1-1-2013

# A Reduction-Based Approach Towards Scaling Up Formal Analysis of Internet Configurations

Anduo Wang
*University of Pennsylvania*, anduo@seas.upenn.edu

Alexander JT Gurney
*University of Pennsylvania*, agurney@seas.upenn.edu

Xianglong Han
*University of Pennsylvania*, hanxiang@seas.upenn.edu

Jinyan Cao
*University of Pennsylvania*, jinyan@seas.upenn.edu

Boon T. Loo
*University of Pennsylvania*, boonloo@cis.upenn.edu

*See next page for additional authors*



University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-13-07.

# A Reduction-Based Approach Towards Scaling Up Formal Analysis of Internet Configurations


## Abstract

The Border Gateway Protocol (BGP) is the single inter-domain routing protocol that enables network operators within each autonomous system (AS) to influence routing decisions by independently setting local policies on route filtering and selection. This independence leads to fragile networking and makes analysis of policy configurations very complex. To aid the systematic and efficient study of the policy configuration space, this paper presents network reduction, a scalability technique for policy-based routing systems. In network reduction, we provide two types of reduction rules that transform policy configurations by merging duplicate and complementary router configurations to simplify analysis. We show that the reductions are sound, dual of each other and are locally complete. The reductions are also computationally attractive, requiring only local configuration information and modification. We have developed a prototype of network reduction and demonstrated that it is applicable on various BGP systems and enables significant savings in analysis time. In addition to making possible safety analysis on large networks that would otherwise not complete within reasonable time, network reduction is also a useful tool for discovering possible redundancies in BGP systems.



## Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-13-07.



## Author(s)

Anduo Wang, Alexander JT Gurney, Xianglong Han, Jinyan Cao, Boon T. Loo, Carolyn Talcott, and Andre Scedrov

# A Reduction-based Approach Towards Scaling Up Formal Analysis of Internet Configurations

Anduo Wang* Alexander J.T. Gurney* Xianglong Han* Jinyan Cao*
Boon Thau Loo* Carolyn Talcott‡ Andre Scedrov*
*University of Pennsylvania** *SRI International* ‡
{anduo,hanxiang,agurney,jinyan,boonloo}@seas.upenn.edu, clt@csl.sri.com, scedrov@math.upenn.edu

***Abstract*—The Border Gateway Protocol (BGP) is the single inter-domain routing protocol that enables network operators within each autonomous system (AS) to influence routing decisions by independently setting local policies on route filtering and selection. This independence leads to fragile networking and makes analysis of policy configurations very complex. To aid the systematic and efficient study of the policy configuration space, this paper presents network reduction, a scalability technique for policy-based routing systems. In network reduction, we provide two types of reduction rules that transform policy configurations by merging duplicate and complementary router configurations to simplify analysis. We show that the reductions are sound, dual of each other and are locally complete. The reductions are also computationally attractive, requiring only local configuration information and modification. We have developed a prototype of network reduction and demonstrated that it is applicable on various BGP systems and enables significant savings in analysis time. In addition to making possible safety analysis on large networks that would otherwise not complete within reasonable time, network reduction is also a useful tool for discovering possible redundancies in BGP systems.**

## I. Introduction

The Internet today runs on a complex routing protocol called the *Border Gateway Protocol* (BGP). It enables autonomous systems (ASes) worldwide to achieve global connectivity, subject to each system's local policy (which paths are allowed, and the route preference used to select best paths). The convergence behavior of the global Internet depends on how each AS configures its policy.

Prior work has shown that policy misconfigurations can lead to rapid oscillation between routing states, slowing or even preventing convergence [11]. This happens when the conflicting local policies cannot be reconciled: there is no solution to the routing problem. Other configurations support a unique stable solution, which normal protocol execution is bound to reach. We refer to such configurations as 'safe'. While abstract formal models of BGP [8], [5], [7] allow researchers to explore how local policies affect BGP stability, the membership problem for this safe subset is NP-hard, and real network configurations are very large, drastically limiting the feasibility of the safety test.

In this paper, we present a novel network reduction technique that enables networking researchers to study and analyze large BGP systems in a sound and automatic fashion. Central to network reduction is two forms of rewriting rules that transform policy configurations into smaller and simpler forms while preserving safety property. These rewriting rules are directed at known patterns in real networks, which exhibit considerable structural redundancy. Once a configuration is reduced, safety analysis can be performed directly to check for possible misconfigurations.

To evaluate the effectiveness of the reduction technique for scaling up safety analysis, we use an automated analyzer [20] based on the Maude rewriting logic engine [12]. In the Maude-based analyzer, a BGP system is encoded as a transition system driven by concurrent rewriting rules. Safety analysis is then performed by simulating execution runs on the transition system, as well as exhaustively exploring all execution runs for possible divergence. Our experimental results show that network reduction enables us to perform safety analysis efficiently, often completing the analysis on large networks that would otherwise not be possible to study within reasonable time.

Prior work [19] demonstrates that a limited form of rewriting rule based on merging identical router-level configurations can significantly improve convergence analysis time of BGP instances. This paper introduced a new form of rewriting rule, based on a new unified model (EPD). Using EPD model, we proved that the two forms of rules are dual. Moreover, we proved that they form a complete set of reduction rules that require only local rewrites; this and other properties of the reduction rules, have provided a deeper understanding of the redundancies presented in BGP systems, and established network reduction a sound and effective tool for scaling up formal analysis. Specifically, we make the following contributions:

**Formal model for reduction.** We propose an abstract model for modeling Internet topology and policies. This abstract model, which we call the *Extended Path Digraph* (EPD), extends prior models [7], [17], and provides a basis for reducing instances prior to analysis. EPD enables the unification of configuration specification and analysis within a common model, resulting in simpler reductions. We further provide a tool developed using Maude for automatically extracting EPD from existing network topologies and policies.

**Network reduction.** We present two reduction rules that transform EPD policy configurations to simplify analysis. These two reduction rules merge *duplicate* or *complementary* router-level configurations into one. We show that these reductions are sound and mutually dual, and establish a confluence result for duplicate reduction. These operations are also computationally attractive, since they require only local inspection and modification of the EPD structure. Indeed, we formally prove that no other such reductions are universally sound, meaning that our repertoire of local reduction methods is now complete.

**Case studies and evaluation.** We have developed a prototype of network reduction using Maude, and performed network reduction on a variety of network topologies, ranging from AS-level Caida and router-level Rocketfuel dataset, to topologies that include standard BGP configurations such as full mesh, route reflection and confederations [21]. Through a series of case studies, our experiment results demonstrate that network reduction enables significant savings in analysis time due the the use of reduction. Moreover, it makes possible safety analysis on large networks that would otherwise not complete within reasonable time. Our experiment results further demonstrate that duplicate and complementary reductions apply to different parts of the network, suggesting that reduction can also serve as a tool to identify possible types of policy and topology redundancies. We further apply the network reduction to a subset of Internet topologies obtained from RocketFuel [18] and CAIDA [1], and observed high reduction rates, suggesting that the Internet topology has a high degree of redundancy.

## II. Formal Model

We first present the formal model used for performing reduction and analysis. The central data structure, called the *Enhanced Path Digraph* (EPD), is a compact representation of two configuration aspects of a BGP system: the *topology* of how routers are connected, and for each router, the export, import, and route preference *policies*.

The policy configuration problem can be understood independently of the means for calculating routes—the BGP path-vector mechanism as implemented by the various router vendors. Policy conflicts exist independently from the details of how messages are exchanged and local data structures are updated. So the EPD model abstracts away the mechanism, and focuses on the policy itself.

Enhanced path digraph (EPD) is an extension of the path digraph structure [17], tuned to enable us to conveniently express and perform reduction. As we will see in Section IV, through this modification, EPD allows us to prove the correctness of reduction in a much more intuitive and concise way than reasoning directly with path digraph [19].

*Definition 1 (Path digraph):* Let $(V, E)$ be a directed graph and let $d$ be a designated 'destination' node in $V$. A *path digraph* instance on $(V, E, d)$ is given by $(P, E_v, E_p)$, where $P$ is a set of paths in $(V, E)$ terminating at $d$, and $E_v$ and $E_p$ are binary relations on $P$ fulfilling the following properties:

1) $(p, q)$ is in $E_v$ if and only if $p$ is a suffix of $q$.
2) If $p$ and $q$ have different origin nodes, then $(p, q)$ is not in $E_p$.
3) The restriction of $E_p$ to any set of paths having the same origin node is a strict linear order.

The relations $E_v$ and $E_p$ are called the *transmission* and *preference* relations respectively.

We may also write $E_p$ as '$\prec$', where $p \prec q$ means that $p$ is strictly preferred to $q$. The 'path digraph' structure, then, is the derived graph where the nodes are the elements of $P$ and the arcs are $E_v \cup E_p$. For example, Figure 1 shows a network of three nodes `0`, `1` and `2`, where `0` is the particular destination. The paths are shown alongside their origin nodes, in preference order (so `1` prefers path `120` over `10`); any path not shown is not permitted. The path digraph is shown in Figure 2, where dashed arcs correspond to prefers arcs, and solid arcs for transmission arcs.

We can define a notion of 'stable solution' corresponding to the endpoint of the route computation process.

*Definition 2:* A *stable solution* to a path digraph $(P, E_v, E_p)$ is a subset $S$ of $P$ that contains the empty path $\epsilon_d$ from $d$ to itself, and such that any other path $q$ is in $S$ if and only if

1) there is some $p$ in $S$ such that $(p, q)$ is in $E_v$, and
2) there is no $p'$ in $S$ such that $(p', q)$ is in $E_p$.

In general, there may be zero, one, or many stable solutions. If there is no stable solution then the routing protocol will certainly oscillate; if there are more than one, then it might oscillate (depending on details of the path-vector mechanism). If, however, there is exactly one stable solution, then the protocol will necessarily converge to it [17], [9]. A sufficient condition for this is that the path digraph be *acyclic*.

*Theorem 1:* If a path digraph has no cycle (that is, the transitive closure of $E_v \cup E_p$ is irreflexive) then it has a unique stable solution.

*Proof:* See [10]. ■

The property of a path digraph having a unique stable solution implies that the configuration is both *safe* and *robust* [3], [2], [4], [16]. Informally, a routing configuration is safe if any fair execution sequence for the path-vector protocol must eventually result in convergence of the routing state. It is robust if it remains safe even after removing some subset of the nodes and arcs in the graph.

The transmission relation $E_v$ forms an arborescence rooted at $\epsilon_d$. It therefore contains, implicitly, data about the connectivity of the original graph. The extended structure, which we now define, makes that information more explicit.

*Definition 3 (Extended path digraph (EPD)):* If $(P, E_v, E_p)$ is a path digraph on $(V, E, d)$ then the *extended path digraph* is $(P, E_v, E_p, s)$, where $s$ is the function from $P \setminus \{\epsilon_d\}$ to $V$ that maps each path to its origin node.

An EPD may be represented diagramatically by grouping the paths in $P$ according to $s$, as in Figure 3. Here, and in the rest of this paper, paths in $P$ are represented by square boxes, nodes in $V$ by ovals, preference arcs by dashed arrows and transmission arcs by solid arrows.

We will also use the following notation, for an EPD instance $(P, E_v, E_p, s)$ on a graph $(V, E, d)$:

- $P_u$ is the set $\{p \in P \mid s(p) = u\}$.
- Concatenation of paths is denoted by '$\circ$'.
- $\Gamma^+(u) = \{v \in V \mid (u, v) \in E\}$ and $\Gamma^-(u) = \{v \in V \mid (v, u) \in E\}$.

We call elements of $\Gamma^+(u)$ the downstream neighbors of $u$, and elements of $\Gamma^-(u)$ the upstream neighbors of $u$. We will also use the notation $\Gamma^+(u, v)$ to mean the union of $\Gamma^+(u)$ and $\Gamma^+(v)$, and similarly for $\Gamma^-(u, v)$.

In general, a cycle in an EPD must involve more than one node in $V$, since the preference arcs alone do not contain cycles. That is, this paper does not consider inconsistent

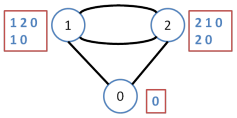


Fig. 1. The network configuration of *Disagee*

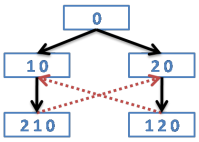


Fig. 2. The path digraph for *Disagee*

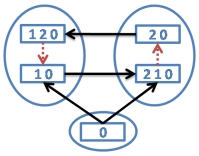


Fig. 3. The EPD notation for *Disagree*.

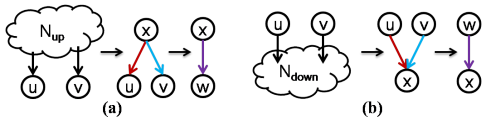


Fig. 4. Relate duplicate and complementary reduction

preference relations within a single node. Moreover, since the transmission arcs form a tree, the EPD cannot be cyclic unless there is a cycle in $(V, E)$. That is, regardless of the routing preferences, convergence is guaranteed for any network that has no cycles.

## III. Network Reduction

This section presents a rewriting calculus for policy-based routing systems, based on the idea of reducing a given network to one which is smaller, but has the same safety property. We define two specific rules, called *duplicate* and *complementary* reduction, for merging two router nodes in an EPD. Both of these are purely *local*, meaning that the operations only require examination of the relevant nodes and their immediate neighbors. In the following, assume we are working with a given EPD instance $G = (P, E_v, E_p, s)$ on a graph $(V, E, d)$.

### A. Definitions

To ease the discussion of these reductions, we first introduce three auxiliary notions: 'consistent node', 'node rewrite', and 'strongly adjacent'. We say two nodes are consistent if their configurations do not form a cycle in the EPD representation, formalized as follows:

*Definition 4 (Consistent nodes):* Given a policy configuration's EPD $(P, E_v, E_p, s)$ on the network graph $(V, E, d)$, nodes $u$ and $v$ in $V$ are *consistent* if there is no cycle in the EPD which consists only of paths $p$ for which $s(p) \in \{u, v\}$. An example of nodes violating consistency is in Figure 3: nodes 1 and 2 are not consistent since there is a cycle of paths (`120`, `10`, `210` and `20`) that only involves these two nodes. This conflicting configuration causes route oscillation behavior. Such examples of consistency violation, where two nodes have a policy conflict, should not be eliminated during reduction, in order for the problem to be detected in the final analysis. A consistency check is hence a precondition for reduction.

*Definition 5 (Strongly adjacent):* Two nodes $u$ and $v$ in $V$ are *strongly adjacent* if for every path in $P_u$, either $v$ does not appear, or it appears as the next node following $u$; and likewise for $P_v$.

Strong adjacency implies that two nodes are either immediate neighbors, or one does not route to destination through the other. It is also a precondition for reduction.

*Definition 6 (Node rewrite):* The procedure to rewrite node $u$ to $v$ is as follows: Rewrite the path $p \in P_u$ in $u$ to $p' \in P_v$ in $v$ by: If $p = u \circ w \circ r$, and $w \neq v$, then rewrite $p$ to $v \circ w \circ r$; If $p = u \circ v \circ r$, then rewrite $p$ to $v \circ r$; For all other cases, abort rewrite. Rewrite the preference among $P_u$ to that among $P_v$ by: Rewrite preference arc $(p, q)$ to $(p', q')$, where $p$ rewrites to $p'$ and $q$ to $q'$.

The global rewrite on the EPD is straightforward, once the two nodes have been merged. All paths in $P$ are rewritten according to the procedure of Definition 6, as is the transmission relation $E_v$. Write $p[u, v \mapsto w]$ for a path $p$ where all occurrences of $u$ or $v$ are replaced with $w$, eliding any '$ww$' subpath. The preferences at the new node $w$ are determined by the specific reduction procedure; for any *other* node $t$, the new path preferences on $P_t$ are obtained as follows. For a path $p$, let $\hat{p}$ denote the path that is minimal according to $E_p$ among $\{q \in P_t \mid q[u, v \mapsto w] = p\}$. Then the written $E_p$ on $t$ is $\{(\hat{p}, \hat{q}) \mid (p, q) \in E_p\}$: that is, a path 'inherits' the relative preference of the highest ranked path in its preimage.

### B. Duplicate Reduction

*Definition 7 (Duplicate condition):* Suppose that $u$ and $v$ are two consistent and strongly adjacent nodes. Then $v$ is a *duplicate* of $u$, if after rewriting $v$ to $u$, the following conditions hold: (1) $v$'s path $P'_v$ is equivalent to (or a subset of) the $u$'s paths $P_u$; (2) For every preference arc $(p, q)$ in $v$, there exists $(p', q')$ in $u$.

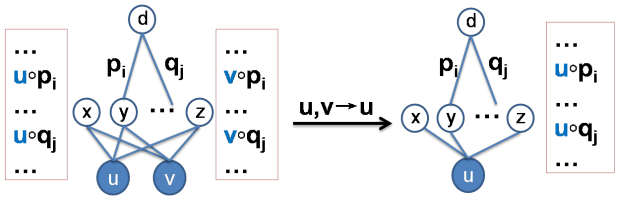


Fig. 5. Nodes $u, v$ are merged by duplicate reduction if they agree on how to route to destination $d$ through their neighbors $x, y, \ldots, z$: For any path $p_i, q_j$, $u, v$ agree on their preference.

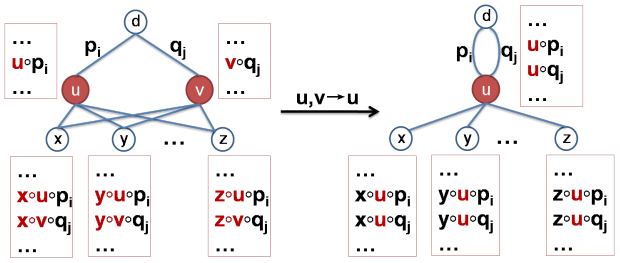


Fig. 6. Nodes $u, v$ are merged by complementary reduction if their neighbors $x, y, \ldots, z$ agree on how to route to destination $d$ through them: After merging, the route preference for any path $p_i, q_j$ are set according to the consensus among $x, y, \ldots, z$.

The duplicate precondition ensures two nodes agree on the paths and their preference to reach the destination. The duplicate 'redundancy' of $u$ and $v$ is characterized in terms

of $u, v$'s views of their neighbors through which they route to destination $d$. The general duplicate reduction process is shown in Figure 5, where $u$ and $v$ are merged into one node $u$. One can view the local change as merging the paths of $v$ into $u$, and this operation can be done consistently since they have the same routing preference.

### C. Complementary Reduction

In contrast to duplicate reduction, complementary reduction captures the redundancy observed by the neighbors of $u$ and $v$. The intuition is that if all the neighbors route to destinations through $u$ and $v$ in a consistent way, then $u$ and $v$ can be combined into one node without changing the routing behavior of these neighbors. This is formalized as follows.

*Definition 8 (Complementary Condition):* Two consistent and strongly adjacent nodes $u$ and $v$ are *complementary* if, for any paths $p$ and $q$ in $P_u \cup P_v$, and any nodes $x$ and $y$ which are immediately downstream from $u$ and $v$, the preference $(x \circ p, x \circ q)$ is in $E_p$ if and only if $(y \circ p, y \circ q)$ is in $E_p$.

The general complementary reduction process is illustrated in Figure 6 where nodes $u$ and $v$, whose neighbors $x, y, \ldots, z$ agree upon routes through them, are merged into one node. The merging is more subtle than the duplicate one: (1) The merged node $w$'s paths *combine* those from $u$ and $v$, i.e.$P_w$ is the union of $P_u$ and $P_v$; (2) The route preferences for the new node $w$ are partly determined by the consensus of preferences of their neighbors (in cases where the preference could not be derived from either $u$ or $v$). That is, if $u$ has path $p$ (and not $q$) and $v$ has path $q$ (and not $p$) then we set $p$ to be preferred over $q$ in $w$ if and only if all downstream neighbors $x$ agree that $x \circ p$ is preferred over $x \circ q$.

### D. Example

Consider the configuration in Figure 7, Here, there is an AS with three border routers (`3`, `7` and `4`) and two internal routers (`5` and `6`), as well as three external router nodes (`1`, `2`, and the destination `0`). The nodes `3` and `4` are complementary because their downstream neighbors (the internal nodes `5` and `6`) agree on the preference among paths to `0`. After merging them, the new node is again complementary to node `7`. Following a second complementary reduction step, the two internal nodes are both duplicate, and can also be eliminated.

## IV. Properties

This section establishes the three properties of network reduction: (1) The *duality* property reveals that duplicate and complementary reductions are symmetric; (2) The *soundness* property enables us to use the reduced configuration to study the original one; (3) The *local completeness* property shows that reduction can be done efficiently just by examining only "local configuration" — the two nodes and their immediate neighbors, and that duplicate and complementary reductions form a complete repertoire of such "local" methods; and (4) The *confluence* property reveals the role of merging order in reduction. In this section we present the definitions and proof sketches.

Assume in the rest of the section that we are working with a given EPD instance $G = (P, E_v, E_p, s)$ on a graph $(V, E, d)$, where $u$ and $v$ are two reducible (duplicate or complementary) nodes. Let $G'$ be an instance to which $G$ reduces by duplicate or complementary reductions.

### A. Duality: Relating Duplicate and Complementary Reduction

*Theorem 2:* **a.** If all the nodes in $\Gamma^-(u, v)$ can be merged into one node by (multiple steps of) complementary reductions, then $u$ and $v$ must be duplicate. **b.** If all the nodes in $\Gamma^+(u, v)$ can be merged into one node by (multiple steps of) duplicate reductions, then $u$ and $v$ must be complementary.

*Proof:* Part **a** may be proved as shown in Figure 4 (a). After all nodes in $\Gamma^-(u, v)$ (left-most EPD) are merged to $x$ (the middle EPD), nodes $u$ and $v$ are duplicate, since they satisfy the criteria of Definition 7. Part **b** can be proved in the same way in Figure 4 (b). ■

The duality theorem reveals the symmetry between duplicate and complementary reduction, as prefigured in Section III-D (where the border routers were complementary but the internal, downstream routers were duplicate). It also implies that if two nodes' upstream (or downstream) neighbors can be reduced into one node in our calculus, then these two nodes themselves can be further merged into one.

### B. Soundness

The main soundness result is that the reduced policy configuration has the same safety and robustness properties as the original one, and so we can use the reduced one to analyze the original.

*Theorem 3:* **a.** If $G'$ is safe then $G$ is safe; **b.** if $G'$ experiences route oscillation, then in running $G$, there exists at least one execution trace that exhibits route oscillation.

According to Theorem 1, to prove part **a**, it is sufficient to prove $G$'s EPD is acyclic. Since $G'$ is already acyclic (it is safe), it reduces to prove that the rewriting process preserves the absence of cycles in EPD representations. For part **b**, we proceed by prove its dual: $G$ is safe implies $G'$ is safe. This can be proved in the same way as **a**.

In sum, we only need to prove that the rewriting process preserves the absence of cycles in the configuration's EPD representation:

*Proposition 1:* The path digraph of $G$ is acyclic if and only if the path digraph of $G'$ is acyclic.

*Proof:* For duplicate reduction, we prove rewriting preserves cyclicity by constructing a cycle in $G'$ for any cycle $c$ in $G$. The duplicate rewrite from $G$ to $G'$ is defined by merging duplicate nodes $u$ and $v$, and the proof proceeds by case analysis of whether any of the paths originating from $u$ or $v$ are on $c$. We prove rewriting preserves acyclicity via the contrapositive: if $G'$ is cyclic then $G$ is cyclic, which is also proved by construction.

For complementary reduction, the proof is similar thanks to the EPD formalization and the dual nature of the two rules. ■

We only provide a proof sketch here, the complete proof and its graph illustration are in Appendix A.

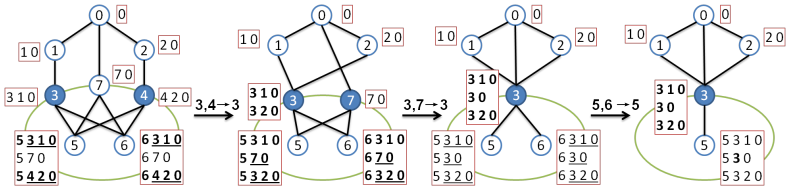


Fig. 7. Application of complementary and duplicate reduction to border and internal routers, respectively.

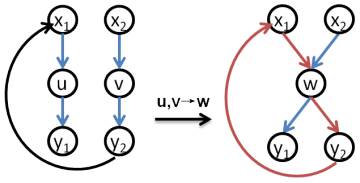


Fig. 8. If $u$ and $v$ are neither duplicate nor complementary, merging them can create a cycle.

### C. Local Completeness

We first formalize the notion of "local reduction" and "local safety", and then prove that duplicate and complementary reductions are locally complete with regard to preserving the presence or absence of EPD cycles. Intuitively, a reduction rule applied to nodes $u$ and $v$ is "local", if it only requires information from $u$, $v$ and their immediate neighbors ($\Gamma^-(u,v)$ and $\Gamma^+(u,v)$) in order to test the reduction precondition, and generate the configuration of the merged node.

Let $N_{\text{rest}}$ stand for the nodes in $V$ which are not within one hop of $u$ or $v$. We introduce a binary relation $\sim_{u,v}$ on EPDs, capturing the idea that they only differ on the configuration of $N_{\text{rest}}$, by $G \sim_{u,v} G'$ if and only if the following hold:

1. $G$ and $G'$ are on graphs having the same set of nodes.
2. They have the same path configuration for $u$ and $v$: so $P_u = P'_u$, $P_v = P'_v$, with the same preference arcs; and they have the same set of transmission arcs to and from $u$ or $v$.
3. A preference arc $(y \circ p, y \circ q)$ is in $E_p$ if and only if it is in $E'_p$, for any $y$ in $\Gamma^+(u,v)$, and any $p$ and $q$ in $P_u \cup P_v$.

*Definition 9 (Local Safety):* A network reduction rule on $G$ by merging $u$ and $v$ is locally safe, if it also preserves safety for any $G'$ with $G' \sim_{u,v} G$.

*Theorem 4 (Local Completeness):* If a network reduction rule that rewrites $G$ by merging $u$ and $v$ is locally safe, then it must be either duplicate or complementary reduction.

*Proof:* Proof by contradiction. We use proof by contradiction to establish that if $u$ and $v$ are neither duplicate nor complementary, then the reduction rule that merges them is not locally safe. To show such reduction is not locally safe, we only need to construct an acyclic EPD $G$ including $u, v$ , where application of the node merge results in $G'$ being cyclic. By assumption, $G$ is acyclic, so in particular $u$ and $v$ are not on a cycle (see the left of Figure 8). We construct an EPD where there is a series of transmission arcs from a downstream neighbor of $v$ to an upstream neighbor of $u$ (illustrated from $y_2$ to $x_1$). Merging $u$ and $v$ creates a cycle, shown in the right of Figure 8. ■

Note that, while duplicate and complementary reduction are locally complete, we do not exclude the existence of other safety preserving reduction that requires checking policy configuration beyond $u, v$ and their neighbors. That is, we do not exclude less efficient algorithms for simplifying networks.

### D. Confluence

This section discusses confluence properties of the reductions: we first prove duplicate reduction is confluent.

*Theorem 5:* [Duplicate reduction is confluent] If, for a set of nodes $V$, any pair of nodes $u$ and $v$ in $V$ are duplicate, then $V$ can be merged into one single node by multiple steps of duplicate reduction, regardless of the reduction order.

*Proof:* By induction on the size of $V$.

**The base case.** $|V| = 2$ is trivial.

**The induction step.** For $|V| = k+1 > 2$. Consider two nodes $u$ and $v$ in $V$, which by assumption are duplicate. By merging them into a new node $z$, we can rewrite $V$ to $V' = W \cup \{z\}$ where $W = V \setminus \{u, v\}$. By the induction hypothesis that any $k$ pair-wise duplicate nodes can be merged into one node, it is sufficient to prove that $V$ reduces to one node by showing that $V'$ is pair-wise duplicate, since $|V'| = k$. By definition, in $V'$, the subset $W$ is pair-wise duplicate, so we only need to show that $z$ is duplicate with any $w$ in $W$. Since $u$ and $v$ are duplicate with $w$, it must be the case that $z$ and $w$ satisfy at least the duplicate conditions. Since $P_z = P_u \cup P_v$, and by the pair-wise duplicate definition we know that paths in $P_u$ and $P_w$, in $P_v$ and $P_w$, and in $P_u$ and $P_v$ always form a unique total ordering. That is, for any three paths $p \in P_u$, $q \in P_v$, and $r \in P_w$, we know how to set the preferences between any two of them. Then there must be be a unique ordering between the three of them, and so all paths from $P_u \cup P_v \cup P_w$ are totally ordered. ■

On the other hand, complementary reductions are not confluent. Consider the EPD in Figure 9(a), in which nodes `u`, `v` and `w` have the same set of downstream neighbors. For example, node `u` has two paths $p_2$ and $p_3$, and there is some downstream preference $p_2 \prec p_3$. All downstream neighbors have consensus on preference among paths from `u` and `v` ($p_2 \prec p_1 \prec p_3$), and among paths from `v` and `w` ($p_2 \prec p_1 \prec p_4$). However, there is no consistent ranking for paths from `u` and `w`, since some nodes prefer $p_3$ over $p_4$, and others prefer the reverse. While complementary reduction can be applied to either `u` and `v` (as in Figure 9(b)), or `u` and `w` (as in Figure 9(c)), a further reduction step is not possible.

Finally, we show that duplicate reduction does not commute with complementary reduction, by exhibiting a counterexample. Consider the EPD in Figure 10, where nodes $u$ and $v$ are duplicate, and $v$ and $w$ are complementary. If $u$ and $v$ are merged into $z$ through duplicate reduction, then this $z$ is not reducible with $w$, due to the lack of consensus on paths $p_3$ and $p_4$ among downstream neighbors.

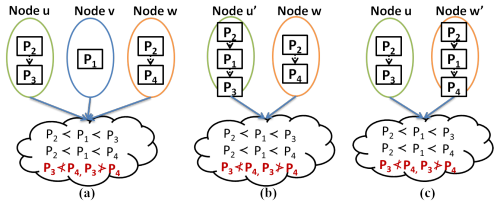


Fig. 9. The EPD in (a) either rewrites to (b) or (c) depending on the order of two complementary reductions $(u, v$ or $v, w)$

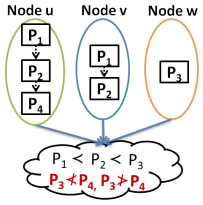


Fig. 10. Duplicate/complementary reductions do not commute

## V. Evaluation

We have implemented a prototype of network reduction using Maude. With the prototype, we demonstrate that network reduction is applicable on various networks, can be done efficiently at low overhead, and enables analysis of BGP configurations that cannot otherwise be completed. Moreover, by comparing BGP systems before and after reduction, we not only validate our reduction theory, but also gain insights into redundancy and conflicts in network configurations. We primarily selected Maude due to its existing libraries [20], [19] for modeling BGP systems and performing safety analysis [20].

### A. Network Generation

We present evaluation on a variety of networks ranging from synthetic networks including configurations of Cisco guidelines [21], and random network topologies generated using GT-ITM, to actual network topologies including CAIDA inter-AS level topologies [1], and Rocketfuel router-level ISP topologies [18]. All experiments are carried out on an Intel Xeon 2.33GHz CPU with 4GB memory, running Linux 2.6.

*a)* **Reduction on Synthetic Networks:** We evaluate network configurations that span multiple ASes, consisting of both iBGP and eBGP configurations. We first develop a model of a BGP system [15] in Maude, which consists of several ASes and routers running the path-vector protocol, and exchanging routes based on their import, export, and route selection policies. In particular, both the *Cisco-Synthetic* and *GT-ITM* network policies are realized by the *local preference* and *AS path* attributes for route selection, and import/export filtering for route exchange. In addition, we develop Maude functions that generates the EPD model from a BGP system in terms of topology and configuration attributes [15]. More details on synthetic network setup are in Appendix B.

*b)* **Reduction on Actual Topologies:** We evaluate the effectiveness of our reduction techniques on actual Internet topologies, obtained from the CAIDA Inter-AS level topologies [1] and the Rocketfuel router-level ISP topologies [18]. In the CAIDA and Rocketfuel dataset, we sample[1] the dataset to derive network of sizes up to 185 and 128 respectively. For all the topology samples, we insert the same policy configurations as our earlier Cisco-Synthetic and GT-ITM setups. We observe that the reduction rate was high, achieving a rate of 75% and 69% on average respectively. This suggests that in practice, there is significant configuration redundancy in actual configurations, observable even for a sample of the network.

[1]Our experimental dataset was limited by the physical memory constraints of storing the entire EPD in memory. As future work, we plan to explore out-of-core implementations or the use of multiple machines for executing a single reduction.

### B. Reduction Performance

Table I summarizes the performance overhead of network reduction and analysis on the two classes of input topologies for various network sizes. *Cisco-Good-22* refers to a 22-node Cisco-Synthetic topology embedded with *Good Gadgets*. The columns shown refer to:

- **EPD Generation.** Time to generate a EPD model from the input BGP configuration.
- **Reduction Time.** Reduction time required to generate the reduced EPD from the corresponding input EPD. Both reduction rules are applied, duplicate followed by complementary.
- **Reduction Time (Dup).** Same as above, except that complementary reduction is not applied. The difference allows us to compare the marginal overhead of applying complementary reduction.
- **Reduction Rate.** Percentage of redundant nodes that are reduced. For example, *68%* for *Cisco-Good-22* means that the reduced EPD is only *1-68% = 32%* of the original network size.
- **Reduction Rate (Dup).** Rate of reduction achieved by only merging duplicate nodes.
- **Reduced Analysis.** Time required to run the safety analysis on the reduced EPD after reduction, using existing Maude-based safety analyzer [20].

*c)* **EPD Generation and Reduction:** The overhead of reduction includes the time required to generate the EPD representation of the policy configuration, and the overhead of doing the reduction itself. Due to space constraints, we will show performance graphs (derived from Table I) for the the Cisco-Synthetic networks, but discuss conclusions drawn from both input topology classes.

Figure 11 shows the EPD generation time (left) and reduction time (right) as the number of nodes increases. We observe that the execution times are polynomial (cubic/quadratic) with respect to network size. While the complexity bounds are not ideal for scaling up, we note that the absolute numbers are easily within the realm of practicality. For instance, on a single commodity PC, EPD and reduction using our unoptimized Maude code requires only 16 minutes and 32 seconds (or

| Input Topology | EPD Generation Time (ms) | Reduction Time (ms) | Reduction Time (ms, Dup) | Reduction Rate | Reduction Rate (Dup) | Analysis Time (ms) |
|---|---|---|---|---|---|---|
| **Cisco-Good-22** | 3 | 74 | 22 | 68% | 63% | 429043 |
| **Cisco-Good-48** | 113 | 863 | 124 | 85% | 84% | 429043 |
| **Cisco-Good-87** | 5299 | 5665 | 649 | 92% | 92% | 429043 |
| **Cisco-Good-104** | 26567 | 10341 | 1814 | 93% | 93% | 429043 |
| **Cisco-Good-140** | 983300 | 32562 | 1814 | 95% | 94% | 429043 |
| **Cisco-Bad-22** | 5 | 96 | 23 | 69% | 68% | 80224 |
| **Cisco-Bad-49** | 112 | 935 | 119 | 86% | 86% | 80224 |
| **Cisco-Bad-87** | 5204 | 6075 | 465 | 92% | 92% | 80224 |
| **Cisco-Bad-104** | 25449 | 11258 | 725 | 93% | 93% | 80224 |
| **Cisco-Bad-121** | 177421 | 19741 | 1111 | 94% | 94% | 80224 |
| **Cisco-Disagree-23** | 2 | 30 | 14 | 78% | 80% | 184 |
| **Cisco-Disagree-53** | 40 | 352 | 73 | 90% | 90% | 184 |
| **Cisco-Disagree-70** | 182 | 901 | 164 | 93% | 92% | 184 |
| **Cisco-Disagree-103** | 3951 | 3641 | 469 | 95% | 95% | 184 |
| **Cisco-Disagree-122** | 20792 | 6430 | 810 | 96% | 96% | 184 |
| **GT-ITM-12** | 1 | 6 | 2 | 82% | 81% | 1 |
| **GT-ITM-38** | 7 | 24 | 9 | 94% | 94% | 1 |
| **GT-ITM-77** | 57 | 2279 | 68 | 95% | 95% | 1 |
| **GT-ITM-80** | 71 | 5241 | 84 | 90% | 90% | 2 |
| **GT-ITM-118** | 350 | 583143 | 455 | 86% | 91% | 2 |

TABLE I
SUMMARY OF RESULTS ACROSS VARIOUS INPUT TOPOLOGIES. AVERAGES ACROSS MULTIPLE RUNS ARE PRESENTED.

18 seconds with duplicate only reduction) respectively, for a network of 140 nodes (*Cisco-Good-140*).

We observe that in Cisco-Synthetic networks, the reduction overhead is dominated by the EPD generation time. Note however that EPD generation is amortized across both reduction and analysis, since the subsequent analysis essentially uses the same EPD representation. In contrast, in GT-ITM networks, we observe that the actual reduction dominates over EPD generation, suggesting that a nosier (more randomize) configuration increases reduction overhead. Among Cisco-Synthetic networks, we observe that reduction times are increased on denser topologies with full meshes within an AS, as compared to ASes that use route reflectors internally.

*d)* **Reduction rate:** Table I shows that reduction is very effective at reducing the size of the EPDs. In some cases, as the network sizes increases, the reduction can reduce the original EPD by 95%. Figure 12 shows the reduction rates on the Cisco-Synthetic networks. For networks beyond 40 nodes, the reduction rate is above 80% and relatively stable. The effectiveness of reduction can be attributed to the highly structured natures of these topologies, where the resulting reduced EPD is often identical to the original embedded gadgets themselves. Another source of irreducibility is if the BGP decision procedure falls through to attributes we do not analyze.

The trends observed in GT-ITM are largely similar, though we note that since these topologies are randomly generated, the reduction times and rates have higher variance across experimental runs. In Cisco-Synthetic networks, the reduction rate exhibits smaller variance due to its regular structure. In general, when a network becomes more hierarchical, (from GT-ITM to Cisco, from full-mesh to reflection), reduction rate improves due to increased redundancies. Moreover, the reduction overhead is relatively smaller (compared with the growth of network size). All in all, our results imply that a well structured hierarchical network configuration is easier to analyze in terms of reduction times. They are also more likely to result in safer configurations that do not oscillate.

*e)* **Duplicate vs Complementary:** As we noted in Section III, the complementary condition is more complex. While duplicate reduction only requires two nodes to agree upon what they learned from their neighbors, complementary requires all the neighbors of the two nodes to agree upon what are learnt from them. Our experimental results summarized in Table I validate that the overall reduction time tends to be dominated by complementary reduction. In addition, the marginal benefit of performing complementary reduction on top of duplicate reduction is often small. For instance, *Cisco-Good-22* results in a 63% reduction compared to the original EPD when only duplicate reduction is used, and 68% (i.e. an additional 5%) with both forms. While complementary reduction is less effective, we note that in almost all cases, the EPD is further reduced by the reduction. Moreover, as noted in Section V-C, both forms of reduction allow us to shed light into the policy configurations themselves.

*f)* **Analysis time:** To understand the benefits of performing safety analysis on the reduced EPDs, we compare analysis results on the original and the reduced EPD, by running existing safety analyzer [20]. The analyzer [20] uses an exhaustive search strategy to explore all possible execution sequences. Oscillation is detected if the same best route is selected multiple times during protocol execution. Overall, we observe that after reduction, the analyzer is able to detect the same route oscillation pattern found in the original network. While the original pre-reduction EPDs did not terminate within minutes, all the post-reduction EPDs completed successfully, while requiring significantly less time and state exploration. The Cisco-Synthetic topologies with *Good Gadget* requires the longest analysis time, since these are safe instances, and hence require enumerating all possible states before completing the

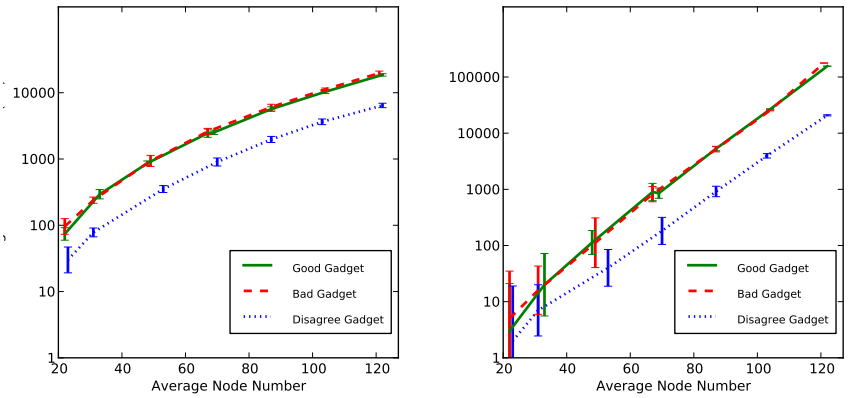


Fig. 11. EPD (left) and Reduction time (right) as number of nodes increases for the Cisco-Synthetic topologies

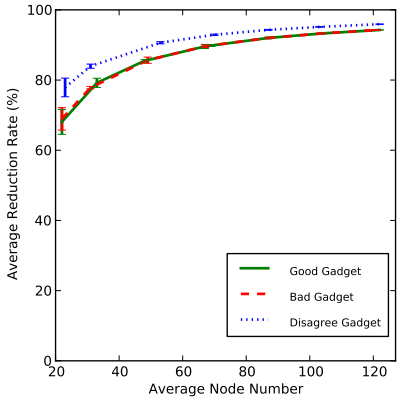


Fig. 12. Reduction rate as number of nodes increases for the Cisco-Synthetic topologies.

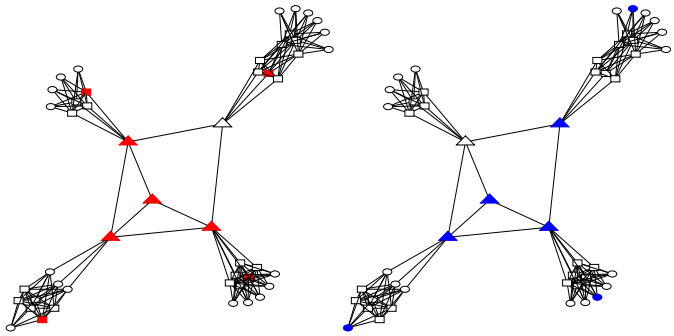

Fig. 13. In a Cisco-Synthetic network, duplicate reduction (left) merges core (triangles), internal routers (ovals) and retains the border gateway nodes (highlighted squares) post-reduction; Complementary reduction (right) merges core, border gateway routers and retains internal nodes (highlighted ovals).

analysis. In the case of unsafe instances, the analysis time was quicker and terminated whenever an unsafe execution trace was obtained.

In the GT-ITM networks, since they were less structured, not all of input topologies reduced to small gadgets (like the Cisco-Synthetic examples do) that can be analyzed quickly. For instance, in *GT-ITM-118*, only 25% of the reduced EPDs were analyzable, since the other reduced instances themselves still contain 20 nodes. We note however that when these instances are analyzable, they are typically reduced to small EPDs which can be analyzed quickly.

### C. Observations and Implications

In addition to performance benefits of feasible analysis, the process of reduction allows us to gain new insights into policy configurations.

In Section IV, we prove that duplicate and complementary are dual concepts. Figure 13 illustrates these effects, by comparing the EPDs before and after reduction, while only applying duplicate and complementary reductions respectively. In these figures, triangles denote core network (transit) ASes (which includes the embedded gadgets) in the *Cisco-Synthetic* networks, and transit AS nodes in GT-ITM. Squares refer to reflectors in stub ASes, and all other nodes are drawn as ovals. Nodes that remain after reduction are highlighted.

The duplicate reduction (left) removes some of the core network nodes, as well as some internal nodes in stub networks (those which are not border routers) In contrast, complementary reduction (right) removes an opposite family of nodes, namely the border routers that connect each stub network to the core.

This duality reveals deeper insights into the role of redundancy in networks. For core and iBGP internal nodes, duplicates arise because they are likely configured to agree upon how to route to their neighboring BGP routers for a given destination node. Such redundancies are typically eliminated by duplicate reduction. On the other hand, redundancies among border routers may be caused by configuring one router as a backup for another, so that the internal nodes that route through them view them in the same way. This falls into the definition of complementary nodes.

## VI. Related Work

*g)* **Rigorous theory of BGP safety properties:** Researchers have used abstract models including routing algebra [6], [4], [17] and combinatorial models [5] to identify sufficient safety conditions, as well as analyzing particular BGP routing systems. Practitioners have also discovered global constraints, or policy guidelines [3], which ensure safety if universally adopted. These static analysis techniques advance our understanding of how local policy configurations affect global convergence, and provide the theoretical foundation for the analysis of safety. This paper utilizes these safety conditions to prove the properties of our reduction techniques.

*h)* **Automated detection of safety violations:** Existing practical tools such as rcc [14], [2] statically check BGP configurations for possible faults due to policy conflicts across routers. This contrasts with the Maude approach of exhaustive search, based on the above rigorous theory [20]. These automated tools all suffer problems: rcc scales well but is neither sound nor complete, whereas though the analysis in Maude is sound, it suffers a state explosion problem due to the NP-hard

nature of detecting safety problems. These difficulties motivate reduction as a means of providing rigorous techniques to accelerate existing analysis.

*i)* **Accelerated automatic analysis with duplicate reduction:** In [19], a restricted form of duplicate reduction was proposed, it dramatically reduces both the state space and the execution time required for detecting safety misconfigurations, enabling users to analyze configurations up to 100 nodes (compared with the previous limit of 25 [20]).

Encouraged by the success of duplicate reduction, this paper introduced its dual, complementary reduction, based on a new unified model (EPD). Using EPD model, we proved that no other rules are locally complete; this and other properties of the reduction rules, have provided a deeper understanding of the redundancies presented in BGP systems. We also implemented a prototype with which we experimentally validated the reduction rules on various realistic topologies. In sum, the rigorously proved reduction properties together with the experiments establish our reduction rules as a sound, efficient, and complete method for scaling up existing analysis techniques.

## VII. Conclusion

We present network reduction, a scalability technique for efficiently analyzing BGP configurations in a sound and automated fashion. Based on a unified EPD model of policy configurations for both specification and analysis, we develop two reduction rules that transform a policy configuration to a smaller one by only local inspection. We proved that the two reduction rules (duplicate and complementary) are dual to each other, and are sound and locally complete with respect to the safety property. Our evaluation results not only show the benefits of reduction by scaling up safety analysis of BGP systems, but also allow us to gain insights into structural redundancies among Internet routing policy configurations.

## References


[1] The ipv4 routed /24 topology dataset. http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml.

[2] N. Feamster, R. Johari, and H. Balakrishnan. Implications of autonomy for the expressiveness of policy routing. In *ACM SIGCOMM*, 2005.

[3] L. Gao and J. Rexford. Stable Internet routing without global coordination. In *ACM SIGMETRICS*, 2000.

[4] T. G. Griffin. The stratified shortest-paths problem. In *COMSNETS*, 2010.

[5] T. G. Griffin, F. B. Shepherd, and G. Wilfong. The stable paths problem and interdomain routing. *IEEE Trans. on Networking*, 10:232–243, 2002.

[6] T. G. Griffin and J. L. Sobrinho. Metarouting. In *ACM SIGCOMM*, 2005.

[7] T. G. Griffin and G. Wilfong. An analysis of BGP convergence properties. In *SIGCOMM*, 1999.

[8] T. G. Griffin and G. Wilfong. A Safe Path Vector Protocol. In *INFOCOM*, 2000.

[9] A. J. T. Gurney. *Construction and verification of routing algebras*. PhD thesis, 2009.

[10] A. J. T. Gurney, L. Jia, A. Wang, and B. T. Loo. Partial specification of routing configurations. In *Workshop on Rigorous Protocol Engineering (WRiPE)*, 2011.

[11] C. Labovitz, G. Malan, and F. Jahanian. Internet Routing Instability. *TON*, 1998.

[12] Maude. http://maude.cs.uiuc.edu/.

[13] Modeling Topology of Large Internetworks. http://www.cc.gatech.edu/projects/gtitm/.

[14] N. Feamster and H. Balakrishnan. Detecting BGP configuration faults with static analysis. In *NSDI*, 2005.

[15] Y. Rekhter., T. Li., and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271, 2006.

[16] M. Schapira, Y. Zhu, and J. Rexford. Putting BGP on the right path: A case for next-hop routing. In *ACM SIGCOMM HotNets*, Oct. 2010.

[17] J. Sobrinho. Network routing with path vector protocols: theory and applications. In *SIGCOMM*, 2003.

[18] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *ACM SIGCOMM*, 2002.

[19] A. Wang, C. Talcott, A. J. T. Gurney, B. T. Loo, and A. Scedrov. Reduction-based Formal Analysis of BGP Instances. In *18th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2012.

[20] A. Wang, C. Talcott, L. Jia, B. T. Loo, and A. Scedrov. Analyzing BGP Instances in Maude. In *(FMOODS/FORTE)*, 2011.

[21] R. Zhang and M. Bartell. *BGP Design and Implementation*. Cisco Press, 2003.


## Appendix

### A. Soundness

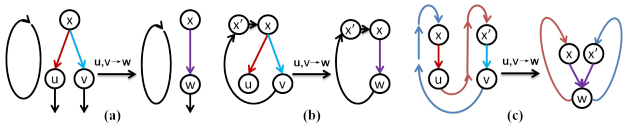


Fig. 14. Proposition 2: **Case (a.2) and (b)** None of $\Gamma^-(u,v)$ are on a cycle; **Case (c.1)** Some of $\Gamma^-(u,v)$ and $u,v$ are in the cycle, and at least one of those in $\Gamma^-(u,v)$ is in $\Gamma^-(u)$ and $\Gamma^-(v)$; **Case(c.2)** Same as **(c.1)** except that no nodes in $\Gamma^-(u,v)$ are both in $\Gamma^-(u)$ and $\Gamma^-(v)$.

Our main soundness result is that the reductions preserve the presence or absence of cycles in the EPD. From Theorem 1, this means that the reduced EPD has the same safety and robustness properties as the original. In the following, let $G$ be the original EPD, containing nodes $u$ and $v$ which are merged (by duplicate or complementary reduction) to a single node $w$ in the reduced EPD $G'$.

*1) Duplicate Reduction Preserves Cyclicity:*

*Proposition 2:* If $G$ rewrites to $G'$ by duplicate reduction, then **(1)** $G$ is cyclic implies $G'$ is cyclic, and **(2)** $G$ is acyclic implies $G'$ is acyclic.

*Proof:* For **(1)**, we construct a cycle $c'$ in $G'$ for any cycle $c$ in $G$. The duplicate rewrite from $G$ to $G'$ is defined on $u$,$v$, and $\Gamma^-(u,v)$, and the proof proceeds by case analysis of whether any of these nodes are on $c$.

**Case (a)** None of the nodes in $\Gamma^-(u,v)$ are on $c$. Consider two sub-cases: **(a.1)** $\Gamma^-(u,v) = \emptyset$ when $u$ and $v$ have no common upstream neighbor. Regardless of whether $u$ or $v$ is on $c$, a cycle $c'$ in $G'$ is constructed from $c$ by the global rewrite $c[u,v \mapsto w]$. **(a.2)** $\Gamma^+(u,v) \neq \emptyset$. We know $u$ and $v$ cannot be on $c$ either, as then one of the upstream nodes from $\Gamma^-(u,v)$ would be on $c$ too. Merging $u$ and $v$ will not affect $c$, and $c'$ is obtained by $c[u,v \mapsto w]$ (Figure 14 (a)).

**Case (b)** Some of the nodes in $\Gamma^-(u,v)$, but neither of $u$ and $v$, are on $c$. As in case **(a.1)**, $c'$ can be constructed from $c$ by $c[u,v \mapsto w]$.

**Case (c)** A subset of $\Gamma^-(u,v)$ (call it $X$), and $u$ and $v$, are on $c$. Consider two sub-cases: **(c.1)** Some of $\Gamma^-(u,v)$, $u$ and $v$ are on $c$, and at least one of those in $\Gamma^-(u,v)$ is an upstream neighbor of both $u$ and $v$. On the cycle, $x$ must be the last element in $X$, shown in Figure 14 (b). After merging $u$ and

$v$, they are replaced by $w$ in $c'$. The rest of the changes in $c'$ are obtained by $c[u, v \mapsto w]$. Note that the presence of arcs between $u$ and $v$ will not affect the result, represented by the line between $u$ and $v$ in the figure. **(c.2)** Some of $\Gamma^-(u, v)$, $u$ and $v$ are on $c$, and none of those in $\Gamma^-(u, v)$ are upstream neighbors of both $u$ and $v$. There must exist at least two nodes $x$ and $x'$ in $c$, shown in Figure 14 (c). After merging $u$ and $v$, $c$ is broken into two cycles. Pick either for $c'$.

Part **(2)** is proved via the contrapositive: if $G'$ is cyclic then $G$ is cyclic. The proof is similar to that of **(1)**: for any cycle $c$ in $G'$, we construct a cycle $c'$ in $G$. Consider the two cases:
**Case (a)** None of the nodes in $\Gamma^-(u, v)$ are on $c$. There are two sub-cases: **(a.1)** $\Gamma^-(u, v) = \emptyset$. Regardless of whether $w$ is on $c$, in the preimage, a cycle $c'$ must exist in $G$ where $w$ replaces either $u$ or $v$. **(a.2)** $\Gamma^-(u, v) \neq \emptyset$. In this case, $w$ cannot be on $c$. In the preimage $G$, where $w$ is split into $u$ and $v$, the cycle is still present. (Note that we only depict one possible splitting here.)
**Case (b)** Some $x$ in $\Gamma^-(u, v)$, but not $w$, is on $c$. The proof is similar to that of case **(a.1)**.
**Case (c)** At least one $x$ in $\Gamma^-(u, v)$, and $w$, are on $c$. In the preimage, there are two cycles, one through $u$ and the other through $v$ (but which are otherwise identical). ■

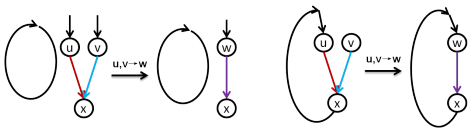


Fig. 15. Proof sketch of Proposition 3: Case 1 (left) and Case 2.1 (right).

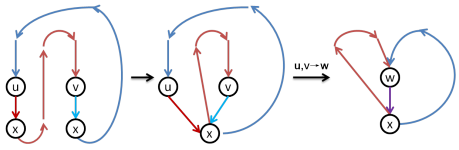


Fig. 16. Proof sketch of Proposition 3: Case 3.

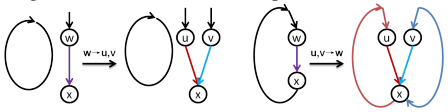


Fig. 17. Proof sketch of Proposition 4, Case 1 (left), and Case 2 (right).

*2) Complementary Reduction Preserves Cyclicity:*

*Proposition 3:* If $G$ rewrites to $G'$ by complementary reduction, then $G$ is cyclic implies $G'$ is cyclic.

*Proof:* Proof by case analysis, for any cycle $c$ in G, consider whether $u, v$ are on it.
**Case (1)** Neither $u$ nor $v$ is in the cycle. Merging $u, v$ does not affect cycle, shown on the left of Figure 15.
**Case (2)** Either $u$ or $v$ is in the cycle. Then according to complementary reduction definition, consider two sub-cases: **(2.1)** shown on the right of Figure 15, if a common downstream neighbor $x$ of $u, v$ is also on $c$, then after merging $u, v$, $c$ transforms to a new cycle $c'$ where $u$ is replaced by $w$. **(2.2)** If none of $u, v$'s common neighbor is on $c$, $c'$ can still be constructed similarly.
**Case (3)** Both $u$ and $v$ is in the cycle, as shown in Figure 16. Regardless whether some of $u, v$'s common downstream neighbor (the figure depicts the case where such a common neighbor is $x$), after merging $u, v$, cycle $c$ transforms to two cycles in $G'$. ■

*Proposition 4:* If $G$ rewrites to $G'$ by complementary reduction, then $G$ is acyclic implies $G'$ is acyclic.

*Proof:* Prove the dual statement: If $G'$ is cyclic, then $G$ is also cyclic. Proof by case analysis. For any $c$ in $G'$, assume $G'$ is obtained from $G$ by merging complementary nodes $u, v$ into $w$.
**Case (1)** If $w$ is not on $c$, as shown in the left of Figure 17. Obviously, reversing the reduction by splitting $w$ into $u, v$ does not affect the cycle.
**Case (2)** If $w$ is on the cycle of $G'$, as shown in Figure 17. Then at least one of the downstream neighbor $x$ is also on $c$. Reversing the reduction by splitting $u, v$ will split $c$ into two cycles in the original $G$. ■

### B. Reduction on Synthetic Networks

*a) Cisco-Synthetic Network:* To evaluate network reduction on well-designed, highly structural policy configurations proposed by Cisco, we construct various synthetic topologies combining full-mesh and reflection configurations according to these guidelines [21].

To understand how reduction helps in detecting route oscillation due to policy misconfigurations, we embed in the network three small substructures or *gadgets* [7]: namely the *Good*, *Bad* and *Disagree* gadgets that correspond to safe, permanent, and transient oscillation behaviors. These gadgets are embedded within the transit ASes by configuration of the local preference attributes. There are also several stub ASes, set up with full-mesh or reflection topologies (described below), and employing a policy that prefers paths with fewer AS hops. (To break tie, an older route is preferred over a newly generated one.)

*b) GT-ITM networks:* As an alternative dataset, we generate transit-stub topologies using the GT-ITM topology generator [13]. Each transit-stub topology is parameterized by the number of transit domains, nodes within a transit domain, stubs per transit nodes, and finally, nodes per stub. We increase the network size by increasing all of these parameters. We configure routing policies as follows: transit ASes are willing to carry all traffic, while each stub AS carries traffic only for itself. Given the randomness of GT-ITM topology generation, this dataset are less structured compared to the earlier Cisco-Synthetic topologies, resulting in increased variance in our results.