TRANSPARENT PARTIAL ORDER REDUCTION

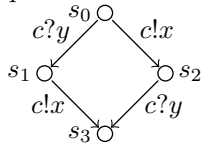
STEPHEN F. SIEGEL

Abstract. Partial Order Reduction (POR) techniques improve the basic model checking algorithm by reducing the numbers of states and transitions explored in verifying a property of the model. In the "ample set" POR framework for the verification of an LTL$_{-X}$ formula ϕ, one associates to each state s a subset T_s of the set of all transitions enabled at s. The approach requires that whenever T_s is a proper subset, the transitions in T_s must be *invisible*, i.e., their execution can never change the truth values of the atomic propositions occurring in ϕ. In this paper, we show that the invisibility restriction can be relaxed: for propositions that only occur negatively in ϕ, it suffices that the transitions in T_s merely never change the truth value from *true* to *false*, and for those that occur only positively, from *false* to *true*. This opens up opportunities for reduction, in many commonly occurring scenarios, that would not be allowed by the stricter invisibility criterion.

1. Introduction

Temporal logic model checking is a powerful tool for establishing the functional correctness of complex concurrent systems. Yet the effectiveness of model checking is often curtailed by the *state explosion problem*—the fact that the number of states of a model tends to grow exponentially in the number of concurrent processes. A variety of methods for mitigating state explosion have been proposed; among these is a family of related methods known collectively as *partial order reduction* techniques.

The basic idea behind partial order reduction is simple. An execution of a concurrent system is usually represented as an interleaved sequence of transitions from the concurrent processes. In many cases it is known *a priori* that the result of executing two transitions from distinct processes is independent of the order in which those transitions are applied. Consider, for example, a system with two processes P_1 and P_2 that access a shared channel c modeled as a FIFO queue. Suppose that only P_1 sends using c (denoted $c!x$) and only P_2 receives from c (denoted $c?y$). Then in any system state s_0 in which both a send and receive operation are enabled, the same final state s_3 will result regardless of the order in which those two operations take place:



Date: October 19, 2007.

Technical Report UDEL-CIS 2007/341.

This material is based upon work supported by the National Science Foundation under Grant No. 0541035.

This suggests that in searching the state space of this system, we only consider one of the two possible paths.

Of course, whether such a reduction is safe depends on the property being checked. Say, for example, we wish to verify that c is never empty; this can be expressed as the linear temporal logic (LTL) formula $\mathbf{AG}\ \neg\mathsf{empty}(c)$, where $\mathsf{empty}(c)$ is the proposition that holds precisely in those states in which c is empty. Suppose that s_0 is a state in which c contains one element and the send and receive operations are both enabled. In that case c will be empty in s_1 and so the property does not hold. However, if we were to choose to explore only the send transition from s_0, we would miss s_1 and visit only s_2 and s_3—states where c is non-empty. Our reduced search might therefore terminate without ever encountering a state in which c is empty, and we might erroneously conclude that the property holds.

For this reason, traditional POR methods, such as the *ample set* framework for verifying a next-time-free LTL formula $\mathbf{A}\phi$, impose an *invisibility* condition. A transition t is *invisible* if its execution in any state can never change the truth value of any atomic proposition occurring in ϕ. The invisibility condition requires that whenever the search is restricted to a proper subset of transitions departing from a state then all the transitions in that subset must be invisible. In our example, the only atomic proposition is $\mathsf{empty}(c)$. Since the send operation can change the value of this proposition from *true* to *false* and the receive operation can change it from *false* to *true*, neither operation is invisible, and so the search is required to explore both paths departing from s_0.

A well-known problem with this approach is that the effectiveness of the reduction technique drops off rapidly with the number of visible transitions. Several methods have been proposed to mitigate this problem (see Sec. 5). This paper contributes to those efforts by showing that the invisibility condition of the ample set framework can be safely replaced with a weaker *transparency* condition.

The notion of transparency refines that of invisibility by distinguishing between those atomic propositions that occur only *positively* in ϕ and those that occur only *negatively*. Roughly speaking, a proposition p occurs positively if some appearance of p in the syntax tree of ϕ occurs under an even number of negation operations. (In an expression of the form $p \to q$, p is considered to occur under one negation operation as the expression is equivalent to $(\neg p) \vee q$). Similarly, p occurs negatively if some appearance occurs under an odd number of negation operations. The transition t is *transparent* if for all p which occur positively in ϕ, t can never change the truth value of p from *false* to *true*, and for all p which occur negatively in ϕ, t can never change the value of p from *true* to *false*. Of course, some p may occur both positively and negatively in ϕ, in which case the transparency requirement for p reduces to the invisibility requirement for p.

In our example, the predicate $\mathsf{empty}(c)$ occurs only negatively in ϕ. Since the receive operation can never change $\mathsf{empty}(c)$ from *true* to *false*, the transparency condition permits a search in which only the receive transition departing from s_0 is followed. Note that if s_1 is a state in which c is empty then this reduced search would indeed catch the violation. On the other hand, the send operation may change $\mathsf{empty}(c)$ from *true* to *false*, and so a reduced search in which only the send operation departing from s_0 is followed would be rejected by the transparency condition.

A formal statement and proof of the transparency result are given below, but it will help at this point to provide the main intuition behind the proof. The correctness of the standard ample set approach comes down to showing that any path through the structure resulting from the full search can be transformed to a stutter-equivalent path through the structure resulting from the reduced search. Since stutter-equivalent sequences satisfy the same next-time-free LTL path formulas, the path through the reduced structure satisfies ϕ iff the original path satisfies ϕ. This is, however, stronger than what is required for correctness: we only need to know that any path in the full structure which violates ϕ can be transformed into a reduced path that violates ϕ. Only this weaker condition holds in the transparent setting. For example, if ϕ has the form $p\mathbf{U}q$, a path violating ϕ may be transformed by stuttering *and also* by changing any number of values taken on by p and q from *true* to *false*. It is not hard to see that such a transformed sequence must also violate $p\mathbf{U}q$.

The remainder of this paper is organized as follows. The formal framework and statement of the main result are presented in Sec. 2. The proof of the main result is then given in Sec. 3. Some applications and preliminary experiments applying the transparent technique to the verification of message-passing programs are outlined in Sec. 4. Related work is discussed in Sec. 5 and conclusions and future work are discussed in Sec. 6.

2. The Main Theorem

We adopt the notation of [2, Chap. 10]. So we let AP be a set of *atomic propositions*, and $\mathcal{M} = (S, T, S_0, L)$ a *state transition system* over AP. This means that S is a finite set of *states*, $S_0 \subseteq S$ is the set of *initial states*, T is a finite set of *transitions*, i.e., if $\alpha \in T$ then $\alpha \subseteq S \times S$, and $L : S \to 2^{AP}$ is a labeling function. We assume that there are elements $\mathsf{true}, \mathsf{false} \in AP$ with the property that $\mathsf{true} \in L(s)$ and $\mathsf{false} \notin L(s)$ for all $s \in S$. For $s \in S$ we let $\mathsf{enabled}(s) = \{\alpha \in T \mid \exists s' \in S\colon (s, s') \in \alpha\}$. For $\alpha \in T$ we let $\mathsf{enabled}(\alpha) = \{s \in S \mid \exists s' \in S\colon (s, s') \in \alpha\}$. We also assume that the transitions are *deterministic*, that is, if $s \in \mathsf{enabled}(\alpha)$ then there is a unique $s' \in S$ for which $(s, s') \in \alpha$; we denote this state s' by $\alpha(s)$.

Definition 2.1. *Let $p \in AP$, $\alpha \in T$. We say that α* is 1-transparent to p *if $p \in L(s) \Rightarrow p \in L(\alpha(s))$ for all $s \in \mathsf{enabled}(\alpha)$. We say that α* is 0-transparent to p *if $p \notin L(s) \Rightarrow p \notin L(\alpha(s))$ for all $s \in \mathsf{enabled}(\alpha)$.*

Hence if α is 1-transparent to p then it must preserve the truth of p: for any state in which p holds and α is enabled, p must also hold after executing α. Similarly, if α is 0-transparent to p then it must preserve falsehood of p. Notice that α is invisible to p if, and only if, α is both 1- and 0-transparent to p.

Having defined the notions of transparency for atomic propositions, we now extend these definitions to arbitrary LTL_{-X} formulas. Recall that an LTL_{-X} formula is a state formula of the form $\mathbf{A}\phi$, where ϕ is a path formula over AP such that the operators used in ϕ all lie in the set $\{\neg, \rightarrow, \wedge, \vee, \mathbf{G}, \mathbf{F}, \mathbf{U}, \mathbf{W}, \mathbf{R}\}$.

Let $\mathbf{A}\phi$ be an LTL_{-X} formula over AP. We will consider the syntax tree associated to ϕ. To each node u in this tree is associated a subformula ϕ_u of ϕ. If u is a leaf node then $\phi_u \in AP$.

Now let $\epsilon \in \{0, 1\}$. We will define, inductively, a number $\epsilon_u \in \{0, 1\}$ for each node u in the syntax tree of ϕ. If u is the root node, let $\epsilon_u = \epsilon$. Now assume u is

any node and we have defined ϵ_u. If u corresponds to any operation other than $\neg$ or $\rightarrow$, then we let $\epsilon_v = \epsilon_u$ for all children v of u. If u corresponds to $\neg$, then we let $\epsilon_v = 1 - \epsilon_u$ for the sole child v of u. If u corresponds to $\rightarrow$, then we let $\epsilon_v = 1 - \epsilon_u$ and $\epsilon_w = \epsilon_u$, where v and w are respectively the left and right children of u. Now we define the following:

Definition 2.2. *We say that* α is ϵ-transparent to ϕ *if for all leaf nodes* u *in the syntax tree of* ϕ, α *is* ϵ_u*-transparent to* ϕ_u. *For convenience, we write "*α is transparent to ϕ*" to mean "*α *is* 0*-transparent to* ϕ*", and simply "*α is transparent*" if* ϕ *is understood.*

Suppose α is ϵ-transparent to ϕ, and ψ is a subformula of ϕ. Thus there is some node u in the syntax tree for ϕ, such that $\psi = \phi_u$. It follows immediately from the definition that α is ϵ_u-transparent to ψ: this is because the syntax tree for ψ is the subtree of the syntax tree for ϕ with root node u.

The POR framework requires two additional structures: an *independence relation*, and a choice of *ample sets*.

Definition 2.3. *An* independence relation $I \subseteq T \times T$ *is a symmetric, antireflexive relation on* T *such that, for any* $(\alpha, \beta) \in I$, *and for all* $s \in S$ *for which* $\alpha, \beta \in \mathsf{enabled}(s)$, *the following both hold: (i)* $\alpha \in \mathsf{enabled}(\beta(s))$, *and (ii)* $\alpha(\beta(s)) = \beta(\alpha(s))$. *We say that* α *and* β *are* independent *if* $(\alpha, \beta) \in I$. *We say* α *and* β *are* dependent *if* $(\alpha, \beta) \notin I$.

A Kripke structure $M = (S, R, S_0, L)$ may be obtained from $\mathcal{M}$ by defining R so that $(s, s') \in R$ iff $(s, s') \in \alpha$ for some $\alpha \in T$. A *path in* M is an ordered pair $\pi = \langle s, \zeta \rangle$, where $s \in S$ and $\zeta = (\alpha_0, \alpha_1, \ldots)$ is a (finite or infinite) sequence of transitions, such that there exist $s_0, s_1, \ldots \in S$ satisfying $s_0 = s$ and $(s_i, s_{i+1}) \in \alpha_i$ for all i. It follows from the deterministic hypothesis that if the s_i exist, they are unique, and so we define $\mathsf{state}_i(\pi) = s_i$.

Now suppose for each $s \in S$, we are given a set $\mathsf{ample}(s) \subseteq \mathsf{enabled}(s)$. These define the *reduced Kripke structure* $M^\flat = (S, R^\flat, S_0, L)$, where $(s, s') \in R^\flat$ iff there exists $\alpha \in \mathsf{ample}(s)$ such that $\alpha(s) = s'$. A path in $M^\flat$ is a path in M for which $\alpha_i \in \mathsf{ample}(s_i)$ for all i. We repeat here the four hypotheses on ample sets from [2, Chap. 10], the only difference being we have replaced "invisible" with "transparent to ϕ" in **C2**, where $\mathbf{A}\phi$ is the LTL$_{-X}$ formula we wish to verify. We call the new condition $\mathbf{C2}_\phi$ to emphasize its dependence on ϕ (and not just on AP). The four conditions are then:

- **C0** For all $s \in S$, $\mathsf{ample}(s) = \emptyset \Leftrightarrow \mathsf{enabled}(s) = \emptyset$.
- **C1** For all $s \in S$, along every path in M that starts at s, a transition that is dependent on a transition in $\mathsf{ample}(s)$ cannot occur without a transition in $\mathsf{ample}(s)$ occurring first.
- $\mathbf{C2}_\phi$ For all $s \in S$, if $\mathsf{ample}(s) \neq \mathsf{enabled}(s)$ then every $\alpha \in \mathsf{ample}(s)$ is transparent to ϕ.
- **C3** There is no cycle in $M^\flat$ containing a state at which some transition α is enabled, but is never included in $\mathsf{ample}(s)$ for any state s in the cycle.

We can now state our main result:

Theorem 2.4. *Let* AP *be a set of atomic propositions,* $\mathcal{M} = (S, T, S_0, L)$ *a state transition system over* AP, $\mathbf{A}\phi$ *an LTL*$_{-X}$ *formula over* AP, *and* I *an independence relation for* $\mathcal{M}$. *Suppose we are given, for each* $s \in S$, *a set* $\mathsf{ample}(s) \subseteq \mathsf{enabled}(s)$,

such that **C0**, **C1**, $\mathbf{C2}_\phi$, and **C3** *all hold. Let M be the Kripke structure corresponding to $\mathcal{M}$, and $M^\flat$ the reduced Kripke structure. Then $M \models \mathbf{A}\phi \Leftrightarrow M^\flat \models \mathbf{A}\phi$.*

3. Proof of the Main Theorem

In this section, we prove Theorem 2.4. Hence we assume we are given a state transition system $\mathcal{M} = (S, T, S_0, L)$ over a set of atomic propositions AP, an LTL_{-X} formula $\mathbf{A}\phi$ over AP, an independence relation I for $\mathcal{M}$, and ample sets $\mathsf{ample}(s)$ ($s \in S$) satisfying **C0**, **C1**, $\mathbf{C2}_\phi$, and **C3**. As before, we let M be the Kripke structure corresponding to $\mathcal{M}$, and $M^\flat$ the reduced Kripke structure.

3.1. **Preliminaries.** Let $\mathbf{N} = \{0, 1, \ldots.\}$ and $\mathbf{N}^\infty = \mathbf{N} \cup \{\infty\}$. For any sequence ζ we define $|\zeta| \in \mathbf{N}^\infty$ to be the length of ζ.

Let $\pi = \langle s, \zeta\rangle$ be a path in M. Say $\zeta = (\beta_0, \beta_1, \ldots)$. We define $\mathsf{first}(\pi) = \mathsf{state}_0(\pi)$. For $i \geq 0$, we let

$$\mathsf{Suffix}_i(\pi) = \langle \mathsf{state}_i(\pi), (\beta_i, \beta_{i+1}, \ldots)\rangle.$$

The *length of* π, denoted $|\pi|$, is defined to be $|\zeta|$. If $|\pi| = n < \infty$, $\mathsf{last}(\pi)$ is defined to be $\mathsf{state}_n(\pi)$. If π is finite and σ is any path with $\mathsf{first}(\sigma) = \mathsf{last}(\pi)$, then we define $\pi * \sigma$ to be the concatenation of π with σ; it is a path starting from $\mathsf{first}(\pi)$.

We now define certain transformations on paths in M.

Definition 3.1. *Let $\pi = \langle s, (\beta_0, \beta_1, \ldots)\rangle$ be an infinite path in M, $i \in \mathbf{N}^\infty$, $\alpha \in T$, and suppose all of the following hold:*

(a) $\alpha \in \mathsf{enabled}(s)$,
(b) *if $i < \infty$ then $\alpha = \beta_i$,*
(c) *for all $j < i$, α is independent of β_j, and*
(d) *if $i > 0$ then α is transparent.*

Then we define

$$\Gamma_i^\alpha(\pi) = \begin{cases} \langle \alpha(s), (\beta_0, \ldots, \beta_{i-1}, \beta_{i+1}, \ldots)\rangle & \text{if } i < \infty \\ \langle \alpha(s), (\beta_0, \beta_1, \ldots)\rangle & \text{if } i = \infty. \end{cases}$$

Hence Γ_i^α is a function from a certain subset of the set of all infinite paths in M, to the set of infinite paths in M. The fact that $\Gamma_i^\alpha(\pi)$ is a path follows easily from the first three conditions of Definition 3.1, and Definition 2.3. Note also that Definition 3.1 depends on ϕ—or more precisely, on the set of propositions that appear positively in ϕ and the set of propositions that appear negatively in ϕ—since part (d) refers to transparency. However, since the formula ϕ under consideration is usually clear, we will not emphasize this.

Let $j \geq 0$, $\zeta = (\alpha_0, \ldots, \alpha_{j-1})$ a sequence of length j of elements of T, and $\nu = (i_0, \ldots, i_{j-1})$ a sequence of length j of non-negative integers. We will define a function Γ_ν^ζ, which again is defined on a subset of infinite paths in M. If $j = 0$ (i.e., ν and ζ are empty sequences) we let $\Gamma_\nu^\zeta(\pi) = \pi$ for all infinite paths π. For $j \geq 1$, we let

$$\Gamma_\nu^\zeta = \Gamma_{i_{j-1}}^{\alpha_{j-1}} \circ \cdots \circ \Gamma_{i_0}^{\alpha_0}, \tag{1}$$

where here $\circ$ denotes function composition. Implicit in (1) is the fact that $\Gamma_\nu^\zeta(\pi)$ is defined iff $\Gamma_{i_0}^{\alpha_0}(\pi)$ is defined and $\Gamma_{i_1}^{\alpha_1}(\Gamma_{i_0}^{\alpha_0}(\pi))$ is defined, and so on.

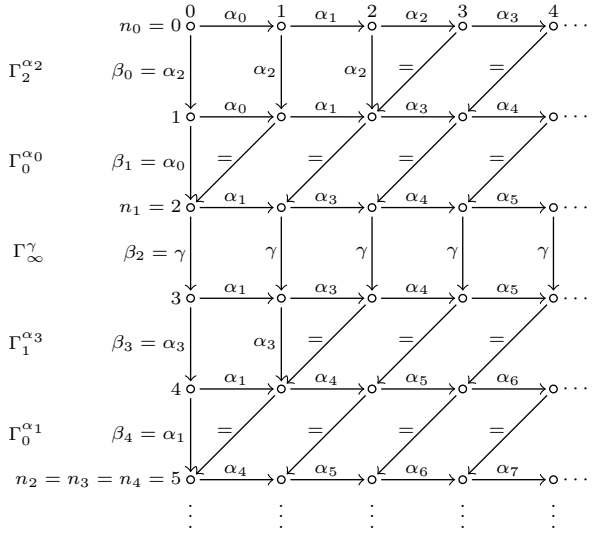


FIGURE 1. A transformation $\rho \rightsquigarrow \sigma$. The top row represents ρ; the leftmost column represents σ.

Lemma 3.2. *If $\Gamma_\nu^\zeta(\pi)$ is defined then $\sigma = \langle \mathsf{first}(\pi), \zeta \rangle$ is a finite path in M and $\mathsf{last}(\sigma) = \mathsf{first}(\Gamma_\nu^\zeta(\pi))$.*

Proof. The proof proceeds by induction on $|\zeta|$. For $|\zeta| = 0$, σ is the path of length 0 starting at $\mathsf{first}(\pi)$, and so $\mathsf{last}(\sigma) = \mathsf{first}(\pi) = \mathsf{first}(\Gamma_\nu^\zeta(\pi))$, as required.

Assume $j \geq 0$ and the Lemma holds whenever $|\zeta| \leq j$. Suppose now that $\zeta = (\alpha_0, \ldots, \alpha_j)$, $\nu = (i_0, \ldots, i_j)$, and $\Gamma_\nu^\zeta(\pi)$ is defined. Let $\zeta' = (\alpha_0, \ldots, \alpha_{j-1})$ and $\nu' = (i_0, \ldots, i_{j-1})$. Since

$$\Gamma_\nu^\zeta(\pi) = \Gamma_{i_j}^{\alpha_j}(\Gamma_{\nu'}^{\zeta'}(\pi))$$

is defined, $\pi' = \Gamma_{\nu'}^{\zeta'}(\pi)$ is defined. Hence by the inductive hypothesis, $\langle s, \zeta' \rangle$ is a path, where $s = \mathsf{first}(\pi)$, and $\mathsf{last}(\langle s, \zeta' \rangle) = \mathsf{first}(\pi')$. Now since $\Gamma_{i_j}^{\alpha_j}(\pi')$ is defined,

$$\alpha_j \in \mathsf{enabled}(\mathsf{first}(\pi')) = \mathsf{enabled}(\mathsf{last}(\langle s, \zeta' \rangle)),$$

which means that $\langle s, \zeta \rangle$ is a path. Moreover,

$$\begin{aligned} \mathsf{last}(\langle s, \zeta \rangle) = \alpha_j(\mathsf{last}(\langle s, \zeta' \rangle)) &= \alpha_j(\mathsf{first}(\pi')) \\ &= \mathsf{first}(\Gamma_{i_j}^{\alpha_j}(\pi')) = \mathsf{first}(\Gamma_\nu^\zeta(\pi)), \end{aligned}$$

completing the inductive step. □

We now define a relation on infinite paths in M.

Definition 3.3. *Let $\rho = \langle s, (\alpha_0, \alpha_1, \ldots) \rangle$ and let $\sigma = \langle s, (\beta_0, \beta_1, \ldots) \rangle$ be infinite paths in M starting at the same state s. We write $\rho \rightsquigarrow \sigma$ if there exist $i_0, i_1, \ldots \in \mathbf{N}^\infty$ such that (i) $i_j = 0$ for an infinite number of j, and (ii) for all $j \geq 0$, $\Gamma_{i_0,\ldots,i_{j-1}}^{\beta_0,\ldots,\beta_{j-1}}(\rho)$ is defined.*

By taking $i_j = 0$ for all j, we see that $\rightsquigarrow$ is reflexive, i.e., $\rho \rightsquigarrow \rho$ for any infinite path in M.

An example illustrating the concepts introduced so far is given in Figure 1. In the figure, each node represents a state. The rows are numbered starting with the

top (row 0), and working down (rows $1, 2, \ldots$). The columns are numbered similarly from left to right. Row 0 represents a path $\rho = \langle s, (\alpha_0, \alpha_1, \ldots)\rangle$, while the leftmost column represents a path $\sigma = \langle s, (\beta_0, \beta_1, \ldots)\rangle$ for which $\rho \rightsquigarrow \sigma$. The transformation from ρ to σ is illustrated one step at a time. Hence, the path in row 1 represents $\Gamma_2^{\alpha_2}(\rho)$, the path in row 2 represents $\Gamma_0^{\alpha_0}(\Gamma_2^{\alpha_2}(\rho))$, and so on. An edge denotes either a transition from one state to another, or equality, i.e., that the source and destination states are equal. From the figure, we can discern that α_2, α_3, and γ must all be transparent transitions, that α_2 is independent of α_0 and α_1, and so on. The numbers $n_0, n_1, \ldots$ will be defined in Sec. 3.2.

Lemma 3.4. *If ρ is an infinite path in M starting at a state s then there exists an infinite path σ in $M^\flat$ starting at s such that $\rho \rightsquigarrow \sigma$.*

Proof. First, we define elements $\beta_j \in T$ and $i_j \in \mathbf{N}^\infty$ for all $j \geq 0$, by induction on j. Along the way, we will show that for all j

$$\pi_j = \Gamma_{i_0,\ldots,i_{j-1}}^{\beta_0,\ldots,\beta_{j-1}}(\rho) \text{ is defined, and} \tag{2}$$

$$\langle s, (\beta_0, \ldots, \beta_{j-1})\rangle \text{ is a path in } M^\flat. \tag{3}$$

This will imply that $\sigma = \langle s, (\beta_0, \beta_1, \ldots)\rangle$ is a path in $M^\flat$ and it will only remain to show that an infinite number of the i_j are 0 to conclude that $\rho \rightsquigarrow \sigma$.

The case $j = 0$ is vacuous, so suppose $j \geq 0$ and that the β_k and i_k have been defined for $0 \leq k < j$ to satisfy (2) and (3). Write $\pi_j = \langle s_j, (\gamma_{j,0}, \gamma_{j,1}, \ldots)\rangle$. If $\gamma_{j,0} \in \mathsf{ample}(s_j)$ we may set $i_j = 0$ and $\beta_j = \gamma_{j,0}$. It is immediate that $\Gamma_{i_j}^{\beta_j}(\pi_j)$ is defined, and thus, by Lemma 3.2, $\chi = \langle s, (\beta_0, \ldots, \beta_j)\rangle$ is a path. By the inductive hypothesis, $\langle s, (\beta_0, \ldots, \beta_{j-1})\rangle$ is a path in $M^\flat$, so since $\beta_j \in \mathsf{ample}(s_j)$, χ is a path in $M^\flat$.

So assume $\gamma_{j,0} \notin \mathsf{ample}(s_j)$. Then $\gamma_{j,0} \in \mathsf{enabled}(s_j) \setminus \mathsf{ample}(s_j)$, and so by $\mathbf{C2}_\phi$, all of the transitions in $\mathsf{ample}(s_j)$ are transparent.

Now either there is some $k \geq 1$ such that $\gamma_{j,k} \in \mathsf{ample}(s_j)$, or there is no such k.

If the first is the case, choose the least such k. Then by **C1**, $\gamma_{j,k}$ is independent of $\gamma_{j,l}$ for all $l < k$, so we may take $i_j = k$ and $\beta_j = \gamma_{j,k}$. Since all transitions in $\mathsf{ample}(s_j)$ are transparent, it is again the case that $\Gamma_{i_j}^{\beta_j}(\pi_j)$ is defined, and we may reason exactly as before to see that $\langle s, (\beta_0, \ldots, \beta_j)\rangle$ is a path in $M^\flat$.

So suppose there is no such k. Then **C1** implies that for all $k \geq 0$, $\gamma_{j,k}$ is independent of every transition in $\mathsf{ample}(s_j)$. By **C0**, $\mathsf{ample}(s_j)$ is nonempty. So we let $i_j = \infty$ and β_j be any element of $\mathsf{ample}(s_j)$, and this completes the inductive step.

It remains to see that $i_j = 0$ for an infinite number of j. Suppose that this is not the case. Then there exists $j \geq 0$ such that for all $k \geq j$, $i_k > 0$. Hence $\gamma_{k,0} = \gamma_{j,0}$ for all $k \geq j$. By construction, this implies

$$\gamma_{j,0} \in \mathsf{enabled}(s_k) \setminus \mathsf{ample}(s_k) \qquad (k \geq j).$$

Now $s_k = \mathsf{state}_k(\sigma)$ for all k, and, since S is finite, there must exist $l > k \geq j$ such that $s_l = s_k$. Hence s_k lies on a cycle in $M^\flat$ in which $\gamma_{j,0}$ is enabled, but not included in the ample set for any of the states of the cycle, contradicting **C3**. □

Lemma 3.5. *Suppose π is an infinite path in M, $i \in \mathbf{N}$, $j \in \mathbf{N}^\infty$, and $\Gamma_j^\alpha(\pi)$ is defined. Then*

$$\mathsf{Suffix}_i(\Gamma_j^\alpha(\pi)) = \begin{cases} \Gamma_{j-i}^\alpha(\mathsf{Suffix}_i(\pi)) & \text{if } j \geq i \\ \mathsf{Suffix}_{i+1}(\pi) & \text{if } j \leq i. \end{cases}$$

Proof. This is an easy exercise in the definitions. Note that the two cases overlap when $i = j$. This is correct since $\Gamma_0^\alpha(\mathsf{Suffix}_i(\pi)) = \mathsf{Suffix}_{i+1}(\pi)$. □

3.2. **A Proposition.** The purpose of this section is to prove the following:

Proposition 3.6. *Suppose $\rho = \langle s, (\alpha_0, \alpha_1, \ldots)\rangle$ and $\sigma = \langle s, (\beta_0, \beta_1, \ldots)\rangle$ are infinite paths in M starting at the same state s. Suppose also $\rho \rightsquigarrow \sigma$. Then there exist finite paths $\tau_0, \tau_1, \ldots$ in M, and non-negative integers $n_0, n_1, \ldots$ such that the following hold for all $d \geq 0$:*

(a) $d \leq n_d \leq n_{d+1}$,
(b) τ_d *consists entirely of transparent transitions,*
(c) $\mathsf{first}(\tau_d) = \mathsf{state}_d(\rho)$ *and* $\mathsf{last}(\tau_d) = \mathsf{state}_{n_d}(\sigma)$,
(d) $\mathsf{Suffix}_d(\rho) \rightsquigarrow \tau_d * \mathsf{Suffix}_{n_d}(\sigma)$, *and*
(e) β_k *is transparent whenever* $n_d \leq k < n_{d+1} - 1$.

The remainder of this section will be devoted to a proof of Proposition 3.6.

By Definition 3.3, there are non-negative integers $i_0, i_1, \ldots$ such that (i) $i_k = 0$ for an infinite number of k, and (ii) for all $k \geq 0$, $\Gamma_{i_0,\ldots,i_{k-1}}^{\beta_0,\ldots,\beta_{k-1}}(\rho)$ is defined.

Fix $d \geq 0$. We will define non-negative integers $m_{d,k}$ ($k \geq 0$) by induction on k, as follows:

$$m_{d,0} = d \tag{4}$$

$$m_{d,k+1} = \begin{cases} m_{d,k} & \text{if } i_k \geq m_{d,k} \\ m_{d,k} - 1 & \text{otherwise.} \end{cases} \tag{5}$$

Hence $(m_{d,0}, m_{d,1}, \ldots)$ is a non-increasing sequence of integers starting at d. At each step, the sequence either decreases by one or remains unchanged. If it reaches 0, it remains at 0 from that point forward, since every $i_k \geq 0$. We claim that in fact the sequence must reach 0: this follows easily from the fact that there are an infinite number of k for which $i_k = 0$. We let n_d be the least k for which $m_{d,k} = 0$.

In Figure 1, we may interpret the $m_{d,k}$ as follows: consider the path that begins at the state in column d of row 0, and that progresses by following the unique edge that moves down one row at each step. Then $m_{d,k}$ is the column number of the state that results after taking k steps in this path. For example, taking $d = 2$, we see that $m_{2,0} = m_{2,1} = 2$, $m_{2,2} = m_{2,3} = m_{2,4} = 1$, and $m_{2,5} = 0$. In particular, $n_2 = 5$.

The following gathers together some facts about the $m_{d,k}$ which will be used later on:

Lemma 3.7. *The following hold for all $d, k \geq 0$:*

(a) $m_{d,k} - 1 \leq m_{d,k+1} \leq m_{d,k}$,
(b) $m_{d,n_d} = 0$,
(c) $m_{d,n_d-1} > 0$,
(d) $m_{d,k+1} = m_{d,k} \Leftrightarrow i_k \geq m_{d,k}$,
(e) $m_{d,k} \geq d - k$,

(f) $|\{l \mid 0 \leq l < n_d, i_l \geq m_{d,l}\}| = n_d - d$,
(g) $m_{d,k} = m_{d+1,k} \Rightarrow m_{d,k+1} = m_{d+1,k+1}$, *and*
(h) $m_{d,k} \leq m_{d+1,k} \leq m_{d,k} + 1$.

Proof. Statements (a)–(d) are immediate from the definitions. To prove (e), fix d, and use induction on k: for $k = 0$, the statement holds since $m_{d,0} = d$, and the inductive step follows from (a).

To see (f), let $A = \{0, 1, \ldots, n_d - 1\}$ and $B = \{l \in A \mid i_l < m_{d,l}\}$. Let

$$\bar{m}_{d,l} = m_{d,l} - m_{d,l+1} \qquad (l \in A).$$

By (a), $m_{d,l} \in \{0, 1\}$ for all $l \in A$. Moreover, $\bar{m}_{d,l} = 1 \Leftrightarrow l \in B$, by (d). By (b), $m_{d,n_d} = 0$, and hence

$$d = m_{d,0} = \sum_{l \in A} \bar{m}_{d,l} = |\{l \in A \mid \bar{m}_{d,l} = 1\}| = |B|.$$

So $|A \setminus B| = |A| - |B| = n_d - d$, proving (f).

Statement (g) holds since, by (5), the value of $m_{d,k+1}$ depends only on $m_{d,k}$ and i_k.

We now turn to the proof of (h). We first claim that for all $d, k \geq 0$,

$$m_{d,k} \leq m_{d+1,k}. \tag{6}$$

To show this, we fix d, and use induction on k. For $k = 0$, (6) reduces to the statement $d \leq d + 1$, which clearly holds. Suppose now that (6) holds and we wish to show it still holds when k is replaced by $k + 1$. There are two cases to consider: either (i) $m_{d,k} < m_{d+1,k}$, or (ii) $m_{d,k} = m_{d+1,k}$. In the first case, we have

$$m_{d,k+1} \leq m_{d,k} \leq m_{d+1,k} - 1 \leq m_{d+1,k+1},$$

as required. In the second case, we have $m_{d,k+1} = m_{d+1,k+1}$, by (g), which completes the inductive step.

We will now show that for $d, k \geq 0$,

$$m_{d+1,k} - m_{d,k} \leq 1, \tag{7}$$

which, in light of (6), will complete the proof of (h). Again, we fix $d \geq 0$ and use induction on k. For $k = 0$, (7) reduces to the statement $d + 1 - d \leq 1$, which clearly holds. Suppose now that (7) holds, and we wish to show that it holds with $k + 1$ in place of k. Then $m_{d+1,k} - m_{d,k}$ must equal either 0 or 1. In the first case, we have $m_{d+1,k} = m_{d,k}$, and so by (g), $m_{d+1,k+1} - m_{d,k+1} = 0$, and the inductive step holds.

So assume that $m_{d+1,k} - m_{d,k} = 1$. There are again two cases to consider: either (i) $m_{d,k+1} = m_{d,k}$ or (ii) $m_{d,k+1} = m_{d,k} - 1$. If (i) is the case, we have

$$m_{d+1,k+1} - m_{d,k+1} \leq m_{d+1,k} - m_{d,k} = 1,$$

and the inductive step holds. If (ii) is the case, we must have $i_k < m_{d,k}$, by (d). Hence by (6), $i_k < m_{d,k} \leq m_{d+1,k}$, and again by (d), $m_{d+1,k+1} = m_{d+1,k} - 1$. Hence

$$m_{d+1,k+1} - m_{d,k} = m_{d+1,k} - 1 - m_{d,k} + 1 = m_{d+1,k} - m_{d,k} \leq 1,$$

completing the inductive step and the proof of Lem. 3.7. □

We now return to the proof of Proposition 3.6. According to Lemma 3.7(a), if $m_{d+1,k} = 0$ then $m_{d,k} \leq m_{d+1,k} = 0$. Hence $n_d \leq n_{d+1}$. Moreover, Lemma 3.7(e) implies $m_{d,d-1} \geq 1$, which shows that $n_d \geq d$, establishing Proposition 3.6(a).

Suppose $n_d \leq k < n_{d+1} - 1$. Then $m_{d,k} = 0$ but $m_{d+1,k} \neq 0$, which implies $m_{d+1,k} = 1$ by Lemma 3.7(h). We also have $m_{d+1,k+1} = 1$, since $k + 1 < n_{d+1}$. Hence $m_{d+1,k+1} = m_{d+1,k}$, which, by Lemma 3.7(d), implies $i_k \geq m_{d+1,k} = 1$. By Definition 3.1, this means that β_k is transparent, proving Proposition 3.6(e).

Fix $d \geq 0$. For each $k \geq 0$, we will define a sequence $\xi_{d,k}$ of transitions and a sequence $\mu_{d,k}$ of non-negative integers. For $k = 0$, these are both the empty sequence. Assuming they have been defined for k, we let

$$\xi_{d,k+1} = \begin{cases} \xi_{d,k} * (\beta_k) & \text{if } i_k \geq m_{d,k} \\ \xi_{d,k} & \text{otherwise} \end{cases} \tag{8}$$

$$\mu_{d,k+1} = \begin{cases} \mu_{d,k} * (i_k - m_{d,k}) & \text{if } i_k \geq m_{d,k} \\ \mu_{d,k} & \text{otherwise.} \end{cases} \tag{9}$$

By Lemma 3.7(f), we have

$$|\xi_{d,n_d}| = |\mu_{d,n_d}| = n_d - d. \tag{10}$$

Now if $k < n_d$, then $m_{d,k} > 0$, so if $i_k \geq m_{d,k}$ then $i_k > 0$ and therefore β_k is transparent. Hence ξ_{d,n_d} consists solely of transparent transitions.

On the other hand, if $k \geq n_d$ then $m_{d,k} = 0$ and so $i_k \geq m_{d,k}$. It follows that

$$\xi_{d,k} = \xi_{d,n_d} * (\beta_{n_d}, \beta_{n_d+1}, \ldots, \beta_{k-1}) \qquad (k \geq n_d) \tag{11}$$

$$\mu_{d,k} = \mu_{d,n_d} * (i_{n_d}, i_{n_d+1}, \ldots, i_{k-1}) \qquad (k \geq n_d) \tag{12}$$

For any $k \geq 0$, let $\zeta_k = (\beta_0, \ldots, \beta_{k-1})$ and let $\nu_k = (i_0, \ldots, i_{k-1})$. Recall that, by assumption, $\Gamma^{\zeta_k}_{\nu_k}(\rho)$ is defined for all k.

Lemma 3.8. *For all $k \geq 0$,*

$$\Gamma^{\xi_{d,k}}_{\mu_{d,k}}(\mathsf{Suffix}_d(\rho)) = \mathsf{Suffix}_{m_{d,k}}(\Gamma^{\zeta_k}_{\nu_k}(\rho)). \tag{13}$$

Proof. Implicit in (13) is the claim that the left-hand side is defined. We prove (13) by induction on k. For $k = 0$, (13) reduces to the equation $\mathsf{Suffix}_d(\rho) = \mathsf{Suffix}_d(\rho)$. Now suppose that (13) holds and we wish to show that it holds when k is replaced by $k + 1$. There are two cases to consider: (i) $i_k \geq m_{d,k}$ and (ii) $i_k < m_{d,k}$.

Suppose $i_k \geq m_{d,k}$, so $m_{d,k+1} = m_{d,k}$. Since

$$\Gamma^{\beta_k}_{i_k}(\Gamma^{\zeta_k}_{\nu_k}(\rho)) = \Gamma^{\zeta_{k+1}}_{\nu_{k+1}}(\rho) \tag{14}$$

is defined, Lemma 3.5 implies

$$\mathsf{Suffix}_{m_{d,k}}(\Gamma^{\beta_k}_{i_k}(\Gamma^{\zeta_k}_{\nu_k}(\rho))) = \Gamma^{\beta_k}_{i_k - m_{d,k}}(\mathsf{Suffix}_{m_{d,k}}(\Gamma^{\zeta_k}_{\nu_k}(\rho))). \tag{15}$$

In particular, the right hand side of (15) is defined. Hence

$$\Gamma^{\xi_{d,k+1}}_{\mu_{d,k+1}}(\mathsf{Suffix}_d(\rho)) = \Gamma^{\beta_k}_{i_k - m_{d,k}}(\Gamma^{\xi_{d,k}}_{\mu_{d,k}}(\mathsf{Suffix}_d(\rho))) \tag{16}$$

$$= \Gamma^{\beta_k}_{i_k - m_{d,k}}(\mathsf{Suffix}_{m_{d,k}}(\Gamma^{\zeta_k}_{\nu_k}(\rho))) \tag{17}$$

is defined. Moreover, combining (14)–(17), we have

$$\begin{aligned}\Gamma^{\xi_{d,k+1}}_{\mu_{d,k+1}}(\mathsf{Suffix}_d(\rho)) &= \mathsf{Suffix}_{m_{d,k}}(\Gamma^{\beta_k}_{i_k}(\Gamma^{\zeta_k}_{\nu_k}(\rho)))\\ &= \mathsf{Suffix}_{m_{d,k}}(\Gamma^{\zeta_{k+1}}_{\nu_{k+1}}(\rho))\\ &= \mathsf{Suffix}_{m_{d,k+1}}(\Gamma^{\zeta_{k+1}}_{\nu_{k+1}}(\rho)),\end{aligned}$$

completing the inductive step for this case.

Suppose instead that $i_k < m_{d,k}$. Then $i_k \le m_{d,k+1} = m_{d,k} - 1$. Hence $\xi_{d,k+1} = \xi_{d,k}$ and $\mu_{d,k+1} = \mu_{d,k}$. Moreover, Lemma 3.5 implies

$$\mathsf{Suffix}_{m_{d,k}-1}(\Gamma^{\beta_k}_{i_k}(\Gamma^{\zeta_k}_{\nu_k}(\rho))) = \mathsf{Suffix}_{m_{d,k}}(\Gamma^{\zeta_k}_{\nu_k}(\rho)). \tag{18}$$

Whence

$$\begin{aligned}\Gamma^{\xi_{d,k+1}}_{\mu_{d,k+1}}(\mathsf{Suffix}_d(\rho)) &= \Gamma^{\xi_{d,k}}_{\mu_{d,k}}(\mathsf{Suffix}_d(\rho))\\ &= \mathsf{Suffix}_{m_{d,k}}(\Gamma^{\zeta_k}_{\nu_k}(\rho))\\ &= \mathsf{Suffix}_{m_{d,k}-1}(\Gamma^{\beta_k}_{i_k}(\Gamma^{\zeta_k}_{\nu_k}(\rho)))\\ &= \mathsf{Suffix}_{m_{d,k+1}}(\Gamma^{\zeta_{k+1}}_{\nu_{k+1}}(\rho)),\end{aligned}$$

completing the inductive step for this case as well. □

We let

$$\tau_d = \langle \mathsf{state}_d(\rho), \xi_{d,n_d}\rangle \qquad (d \ge 0). \tag{19}$$

From Lemmas 3.2 and 3.8, we conclude that τ_d is a path, and, since $m_{d,n_d} = 0$,

$$\begin{aligned}\mathsf{last}(\tau_d) &= \mathsf{first}(\Gamma^{\xi_{d,n_d}}_{\mu_{d,n_d}}(\mathsf{Suffix}_d(\rho)))\\ &= \mathsf{first}(\mathsf{Suffix}_0(\Gamma^{\zeta_{n_d}}_{\nu_{n_d}}(\rho)))\\ &= \mathsf{first}(\Gamma^{\zeta_{n_d}}_{\nu_{n_d}}(\rho))\\ &= \mathsf{last}(\langle s, (\beta_0, \ldots, \beta_{n_d-1})\rangle)\\ &= \mathsf{state}_{n_d}(\sigma).\end{aligned}$$

This proves Proposition 3.6(c), and also shows that $\tau_d * \mathsf{Suffix}_{n_d}(\sigma)$ is a path. We have already seen τ_d consists solely of transparent transpositions, proving Prop. 3.6(b).

Write $\tau_d * \mathsf{Suffix}_{n_d}(\sigma) = \langle \mathsf{state}_d(\rho), (\gamma_0, \gamma_1, \ldots)\rangle$. By (11),

$$(\gamma_0, \ldots, \gamma_{k-1}) = \xi_{d,k+d} \qquad (k \ge n_d - d). \tag{20}$$

Write $\mu_{d,n_d} = (i'_0, i'_1, \ldots, i'_{n_d-d-1})$, and define $i'_k = i_{k+d}$ $(k \ge n_d - d)$. It follows from (12) that

$$(i'_0, \ldots, i'_{k-1}) = \mu_{d,k+d} \qquad (k \ge n_d - d). \tag{21}$$

Since $i_k = 0$ for an infinite number of k, $i'_k = 0$ for an infinite number of k. Moreover, for all $k \ge 0$,

$$\Gamma^{\gamma_0,\ldots,\gamma_{k-1}}_{i'_0,\ldots,i'_{k-1}}(\mathsf{Suffix}_d(\rho)) = \Gamma^{\xi_{d,k+d}}_{\mu_{d,k+d}}(\mathsf{Suffix}_d(\rho))$$

is defined, by Lem. 3.8. Hence $\mathsf{Suffix}_d(\rho) \rightsquigarrow \tau_d * \mathsf{Suffix}_{n_d}(\sigma)$, proving Proposition 3.6(d). This completes the proof of Proposition 3.6.

3.3. **Formula Preservation.** In this section, we prove the following:

Proposition 3.9. *If $\rho \rightsquigarrow \sigma$ and $\rho \not\models \phi$ then $\sigma \not\models \phi$.*

We will first show that it suffices to prove Proposition 3.9 for path formulas in which the only operators are $\mathbf{U}$, $\neg$, and $\wedge$. For suppose we have done this, and that we are given a formula ϕ which involves those three operators and possibly $\vee$ as well. We then transform ϕ into a formula ψ involving only the first three operations by replacing every occurrence of $\theta \vee \chi$ with $\neg((\neg\theta) \wedge (\neg\chi))$. Now a transition α is ϵ-transparent to $\theta \vee \chi$ iff α is ϵ-transparent to $\neg((\neg\theta) \wedge (\neg\chi))$: this is because in the syntax tree of the latter, the θ and χ lie under an even number (2) of negations. It follows that α is ϵ-transparent to ϕ iff α is ϵ-transparent to ψ. Hence a choice of ample sets satisfies $\mathbf{C2}_\phi$ iff that choice satisfies $\mathbf{C2}_\psi$. So if we are given a choice of ample sets satisfying all four conditions for ϕ it will also satisfy those conditions for ψ, since the other three conditions do not depend on the formula. Moreover, we have $M \models \mathbf{A}\phi \Leftrightarrow M \models \mathbf{A}\psi$, and $M^\flat \models \mathbf{A}\phi \Leftrightarrow M^\flat \models \mathbf{A}\psi$, since for any infinite path π in M, $\pi \models \phi \Leftrightarrow \pi \models \psi$. By assumption, Proposition 3.9 holds for ψ, i.e., $M \models \mathbf{A}\psi \Leftrightarrow M^\flat \models \mathbf{A}\psi$. Hence $M \models \mathbf{A}\phi \Leftrightarrow M^\flat \models \mathbf{A}\phi$, as required. This shows that Proposition 3.9 may be extended to formulas using $\vee$, and the same argument shows that it can be extended, one operation at a time, to deal with the remaining operators, by using the identities $\theta \rightarrow \chi \equiv (\neg\theta) \vee \chi$, $\mathbf{F}\theta \equiv \mathsf{true}\mathbf{U}\theta$, $\mathbf{G}\theta \equiv \neg\mathbf{F}\neg\theta$, $\theta\mathbf{R}\chi \equiv \neg((\neg\theta)\mathbf{U}(\neg\chi))$, and $\theta\mathbf{W}\chi \equiv (\mathbf{G}\theta) \vee (\theta\mathbf{U}\chi)$.

So we now assume that the only operators occurring in ϕ are $\mathbf{U}$, $\neg$, and $\wedge$. The proof will work by induction over the syntax tree for ϕ, beginning at the leaf nodes and working up the tree to the root. For each node u in the tree, we define two statements $\mathfrak{p}_u$ and $\mathfrak{q}_u$. The definitions depend on the value of ϵ_u.

If $\epsilon_u = 1$, then we let $\mathfrak{p}_u$ denote the statement

(22) for all infinite paths ρ, σ in M for which $\rho \rightsquigarrow \sigma$, $\rho \models \phi_u \Rightarrow \sigma \models \phi_u$

and $\mathfrak{q}_u$ denote the statement

(23) for all infinite paths $\pi = \langle s, (\alpha_0, \alpha_1, \ldots)\rangle$ in M for which α_0 is ϵ_u-transparent to ϕ_u, $\pi \models \phi_u \Rightarrow \mathsf{Suffix}_1(\pi) \models \phi_u$.

If $\epsilon_u = 0$ then $\mathfrak{p}_u$ and $\mathfrak{q}_u$ are obtained by replacing each occurrence of $\models$ in (22) and (23) with $\not\models$.

We will show by induction that $\mathfrak{p}_u \wedge \mathfrak{q}_u$ holds for all nodes u. This will complete the proof of Proposition 3.9, since if u is the root node then $\mathfrak{p}_u$ is just a restatement of Proposition 3.9. But first, we will need the following:

Lemma 3.10. *Suppose that u is a node in the syntax tree of ϕ, $\mathfrak{p}_u \wedge \mathfrak{q}_u$ holds, ρ and σ are infinite paths in M for which $\rho \rightsquigarrow \sigma$, $d \geq 0$, and n_d is as in Proposition 3.6. Then, if $\epsilon_u = 1$, the following both hold:*

(a) *If $\mathsf{Suffix}_d(\rho) \models \phi_u$ then $\mathsf{Suffix}_{n_d}(\sigma) \models \phi_u$.*
(b) *If $\mathsf{Suffix}_i(\rho) \models \phi_u$ for all i such that $0 \leq i < d$, then $\mathsf{Suffix}_k(\sigma) \models \phi_u$ for all k such that $0 \leq k < n_d$.*

If $\epsilon_u = 0$, then the same statements hold after replacing each occurrence of $\models$ with $\not\models$.

Proof. Let $n_0, n_1, \ldots$, and $\tau_0, \tau_1, \ldots$, be as in Proposition 3.6. We assume $\epsilon_u = 1$, the case $\epsilon_u = 0$ being entirely similar.

We first prove (a). By Proposition 3.6(d), $\mathsf{Suffix}_d(\rho) \rightsquigarrow \tau_d * \mathsf{Suffix}_{n_d}(\sigma)$. By $\mathfrak{p}_u$, this means that $\tau_d * \mathsf{Suffix}_{n_d}(\sigma) \models \phi_u$. By Proposition 3.6(b), τ_d consists entirely of transitions that are 0-transparent to ϕ, i.e., transitions that are ϵ_u-transparent to ϕ_u. By repeated application of $\mathfrak{q}_u$, we conclude that $\mathsf{Suffix}_{n_d}(\sigma) \models \phi_u$, as required.

We now turn to (b). By (a), $\mathsf{Suffix}_{n_i}(\sigma) \models \phi_u$ for $0 \leq i < d$. Now suppose $0 \leq k < n_d$. By Proposition 3.6(a), $0 = n_0 \leq n_1 \leq \ldots$, and so $k = n_i + j$ for some $0 \leq i < d$ and $0 \leq j < n_{i+1}$. Write $\sigma = \langle s, (\beta_0, \beta_1, \ldots)\rangle$. By Proposition 3.6(e), $\beta_{n_i}, \beta_{n_i+1}, \ldots, \beta_{n_i+j-1}$ are all 0-transparent to ϕ, i.e. ϵ_u-transparent to ϕ_u, and so repeated applications of $\mathfrak{q}_u$ imply that $\mathsf{Suffix}_k(\sigma) \models \phi_u$. □

We now return to the proof of Proposition 3.9. Suppose that u is a leaf node, so $p = \phi_u \in AP$. Then for any infinite path ρ in M, $\rho \models \phi_u \Leftrightarrow p \in L(\mathsf{first}(\rho))$. Since $\rho \rightsquigarrow \sigma \Rightarrow \mathsf{first}(\rho) = \mathsf{first}(\sigma)$, $\mathfrak{p}_u$ holds. Statement $\mathfrak{q}_u$ follows from Definition 2.1 since $\mathsf{Suffix}_1(\pi) \models \phi_u \Leftrightarrow p \in L(\alpha_0(s))$.

Now suppose u is any node and $\mathfrak{p}_v \wedge \mathfrak{q}_v$ holds for all children v of u.

Suppose first that u has two children, v and w, and that $\phi_u = \phi_v \wedge \phi_w$. Then $\epsilon_v = \epsilon_w = \epsilon_u$. So $\mathfrak{p}_u \wedge \mathfrak{q}_u$ holds since, for any infinite path ρ, $\rho \models \phi_u$ iff $\rho \models \phi_v$ and $\rho \models \phi_w$.

Suppose instead that u has a single child v, and that $\phi_u = \neg\phi_v$. Then $\epsilon_v = 1 - \epsilon_u$. Now $\mathfrak{p}_u \wedge \mathfrak{q}_u$ holds since, for any infinite path ρ, $\rho \models \phi_u$ iff $\rho \not\models \phi_v$.

Suppose now that u has two children, v and w, and that $\phi_u = \phi_v \mathbf{U} \phi_w$. We will first consider the case $\epsilon_u = 1$. In this case, $\epsilon_v = \epsilon_w = 1$.

Let us first prove $\mathfrak{p}_u$. So assume ρ, σ are infinite paths starting at a state s, and $\rho \rightsquigarrow \sigma$. Suppose $\rho \models \phi_u$. We must show that $\sigma \models \phi_u$. Since $\rho \models \phi_v \mathbf{U} \phi_w$, there exists $d \geq 0$ such that $\mathsf{Suffix}_d(\rho) \models \phi_w$ and, for all $0 \leq i < d$, $\mathsf{Suffix}_i(\rho) \models \phi_v$. By Lemma 3.10(a), $\mathsf{Suffix}_{n_d}(\sigma) \models \phi_w$, while by Lemma 3.10(b), $\mathsf{Suffix}_k(\sigma) \models \phi_v$ whenever $0 \leq k < n_d$. This shows that $\sigma \models \phi_v \mathbf{U} \phi_w$, as required.

Let us now prove $\mathfrak{q}_u$. So suppose $\pi = \langle s, (\alpha_0, \alpha_1, \ldots)\rangle$ is an infinite path in M for which α_0 is 1-transparent to ϕ_u, and that $\pi \models \phi_u$. We must show that $\mathsf{Suffix}_1(\pi) \models \phi_u$. Since $\phi_u = \phi_v \mathbf{U} \phi_w$, there exists $d \geq 0$ such that $\mathsf{Suffix}_d(\pi) \models \phi_w$ and $\mathsf{Suffix}_i(\pi) \models \phi_v$ for $0 \leq i < d$. If $d > 0$, then we have

$$\mathsf{Suffix}_{d-1}(\mathsf{Suffix}_1(\pi)) = \mathsf{Suffix}_d(\pi) \models \phi_w$$
$$\mathsf{Suffix}_j(\mathsf{Suffix}_1(\pi)) = \mathsf{Suffix}_{j+1}(\pi) \models \phi_v \qquad (0 \leq j < d-1)$$

and hence $\mathsf{Suffix}_1(\pi) \models \phi_u$, as required. On the other hand, if $d = 0$, then $\pi \models \phi_w$, and applying the inductive hypothesis $\mathfrak{q}_w$ we conclude that $\mathsf{Suffix}_1(\pi) \models \phi_w$, which implies $\mathsf{Suffix}_1(\pi) \models \phi_u$. This completes the proof that $\mathfrak{p}_u \wedge \mathfrak{q}_u$ holds if $\epsilon_u = 1$.

Consider now the case $\epsilon_u = 0$. In this case, $\epsilon_v = \epsilon_w = 0$. We first prove $\mathfrak{p}_u$. So suppose that $\rho \not\models \phi_v \mathbf{U} \phi_w$. There are two possibilities: (i) for all $d \geq 0$, $\mathsf{Suffix}_d(\rho) \not\models \phi_w$, or (ii) for some $d \geq 0$, $\mathsf{Suffix}_d(\rho) \not\models \phi_v$ and $\mathsf{Suffix}_i(\rho) \not\models \phi_w$ for $0 \leq i \leq d$.

If (i) is the case, then by Lemma 3.10(b), for all $d \geq 0$ and $k < n_d$, $\mathsf{Suffix}_k(\sigma) \not\models \phi_w$. However, by Proposition 3.6(a), $\lim_{d\to\infty} n_d = \infty$, hence $\mathsf{Suffix}_k(\sigma) \not\models \phi_w$ for all $k \geq 0$, which means that $\sigma \not\models \phi_v \mathbf{U} \phi_w$, as required.

If (ii) is the case, then it follows from Lemma 3.10(a) that $\mathsf{Suffix}_{n_d}(\sigma) \not\models \phi_v$, while Lemma 3.10(b) implies that $\mathsf{Suffix}_k(\sigma) \not\models \phi_w$ for $0 \leq k < n_d$. Again, this means that $\sigma \not\models \phi_v \mathbf{U} \phi_w$, establishing $\mathfrak{p}_u$.

We now turn to the proof of $\mathfrak{q}_u$ in the case that $\epsilon_u = 0$. So suppose $\pi = \langle s, (\alpha_0, \alpha_1, \ldots)\rangle$ is an infinite path in M for which α_0 is 0-transparent to ϕ_u, and

that $\pi \not\models \phi_u$. We must show that $\mathsf{Suffix}_1(\pi) \not\models \phi_u$. Again, there are two cases to consider: (i) for all $d \geq 0$, $\mathsf{Suffix}_d(\pi) \not\models \phi_w$, or (ii) for some $d \geq 0$, $\mathsf{Suffix}_d(\pi) \not\models \phi_v$ and $\mathsf{Suffix}_i(\pi) \not\models \phi_w$ for $0 \leq i \leq d$.

If (i) is the case, then we certainly have

$$\mathsf{Suffix}_d(\mathsf{Suffix}_1(\pi)) = \mathsf{Suffix}_{d+1}(\pi) \not\models \phi_w \qquad (d \geq 0),$$

which implies $\mathsf{Suffix}_1(\pi) \not\models \phi_u$, as required.

So suppose (ii) is the case. If $d > 0$, then

$$\mathsf{Suffix}_{d-1}(\mathsf{Suffix}_1(\pi)) = \mathsf{Suffix}_d(\pi) \not\models \phi_v$$
$$\mathsf{Suffix}_j(\mathsf{Suffix}_1(\pi)) = \mathsf{Suffix}_{j+1}(\pi) \not\models \phi_w \qquad (0 \leq j \leq d-1),$$

whence again $\mathsf{Suffix}_1(\pi) \not\models \phi_u$, as required. So suppose $d = 0$. Then $\pi \not\models \phi_v$ and $\pi \not\models \phi_w$. By the inductive hypotheses $\mathfrak{q}_v$ and $\mathfrak{q}_w$, this means that $\mathsf{Suffix}_1(\pi) \not\models \phi_v$ and $\mathsf{Suffix}_1(\pi) \not\models \phi_w$. Hence $\mathsf{Suffix}_1(\pi) \not\models \phi_v \mathbf{U} \phi_w = \phi_u$. This establishes $\mathfrak{q}_u$ for the case $\epsilon_u = 0$, and completes the proof of Proposition 3.9

3.4. **Conclusion.** We can now complete the proof of Theorem 2.4. One direction is clear: since any path in $M^\flat$ is a path in M, we have $M \models \mathbf{A}\phi \Rightarrow M^\flat \models \mathbf{A}\phi$. So we must show that $M \not\models \mathbf{A}\phi \Rightarrow M^\flat \not\models \mathbf{A}\phi$. So suppose $M \not\models \mathbf{A}\phi$, i.e., there is some infinite path ρ in M, whose start state s is in S_0, such that $\rho \not\models \phi$. By Lemma 3.4, there exists an infinite path σ in $M^\flat$ starting at s such that $\rho \rightsquigarrow \sigma$. By Proposition 3.9, $\sigma \not\models \phi$. Hence $M^\flat \not\models \mathbf{A}\phi$, as required.

4. Experiments

4.1. **Methodology.** The applications we consider concern parallel programs that use the Message Passing Interface (MPI) [6]. We will use the formal definition of a model of an MPI program described in [12, 13]. A model essentially consists of an automaton for each process and a set of channels, each with a fixed sending and receiving process. The transitions may be labeled by local, send, or receive events. Each state in an automaton is either a *terminal state* (a state with no outgoing transitions), a *local-event state* (all transitions departing from that state are local), a *sending state* (there is only one departing transition and it is labeled by a send event), a *receiving state* (all the departing transitions are labeled by receive events), or a *send-receive state*—a state from which first a send can happen and then a receive, or first a receive then the send. An independence relation I may be defined so that $(t, t') \in I$ iff t and t' are from distinct processes or they are from the same process and one is a send and the other a receive.

The construction of the reduced structure is fairly standard and so we only summarize it here. The construction proceeds by depth-first search. Given the current state, the following sets are considered candidates for the ample set: (1) all enabled transitions in a single process, if that process is at a local event state; (2) all enabled send transitions in a single process, if that process is at a sending or send-receive state; (3) all enabled receive transitions in a single process, if that process is at a receiving or send-receive state for which every receiving channel has at least one queued message. A candidate set is rejected if (a) it is empty, (b) it contains a transition that is not transparent, or (c) it contains a transition that leads to a state already on the search stack. If no candidate set survives rejection, the set of all enabled transitions is used for the ample set, otherwise one candidate set is chosen using some heuristic.

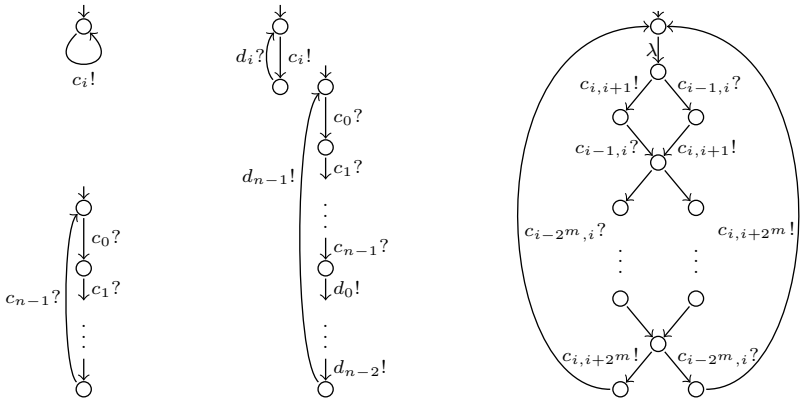
This strategy ensures that if a proper ample set T consists of transitions from a single process P, then any transition dependent on a transition in T is also in P; furthermore none of these dependent transitions are enabled and no action from another process can enable them. This suffices to show that **C1** is satisfied. It is not hard to see the other ample set conditions are satisfied as well.

Our heuristic proceeds by first looping through the processes in order of increasing process ID, searching for a proper ample set that consists entirely of *invisible* local event or receive transitions. If none is found, it then attempts to find a set consisting of invisible send transitions. If this fails, it then repeats these attempts but allowing transparent transitions (that are not necessarily invisible). If this also fails then the set of all enabled transitions is used. The choice of this heuristic is based on experience which suggests that invisible ample sets give the best reduction (when they exist), and those consisting of receives or local events generally do better than those consisting of sends. It also guarantees that the resulting structure is a subgraph of the structure that would result from the invisible ample search; in fact, the heuristic can always be designed to have this property since any invisible transition is also transparent.

The experiments were conducted using a modified version of the Java program described in [11]. The program takes as input a model of an MPI program and performs either a full or a reduced search. For a reduced search, the program also takes as input a predicate specifying which transitions are to be considered invisible/transparent in the POR algorithm described above. The predicates specifying transparency and invisibility were constructed by hand for each experiment, though this process was sufficiently straightforward that it appears it would not be difficult to automate. In each experiment, we used this approach to search (1) the full Kripke structure, (2) the reduced structure using invisible POR, and (3) the reduced structure using transparent POR. In each case, the number of states was recorded. The program and all experimental data can be downloaded from `http://www.cis.udel.edu/~siegel/projects`.

4.2. **3 Models and 8 Experiments.** Our first model (Fig. 2, left) is derived from the "multiple producer, single consumer" (MPSC) program of [14, Ex. 2.18]. In this program, n producers send messages to a single consumer, which consumes from the producers in a cyclic fashion. As is the case with the other models, the data is abstracted away altogether. The property to check is that producer 0 never becomes permanently blocked, which can be expressed in LTL as $\mathbf{AGF}\neg\mathsf{full}(c_0)$, where c_0 is the channel used by producer 0 to send to the consumer. Note that all transitions other than $c_0!$ and $c_0?$ are invisible, while all transitions other than $c_0?$ are transparent. In experiment (a) the number of producers is fixed at 3 and the channel size is scaled. In experiment (b), the channel size is fixed at 4 and the number of producers is scaled.

The second model (Fig. 2, middle) derives from the "coordinator barrier" system described in [1]. This system consists of n worker processes and one coordinator process used to enforce a repeatable barrier among the workers. Each worker loops forever and in each iteration performs some local computation and then enters and exits the barrier. Entering the barrier is modeled by sending a message on c_i and exiting by receiving a message on d_i. Three experiments were performed on this system, each using a different property. The property for experiment (c) states that if worker 0 enters the barrier then eventually all workers will be inside the



MPSC Coordinator Barrier Dissemination Barrier

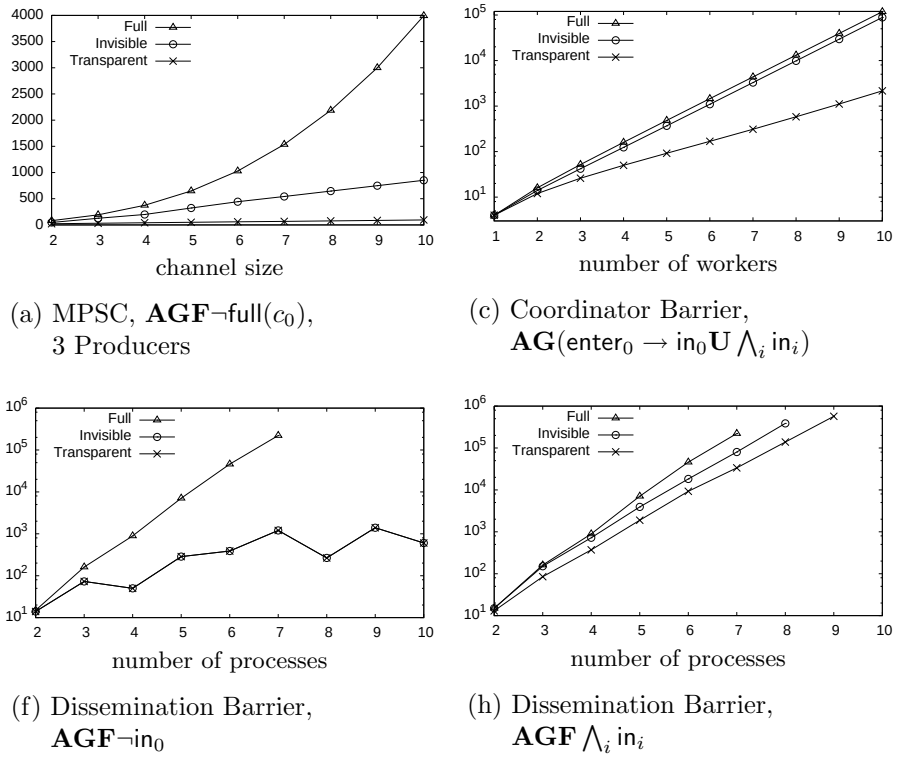
FIGURE 2. Models

barrier and worker 0 will remain in the barrier until that time; this is expressed as $\mathbf{AG}(\mathsf{enter}_0 \rightarrow \mathsf{in}_0 \mathbf{U} \bigwedge_i \mathsf{in}_i)$, where $\mathsf{enter}_i \equiv \neg\mathsf{empty}(c_i)$ and in_i holds when worker i is not at its start state. The property for (d) is that worker 0 will be outside of the barrier infinitely often and is expressed $\mathbf{AGF}\neg\mathsf{in}_0$. The property for (e) is that all processes will be (simultaneously) inside the barrier infinitely often: $\mathbf{AGF} \bigwedge_i \mathsf{in}_i$.

The third model (Fig. 2, right) derives from the "dissemination barrier" system described in [1]. Instead of using a coordinator, the n processes use a symmetric protocol to impose the barrier among themselves. The protocol proceeds in stages $0, 1, \ldots, \lceil \log_2(n) \rceil - 1 \equiv m$. In stage j, for each i $(0 \leq i < n)$, process i sends a message to process $i + 2^j$ and receives a message from process $i - 2^j$ using a send-receive call. (Process IDs are reduced modulo n.) The same three properties used for the coordinator barrier were used here, though the atomic propositions are defined slightly differently: enter_i holds when process i is at the state s_1 which is the target of the transition labeled by local event λ; in_i holds when process i is at neither the start state nor at s_1. This yielded experiments (f), (g), and (h).

The results of experiments (a), (c), (f), and (h) are shown in Fig. 3. The graph for (b) is similar in complexity to that of (a); (d) is similar to (h); (e) is similar to (f); and (g) is similar to (c).

In all cases, the number of states explored by the full search increased exponentially with the number of processes (and with the channel size in the case of MPSC). The effect of the two reduction strategies is more varied. In (a), both the invisible and the transparent searches reduced the complexity to a linear function, and the slope for the transparent function is significantly lower than that of the invisible one. In (c), all functions grow exponentially, but the invisible search appears to have an exponent equal to that of the full search, while the transparent one has a significantly smaller exponent. In the case of (f), the transparent and invisible algorithms search identical spaces, which are significant improvements over the full search. For (h), a moderate improvement is obtained in moving from full to invisible, and again from invisible to transparent.



(a) MPSC, $\mathbf{AGF}\neg\mathsf{full}(c_0)$, 3 Producers

(c) Coordinator Barrier, $\mathbf{AG}(\mathsf{enter}_0 \rightarrow \mathsf{in}_0 \mathbf{U} \bigwedge_i \mathsf{in}_i)$

(f) Dissemination Barrier, $\mathbf{AGF}\neg\mathsf{in}_0$

(h) Dissemination Barrier, $\mathbf{AGF} \bigwedge_i \mathsf{in}_i$

FIGURE 3. Number of states explored (y-axis) by three search strategies

5. Related Work

The reader is referred to [3], [9], [2, Chap. 10], and the references cited in those works for a guide to the large literature on partial order reduction. Two of the pioneering works on LTL$_{-X}$-preserving POR methods are [15] and [7], both of which require the invisibility condition.

A well-known automata-theoretic model checking algorithm involves the search for reachable acceptance cycles in the product of an automaton representing the Kripke structure and a Büchi automaton corresponding to $\neg\phi$. This search can take place *on the fly*, i.e., without first constructing the Kripke structure. An algorithm combining the on-the-fly approach with the (invisibility-based) ample-set POR algorithm is presented in [8] and is similar to the algorithm implemented in Spin [4]. A relaxation of the invisibility condition in this context is investigated in [10]. The idea is to dynamically reduce the set of invisible transitions as the on-the-fly search progresses. The algorithm requires a specific construction for the Büchi automaton which annotates states with subformulas of $\neg\phi$; the set of invisible transitions is a function of the propositions appearing in certain subformulas annotating the current state of the Büchi automaton.

A POR-like algorithm for verifying properties such as freedom from potential deadlock in models of MPI programs is studied in [11]. Freedom from potential deadlock can be expressed as $\mathbf{AG}\neg\mathsf{phalt}$, where phalt holds in any state for which the only enabled transitions are sends which cannot be immediately followed by their

matching receives. The key ingredients in that algorithm are (1) the introduction of synchronous transitions and (2) the observation that receives, synchronous, and local event transitions cannot change the truth value of phalt from *true* to *false*. In the language of this paper, those transitions are transparent to $\mathbf{G}\neg\mathsf{phalt}$. In fact, the main theorem of this paper evolved out of an attempt to generalize the observations of [11].

Other relaxed-visibility techniques are described in [5]. Those techniques apply to two specific CTL formulas ($\mathbf{EF}\ p$ and $\mathbf{AG}\ \mathbf{EF}\ p$) and are investigated in the context of Petri nets. They involve certain sets of transitions called *up sets* and *down sets*. Though the definitions are somewhat complex, the core idea is the distinction between transitions that can only change the truth value of p from *true* to *false* and those that can only change that value from *false* to *true*, an idea that is also central in this paper.

6. Conclusion

We have described a simple modification to the ample set POR framework. The modification is a relaxation of the invisibility condition that distinguishes between propositions that occur only positively in the formula being checked, and those that occur only negatively. The modified framework may open up opportunities for reduction that do not exist in the standard framework. Furthermore, any heuristic for choosing ample sets in the traditional framework can be extended so that the modified algorithm does no worse (in terms of numbers of states or transitions explored) than the standard algorithm.

To take advantage of the modified framework, one must be able to identify program statements that preserve the truth (or falsity) of propositions occurring in the formula. While sophisticated automated reasoning approaches might be brought to bear on this problem, there are plenty of commonly-occurring scenarios that can be easily (and probably automatically) detected. For example, if c is a FIFO channel, then a send operation on c preserves the truth of $\mathsf{full}(c)$ and the falsity of $\mathsf{empty}(c)$; a receive on c preserves the truth of $\mathsf{empty}(c)$ and the falsity of $\mathsf{full}(c)$. If x is a numeric variable then the assignment $x \leftarrow x - 1$ preserves the truth of $x \leq N$ and $x \leftarrow x + 1$ preserves the truth of $x \geq N$. If p is a predicate that holds iff the flow of control of a process is at a particular point then any statement that results in transferring control to that point preserves the truth of p, and any statement that results in transferring control to any other point preserves the falsity of p.

We have applied the improved algorithm in the context of the verification of some simple MPI programs. Our experiments show various degrees of (and in some cases, dramatic) improvement over the standard algorithm. Yet, while the models do share some features with typical MPI programs (e.g., most or all processes are similar or identical), they are relatively simple, and a broader study using more complex examples is needed in order to ascertain the effectiveness of the transparent framework.

Beyond this, the most important work remaining involves combining the optimization described here with other techniques, such as the on-the-fly algorithm or the dynamic invisibility technique of [10]. It is easy enough to see *how* to combine them—one can simply replace *invisible* with *transparent* in the descriptions of these

algorithms—but the correctness proofs are not at all obvious and remain for future work.

References

[1] Gregory R. Andrews. *Foundations of Multithreaded, Parallel, and Distributed Programming*. Addison-Wesley, 2000.

[2] Edmund M. Clarke, Jr., Orna Grumberg, and Doron A. Peled. *Model Checking*. MIT Press, Cambridge, 1999.

[3] Patrice Godefroid. *Partial-Order Methods for the Verification of Concurrent Systems: An Approach to the State-Explosion Problem*. Springer, Berlin, 1996.

[4] Gerard J. Holzmann and Doron Peled. An improvement in formal verification. In Dieter Hogrefe and Stefan Leue, editors, *Proceedings of the 7th IFIP WG6.1 Intl. Conference on Formal Description Techniques (Forte '94)*, volume 6 of *IFIP Conference Proceedings*, pages 197–211. Chapman & Hall, 1995.

[5] L. M. Kristensen, K. Schmidt, and A. Valmari. Question-guided stubborn set methods for state properties. *Formal Methods in System Design*, 29(3):215–251, November 2006.

[6] Message Passing Interface Forum. MPI: A Message-Passing Interface standard, version 1.1. `http://www.mpi-forum.org/docs/`, 1995.

[7] Doron Peled. All from one, one for all: On model checking using representatives. In Costas Courcoubetis, editor, *Computer-Aided Verification, 5th Intl. Conference (CAV '93)*, volume 697 of *LNCS*, pages 409–423. Springer-Verlag, 1993.

[8] Doron Peled. Combining partial order reductions with on-the-fly model-checking. *Formal Methods in System Design*, 8(1):39–64, January 1996.

[9] Doron Peled. Ten years of partial order reduction. In Alan J. Hu and Moshe Y. Vardi, editors, *Computer Aided Verification, 10th Intl. Conference (CAV '98)*, volume 1427 of *LNCS*, pages 17–28. Springer, 1998.

[10] Doron Peled, Antti Valmari, and Ilkka Kokkarinen. Relaxed visibility enhances partial order reduction. *Formal Methods in System Design*, 19(3):275–289, November 2001.

[11] Stephen F. Siegel. Efficient verification of halting properties for MPI programs with wildcard receives. In Radhia Cousot, editor, *Verification, Model Checking, and Abstract Interpretation: 6th Intl. Conference (VMCAI 2005)*, volume 3385 of *LNCS*, pages 413–429, 2005.

[12] Stephen F. Siegel and George S. Avrunin. Modeling MPI programs for verification. Technical Report UM-CS-2004-75, Department of Computer Science, University of Massachusetts, 2004.

[13] Stephen F. Siegel and George S. Avrunin. Modeling wildcard-free MPI programs for verification. In *Proceedings of the 2005 ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '05)*, pages 95–106. ACM Press, 2005.

[14] Marc Snir, Steve Otto, Steven Huss-Lederman, David Walker, and Jack Dongarra. *MPI—The Complete Reference, Volume 1: The MPI Core*. MIT Press, second edition, 1998.

[15] Antti Valmari. A stubborn attack on state explosion. *Formal Methods in System Design*, 1(4):297–322, December 1992.

The Verified Software Laboratory, Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA

E-mail address: `siegel@cis.udel.edu`