges 1

- Modeling of variables, constants and timers

  e.g. seq_num, hop_count variables have large (finite?) ranges

  constants, e.g., MAX_SEQ_NUM_LIFETIME must be set `appropriately' (or modeled with non-determinism)

  (bounded) timers used to count freshness of routing information

  modeling edge capacities

Automating Analysis of AODVv2 --- challenges 2

- Encoding local symmetry conditions between (abstract) nodes/edges (in different protocol instances)
- Checking local symmetry of (abstract) protocol instances

  nodes with differing numbers of neighbors

  edges with differing capacities

Automating Analysis of AODVv2 --- challenges 3

- Establishing a suitable, finite, and sufficient set of nodes/edges in a representative network

  nodes in connected/disconnected sectors

  (re)started protocol instances --- AODVv2 nodes assumed to have some persistent memory
- Calculate the strongest non-dynamic compositional invariant on the representative network
- Prove the React assumption within the automated framework
- Note: seem to be many other properties of AODVv2 of interest

Applying Compositional Symmetry Analysis

- Identify other suitable examples of dynamic/parametrized protocols
- Establish that components are `loosely-coupled'
- Automate the discovery of local symmetry relation

fully automated (local) symmetry discovery --- seems challenging

ideas: guide symmetry discovery by intuition of protocol description

many network topologies described by patterns --- use network patterns to identify suitable global/local symmetries for systems of isomorphic processes

examples of protocols whose network descriptions are `obviously symmetric' --- hardware models, communication and routing protocols, etc.

Related work on proofs of inductive invariants using symmetry and compositional reasoning

- Namjoshi, Trefler VMCAI 2012 --- compositional analysis and local symmetry
- Namjoshi, Trefler VMCAI 2013 --- abstraction and local symmetry
- Namjoshi, Trefler TACAS 2015 --- compositional analysis of dynamic protocols
- Namjoshi, Trefler FORTE 2015 --- proof analysis of AODVv2